Collision Avoidance and Object Following Control for a Mobile Robot in Human Living Environment

（人間生活環境における移動ロボットの衝突回避と物体追従制御）

January, 2016

Doctor of Engineering

TRESNA DEWI

Toyohashi University of Technology

Doctor of Engineering Thesis

in

Systems Engineering Laboratory
Department of Mechanical Engineering
Toyohashi University of Technology

Author:

Tresna Dewi

Supervisor:

Prof. Naoki Uchiyama

Committee members:

Prof. Zhong Zhang
Prof. Tetsuo Miyake

Abstract

With recent technological developments, mobile robots are expected to occupy the same environments as humans and provide fully autonomous or semi-autonomous assistance. Robot navigation is the safe movement of a robot toward its goal and is accomplished by knowledge of the environment. Obstacle avoidance is one of the main requirements of robot navigation. During the past few years, the potential field method for obstacle avoidance and target setting has gained popularity among roboticists. In this approach, the obstacles and targets exert repulsive and attractive forces on the robot, respectively.

As our most powerful sense, vision provides us with an enormous amount of information and enables intelligent interaction with our dynamic environment. Vision-based robotics systems have recently increased in popularity, and several new approaches have been proposed. Despite the huge efforts invested in creating effective low-cost service mobile robots, vision-based motion control and path planning remain theoretically challenging. The basic principle involves iterating two structures from motions captured by a stereo camera. As the distance to a detected object and motion can be predicted and measured by depth mapping, the stereo camera can be substituted by a single red–green–blue depth (RGB-D) camera. A popular RGB-D camera is the Kinect sensor, a motion sensing device developed by Microsoft Corporation.

Vision-based robot navigation might enable the production of service robots that can track and follow moving targets. Tracking mobile robots generally require expensive sensors such as range finders, tend to disregard obstacle avoidance, and are developed as single robots. Moreover, their algorithms are complex.

In this study, we combined the potential field method with visual sensing to create a moving-object-following robot; that is, a robot that tracks and follows an object attached to a human, wheelchair, or another robot (leader robot). The attached object considered was a blue circular mark. The robot was mounted with a Kinect sensor, and the position of the blue circle and depth mapping for the distance calculation were detected by an RGB image detection system. The designed mobile robot system can track and follow a moving object in environments wherein humans coexist; using only the information acquired by vision sensors and IR proximity sensors. The designed reference controller generates a reference trajectory for obstacle avoidance and the robot is moved along its reference trajectory by a simple proportional integral (PI) controller. This proposed method enables collision avoidance, is robust to brief occlusion, and to some extent,

is also applicable to swarm robots. The simple algorithm makes the proposed method easily modifiable to robotic applications other than mobile robots.

First, we derive reference models from the dynamic modeling of the two-wheeled differential drive mobile robot with non-holonomic constraints. The collision avoidance scheme is achieved by the model reference control based on the potential field path planner. A system integration method, by which the robot moves in a dynamic environment, is also presented. The effectiveness of the proposed controller is demonstrated by discussing the results of simulations and experiments. The next step is to present a new collision avoidance approach for four-wheeled human-operated mobile robots. Because the proposed method considers the non-holonomic constraint of a mobile robot, it provides practical collision avoidance control. In a verification experiment, entirely unskilled operators maneuvered the robot to its destination without collisions, demonstrating the effectiveness of the proposed approach.

The proposed method also presents the collision avoidance control technique applicable for swarm robots moving in a rehabilitation environment containing static and moving obstacles. A leader–follower formation is adopted, in which the leader of the swarm robot team follows the rehabilitee, while the other robot follows the leader. The effectiveness of the proposed system is demonstrated by presenting the results of a simulation study in which several static and dynamic obstacles were placed in a human living environment.

The final step is combining the potential field method with visual sensing to create a moving-object-following robot; that is, a robot that tracks and follows an object attached to a human, wheelchair, or another robot (leader robot). In this study, the attached object was a blue circular mark. The robot was mounted with a Kinect sensor, and the position of the blue circle and the depth mapping for the distance calculation were detected by RGB imaging. The effectiveness of the proposed method is discussed in the context of human-following, wheelchair-following, and leader robot-following systems and by presenting an experiment that extended into the hallway outside the laboratory testbed. Experiments were conducted in several environmental settings.

In order to prove that proposed method easily modifiable to robotic applications other than mobile robots, we discuss the possibility of achieving human and object following without specific shape detection. We compare the proposed system design with the work undertaken by several references. The effectiveness of the proposed method was experimentally evaluated in several course settings.

This thesis proposes a moving-object-following mobile robot functioning as a mobile service robot in a human living environment. The algorithms and hardware are simplified as much as possible in this design, enabling the proposed robot to assist in a wide variety of dull, repetitive tasks in human environments, such as transporting the disabled. Despite the application of inexpensive Kinect and proximity sensors, the system performed as expected in this investigation, in which the effectiveness of the proposed method was confirmed in different experimental scenarios. The possibility of creating swarms of service robots by the proposed method is also discussed in this thesis.

Acknowledgements

All praises are to ALLAH, the most beneficent and merciful, the Lord of universe. Without His blessings, this works could not be accomplished.

I would like to express my deep appreciation to Prof. Naoki Uchiyama for his sincere advises and passionate guidance. I have learned not only scientific knowledge, but also the method of scientific thinking, how to identify the problems and formulate the method to find the solutions. I have also learned much about how to communicate an idea to other people so that it could be understandable and acceptable.

I would like to thank to Assoc. Prof. Shigenori Sano, for all his excellent help and rewarding discussion and for always taking his time to answer my questions and welcoming me to discuss my problems.

I would like to express my gratitude to the committee members, Prof. Zhong Zhang and Prof. Tetsuo Miyake, for their great support, valuable advice and comments. I also would like to thank to Assistant Prof. Tatsuhiko Sakaguchi for giving me valuable discussions and suggestions in completion of this work. In addition, I also would like to thank to all members of Robotics and Mechatronics Laboratory for their assistance and friendship during my study.

Furthermore, I would like to thank to Japanese goverment (MONBUKAGAKUSHO: MEXT) for providing me scholarship during my study and State Polytechnic of Sriwijaya, Indonesia, which gave me the chance and support to study at Toyohashi University of Technology. I also would like to thank the Hori Sciences and Arts Foundation, Nagoya, Japan, for their financial support and thanks to all Toyohashi University staff members for their kindly assistance and friendship.

This work is dedicated for my parents, brother, and sister, who always keep on giving me prayers and strength.

Last but not least, I would like to thank to PPI (Indonesia Student Association) members and their family for fun and warm friendship.

Tresna Dewi

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Along with recent technological developments, autonomous mobile robots have been introduced to human living environments. Mobile robots are expected to occupy the same environments as humans and to provide fully autonomous or semi-autonomous assistance. For example, they are expected to be deployed in hospitals [1], restaurants [2], sports activities [3], and museums [4]. This thesis discusses mobile robots that function as service robots in human living environments.

1.1 Introduction

1.1.1 Background

Autonomous mobile robots are widely used in both domestic and industrial applications. Fully automated robots are desired to perform household chores, nursing, and welfare work and to assist with industrial tasks performed by skilled workers. These applications require an efficient path-planning algorithm that can be implemented at low cost. However, despite the immense efforts devoted to creating effective low-cost mobile service robots, motion control and path planning remain theoretically challenging, chiefly because of the non-holonomic constraint that prevents a robot from moving parallel to its main axes [5] [6] [7].

Robot navigation is the means by which a robot safely moves toward a goal using its knowledge of the environment. Navigation is among the most important and elementary functions of mobile robotics. Navigation methods borrowing techniques from artificial intelligence, artificial vision, and other fields have been widely reported in the literature. Most navigation algorithms assume a stationary goal [10] [11]. The problem becomes more difficult when the target is moving [12]-[15]. Obstacle avoidance is one of the main functions required in autonomous robot navigation. The effectiveness of obstacle avoidance will be improved by designing system integration for the robot. A current trend in robotics is installing different types of sensors having different characteristics to improve the performance of robot system and also benefit from the reduced cost of sensors [16] [17] [18]. System integration is defined as the process of bringing together the component subsystems into one system and ensuring that the subsystems function together as a system. This result of linking the process of different systems is expected to improve the performance of a robot system.

According to the International Federation of Robotics (IFR), a service robot is a robot that semi- or fully autonomously operates to enhance the well-being of humans and equipment, excluding those used in manufacturing operations. Designing system integration enable services robots to function well in dynamic environments, such as human living environments. The application range of service robots in human environments is widening as more sensors are being combined in single robots. A service robot that assists a human in daily life (whether semi- or fully autonomously) must be able to continuously track the human. This tracking is ensured by a visual sensor and laser range finder [19] [20].

Human detection and tracking are usually performed by a vision-based method, which senses the environment without making physical contact. As a feedback control method, vision-based control systems guide robots' motion by using one or more visual sensors. The control inputs for robot actuators are produced by processing image data relative to a target object.

As our most powerful sense, vision provides us with an enormous amount of information and enables intelligent interaction with our dynamic environment [21]-[26]. Therefore, robots with sensors that mimic the capabilities of the human vision system have been actively researched. The first such vision systems were based on the stereo vision of pinhole cameras fitted to mobile robots or the end-effectors of robot arms [13] [15]. This method has rapidly improved, as multiple camera features (such as IR sensing, image detection, and depth measurement) can now be combined.

Visual odometry estimates the motion of a robot from the visual input alone. The basic principle involves iterating two structures captured by a moving stereo camera. As the distance to a detected object and the robot's motion can be predicted and measured by depth mapping, the stereo camera can be substituted by a single red-green-blue depth (RGB-D) camera.

A popular RGB-D camera is the Kinect sensor, a motion-sensing device developed by Microsoft Corporation. Kinect sensors not only enable users to develop open-source drivers and programs, but also are relatively inexpensive. Furthermore, Kinect sensors detect images, distances, and depths more easily than other types of camera can.

The potential field method of path planning is effective because it is simple, and its basic mechanism is borrowed from nature (particle navigation in a magnetic field). Robotics emulates this phenomenon by simulating an artificial potential field that attracts a robot to its target. The visual potential field is developed by setting the robot navigation to a sequence of images, not merely by referring to the obstacle configurations.

In this study, the potential field method was combined with visual sensing to create a moving-object-following robot; that is, a robot that could track and follow an object attached to a human, wheelchair, or another robot (leader robot). The attached object in this study was a blue circular mark. The robot was mounted with a Kinect sensor, and the position of the blue circle and depth mapping for the distance calculation were detected by an RGB image detection system [21] [23] [27]-[30].

The developed moving-object-following system could be used in service robots deployed in hospitals and/or rehabilitation environments, for example. The primary objectives of rehabilitation robots are to fully or partially assist disabled people and to support rehabilitees' manipulative functions [33]-[37]. Conventional rehabilitation programs depend on the experience and manual manipulations of a rehabilitator. Because rehabilitation requires care and the number of rehabilitees continues to increase, well-designed service robots may offer the support required for careful rehabilitation.

Most existing service robots are developed as singletons. However, service might be improved if multiple service robots cooperated as a swarm robot team, particularly in human services and rehabilitation applications. Besides supporting rehabilitees' movements, swarm robots could collect or transport certain objects during rehabilitation and perform multiple other tasks [1] [31]-[37].

This thesis proposes a moving-object-following mobile robot functioning as a mobile service robot in a human living environment. The algorithms and hardware are simplified as much as possible in this design, enabling the proposed robot to assist in a wide variety of dull, repetitive tasks in human environments, such as transporting the disabled. Despite the application of inexpensive Kinect and proximity sensors, the system performed as expected in this investigation, in which the effectiveness of the proposed method was confirmed in different experimental scenarios. The possibility of creating swarms of service robots by the proposed method is also discussed in this thesis.

1.1.2 Potential Field Method

During the past few years, the potential field method for obstacle avoidance and target setting has gained popularity among roboticists. The idea of imaginary forces acting on a robot was suggested by Khatib [38]. In this approach, obstacles and targets exert repulsive and attractive forces on the robot, respectively, which combine to determine the magnitude and direction of the total force. By following the direction of the total force, a robot can avoid obstacles and travel a safe path to its target [39][40]

The artificial potential field method uses a scalar function called the potential function. This function has a minimum at the goal point and maxima around obstacles. The function slopes down toward the end points, so the robot follows the negative gradient of the total potential field to reach its target. The shape of the potential field is usually independent of object configurations beyond a predefined distance from the robot; for this reason, the potential field method is often referred to as a local method. The popularity of this method is largely rooted in its simplicity and elegance. Furthermore, it can be quickly implemented and provides acceptable results with few refinements.

The potential field method was designed for real-time efficiency rather than guaranteed completion of an assignment [41]. In essence, it is a fastest-descent optimization procedure; therefore, it could steer the robot into a local potential function other than the target position. Moreover, it may fail to find a free path, even if one exists. The current algorithm copes with local minima problems, dynamic obstacles and field boundaries. If a robot becomes stuck in a local minimum, it can be freed by placing virtual obstacles behind the robot, pushing it onwards. Dynamic obstacles are treated in similar fashion; collision points are calculated and virtual obstacles are placed at those positions, driving the robot away from them [42]. The relevant obstacles are identified from among all obstacles by parameter settings.

1.1.3 Swarm Robot

The swarm intelligence paradigm exhibits interesting properties such as robustness, flexibility, and ability to solve complex problems by exploiting parallelism. In several robotics implementations of this paradigm, these properties were confirmed to control populations of physically independent mobile robots. Most of the swarm intelligence studies have been inspired by the natural swarming behaviors of social insects, fish, or mammals, in which individuals interact with other members of the swarm. Rapid advances in sensing, computing, and communication technologies have led to the development of autonomous robots that function in outdoor environments. Some required tasks are too complex to be achieved by single robots, requiring cooperative multi-robot systems. Such requirements justify the use of swarm intelligence in robotics.

In a swarm robot system, two or more robots are assigned the task of formation control [43]; that is, controlling their relative positions and orientations in the group while allowing the group to move as a whole. The robot formation control problem has been tackled by three main approaches: behavior-based, virtual structure, and leader–follower. In the leader–follower approach, the robot designated as the leader moves along a desired trajectory while the follower robot maintains the desired distance and orientation from the leader. Leader–follower architectures are particularly appreciated for their simplicity and scalability. In the leader–follower approach, it is only necessary to specify the leader's trajectory and the desired relative positions and orientations between the leaders and followers. When the leader's motion is known, the relative desired positions (distances and heading angles) of the followers can be achieved by local control of the followers.

Swarms of mobile service robots operating in human living environments must cope with dynamic environmental changes. Hence, the fundamental function of such robots is to avoid static and dynamic obstacles. In particular, members of the swarm team must maintain their velocities and avoid collisions with their swarm mates [11][44]-[51]. In previous studies, swarm robots have avoided obstacles using proximity [45] and vision sensors [48][52]. They have also used radio frequency identification (RFID) for location and navigation [52][54][53][55].

1.1.4 Vision-based Control

Vision-based robot navigation has long been a fundamental goal in both robotics and computer vision research. Although the problem is largely solved for robots equipped with active range-finding devices, it remains challenging for robots equipped with vision sensors alone, for several reasons. Cameras have evolved as attractive sensors, realizing economically viable systems with sensors having lower technical specification. Another type of vision-based control is visual servoing, which minimizes a visually specified task by visual feedback control of the robot's motion. However, controlling the 6D pose of the end-effector based on a 2D image features is a challenging task and requires non-linear projection and degenerate features. Vision-based sensors are employed in diverse applications, such as industrial, health, space, humanoid, and service robots [5][6].

The flexibility and accuracy of robotic systems have been improved by vision feedback control loops. The visual servoing approach controls the robot using the information provided by a vision system. Vision systems are generally classified by their numbers of cameras and their positions. Most applications adopt single-camera vision systems, which are cheaper and easier to build than multi-camera vision systems. On the other hand, the two cameras in a stereo configuration have a common field of view, which resolves several computer vision problems. In an in-hand system, the camera(s) are mounted on the robot; in an out-hand or stand-alone system, the camera(s) observe the robot from elsewhere. Hybrid systems, in which one camera observes in-hand while the other observes from a stand-alone position, have also been developed.

The penetration of mobile robots into wide consumer markets is hindered by the unavailability of powerful, versatile, and cheap sensing methods. In terms of the information-to-cost ratio, vision technology offers the greatest promise. Cameras of acceptable accuracy are cheaper than laser and sonar scanners, and their cost continues to decrease. With the reduced price and improved performance of mass-marketed camera technologies, vision potentially offers a portable and cost-effective solution. Moreover, they can provide information unavailable to other sensors; for instance, they can provide semantic information about a scene by interpreting its visual appearance, not merely its geometrical information.

The current trend in robot navigation is to replace traditional range sensors with vision-based sensors. Vision-based robotics systems have recently increased in popularity, and several new approaches have been proposed. These systems analyze the images of scenes captured by the camera attached to a robot and plan the robot's actions based on the visual cues [21][23][27]-[30]. The sensors may be regular cameras (single or multiple) or omnidirectional vision sensors that survey the environment [13][14][56] [57]. These approaches differ mainly by their methods of perceiving the scene and extracting the salient features. The non-trivial problems of navigating an agent based on visual information have attracted much attention [40][58][59].

Vision-based robot navigation control has become a fundamental goal in both robotics and computer vision research. Indoor and outdoor robot navigation has significantly progressed throughout the past two decades. For example, in the early stages, an autonomous indoor mobile robot could not navigate through a cluttered hallway; today, this task presents little challenge. A camera affixed to a robot present a greater challenge, because the camera's field of view must reflect changes in the robot's position [23]. Unless the robot's position is known throughout the motion, the system must then choose which portion of the scene will be captured by the camera at a given time instant.

Vision-based robot navigation might enable the production of service robots and/or swarms of mobile robots that can track and follow moving targets. Tracking mobile robots generally require expensive sensors such as range finders [15][19][57] and visual compasses [60], tend to disregard obstacle avoidance [61][62], and are developed as single robots [21][64]. Moreover, their algorithms are complex.

1.2 Motivation and Research Objectives

Mobile robots are most likely to be applied as service robots in human living environments, as they can autonomously move from place to place without external human operators or be semi-autonomously operated by humans. In recent years, mobile robots have become sufficiently autonomous to be deployed in households, hospitals, restaurants, and museums. Unfortunately, mobile service robots still require complex systems, complicated algorithms, and expensive sensors such as laser range finders.

Technological advances and the reduced prices of electronic components have increased the chances of deploying mobile robots in human living environments. The reduced prices of sensors also enable the installation of two or more sensors in a robot without increasing the price of the robot. A simple, easily modifiable method will create a robot that changes its location.

To this end, this thesis proposes a simple design method for vision-based control of a mobile service robot. The robot can track and follow a moving object while avoiding dynamic and static obstacles. The proposed method can be easily modified to robot applications other than mobile robots.

1.3 Thesis Contribution

The main contributions of this thesis are as follows

1. The designed mobile robot system can track and follow a moving object in a human living environment, using only the information acquired by vision sensors and IR proximity sensors.
2. The designed reference controller generates a reference trajectory and the robot is moved along its reference trajectory by a simple proportional integral (PI) controller.
3. The proposed method enables collision avoidance and is robust to brief occlusion.
4. To some extent, the proposed method is also applicable to swarm robots.
5. The proposed method is easily modifiable to robotic applications other than mobile robots.

1.4 Thesis Organization

This thesis is divided into eight chapters. The contents of each chapter are summarized below:

- Chapter 2: This chapter describes the kinematics and dynamic modeling of the two-wheeled differential drive mobile robot with non-holonomic constraints. The derived models are used in the controller design. This chapter provides fundamentals for the controller design in J.1, and C.1-C.3. This chapter also details the experimental testbed, sensor configuration, and testbed parameters.

- Chapter 3: This chapter presents the collision avoidance scheme achieved by the model reference control based on the potential field path planner. The designed controllers installed in the mobile service robot are presented in the subsequent chapters. A system integration method, by which the robot moves in a dynamic environment, is also presented. The effectiveness of the proposed system is demonstrated by discussing the results of simulations and real experiments. This chapter presents the collision avoidance control technique for swarm robots moving in a rehabilitation environment containing static and moving obstacles. The method by which system integration achieves motion in a dynamic environment is presented. The proposed method combines the information obtained by several proximity sensors, an image sensor, and localization sensors (RFID system). A leader–follower formation is adopted, in which the leader of the swarm robot team follows the rehabilitee, while the other robot follows the leader. The robot controllers consist of a reference and a PI controller. The reference controller generates the robot motion trajectory by referring to the sensor information in real time, and the PI controller directs the robots along that trajectory. The effectiveness of the proposed system is demonstrated by presenting the results of a simulation study in which several static and dynamic obstacles were placed in a human living environment. This chapter also provides the controller design applied in J.1, and C.1-C.3.

- Chapter 4: This chapter presents a new collision avoidance approach for four-wheeled human-operated mobile robots. The approach is based on inexpensive IR proximity sensors. Because the proposed method considers the non-holonomic constraint of a mobile robot, it provides practical collision avoidance control. In a verification experiment, entirely unskilled operators maneuvered the robot to its destination without collisions, demonstrating the effectiveness of the proposed approach. In future studies, the presented linear analysis will be extended to more general cases and the proposed robot system will be adapted for more complex environments. The study in this chapter is published in J.2.

- Chapter 5: This chapter combines the potential field method with visual sensing to create a moving-object-following robot; that is, a robot that tracks and follows an object attached to a human, wheelchair, or another robot (leader robot). In this study, the attached object was a blue circular mark. The robot was mounted with a Kinect sensor, and the position of the blue circle and the depth mapping for the distance calculation were detected by RGB imaging. A modified version of the target positioning method described in Chapter 3 is presented as an object detection method. The effectiveness of the proposed method is discussed in the context of human-following, wheelchair-following, and leader robot-following systems and by presenting an experiment that extended into the hallway outside the laboratory testbed. Experiments were conducted in several environmental settings. This chapter is related to the publication in C.2, and C.3.

- Chapter 6: This chapter discusses the possibility of achieving object and human following by a simple algorithm and integrating two cheap sensors' data. The object tracking and following methods in this chapter adopt the RGB color detection technique described in Chapter 5, without specific shape detection. The human-following system utilized skeleton detection, and the Kinect sensor detection was integrated with the data from the IR proximity sensor to achieve an effective system. Both sensors imposed attractive forces toward the target (the blue mark and skeleton for object- and human-tracking and following, respectively). The effectiveness of the proposed method was experimentally evaluated in several course settings. This chapter is related to the publication in C.2, and C.3.
- Chapter 7: This chapter summarizes the thesis and suggests future extensions of the presented approaches.

Chapter 2

Modeling of Mobile Robot Dynamics and Experimental Setup

2.1 Introduction

A mobile robot changes its location, either autonomously without the assistance of a human operator or semi-autonomously under human operation. The difference between mobile robots and most industrial robots is that the former move freely within predefined workspaces, whereas the latter are confined to their assigned workspaces. This mobility capability enables such robot to be deployed in a wide range of structured and unstructured environments.

Wheeled mobile robots (WMRs) are very popular in applications requiring relatively low mechanical complexity and energy consumption. The maneuverability of a WMR depends on the wheels and drives used; three DOFs enable the maximal maneuverability needed for planar motions, such as movement through hospitals, museums, warehouse floors, and roads. Non-holonomic WMRs have less than three planar DOFs but are simpler and cheaper to construct because they require fewer than three motors. The main problems in WMR design are control, stability, maneuverability, and traction.

WMRs are typically installed with differential drives, which consist of two fixed powered wheels mounted on the left and right sides of the robot platform. The two wheels are independently driven. Balance and stability are provided by one or two passive castor wheels. Differential drive is the simplest mechanical drive because it requires no rotation around the driven axis. If the wheels rotate at the same speed, the robot moves

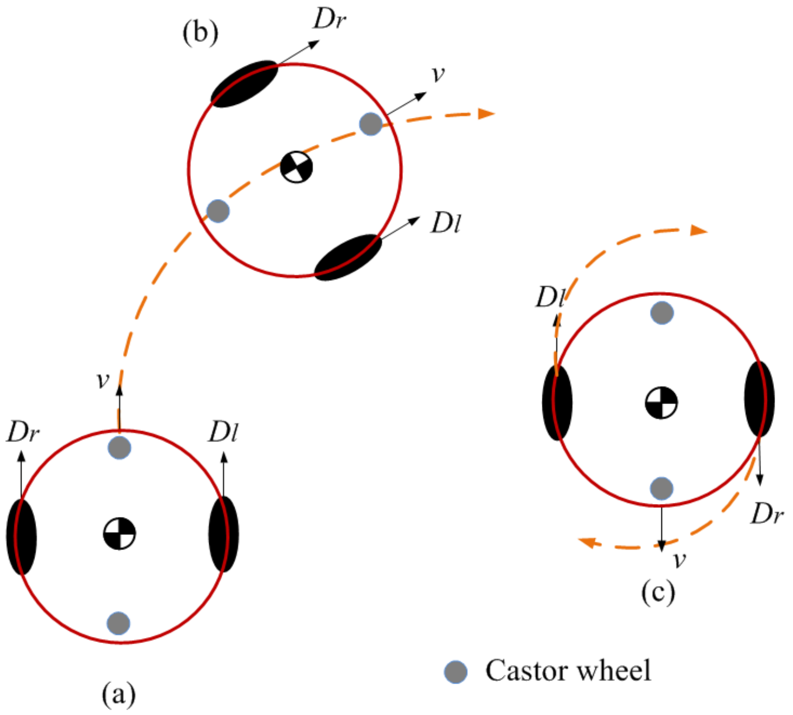


Figure 2.1: Wheel mobile robot. (a). The straight motion, where $D_r = D_l$, (b) The curve motion, where $D_r > D_l$, and (c) Robot turns about the midpoint of the two driving wheels, where where $D_r = -D_l$

straight forward or backward; if one wheel rotates faster than the other, the robot follows a curved path along the arc of the instantaneous circle. If both wheels rotate at the same velocity in opposite directions, the robot turns about the midpoint of the two driving wheels.

The proposed mobile robot is a two-wheeled differential drive mobile robot with non-holonomic constraints, as shown in Figure 2.1. Here, v is the translational velocity of the robot, and D_r and D_l are the driving force for both left and right wheels, respectively. In this chapter, both the kinematics and dynamics of applied mobile robot are discussed. The derived models were used in the controller design in this study. This chapter also details the experimental testbed, sensor configuration, and testbed parameters. The kinematics and dynamics modeling in this chapter are also presented in [82][84][103][112] [113].

2.2 Kinematics of Mobile Robot

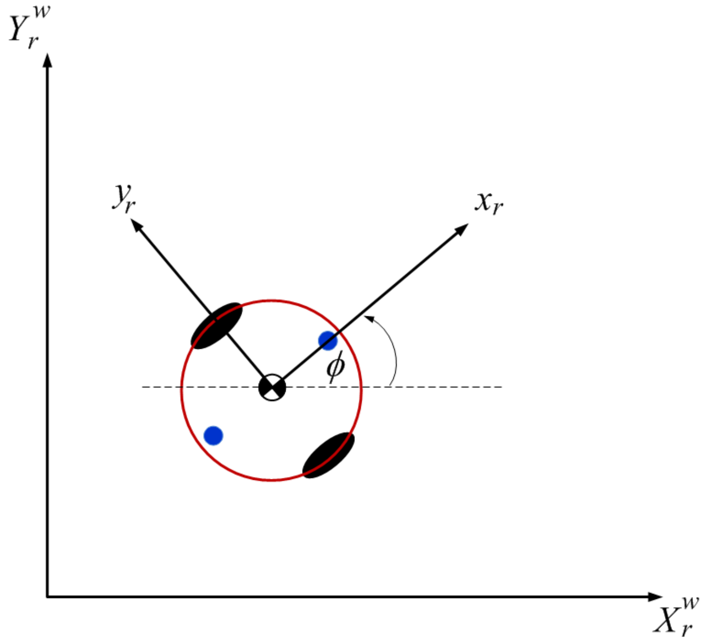
The kinematics of mobile robot is the study of how mechanical system behaves and how the robot deals with the configuration of robot in its workspace [5][7] [65]. A kinematic study is necessary to design the mechanical behavior of the robot and thereby determine appropriate mobile robot tasks and control software for the mobile robot hardware. By studying the mobile robot workspace, the range of possible poses of the robot in its environment can be identified. Mobile robot controllability defines the possible paths and trajectories in the robot's workspace. Unlike arm robot manipulators, mobile robots need position estimation as they are not end-fixed to the applied environment. A mobile robot is a self-contained automaton that can wholly move with respect to its environment. The instantaneous position of a mobile robot cannot be directly measured; instead, the robot's motion must be integrated over time.

To understand the motions of this type of robot, it is first necessary to determine the contribution of each wheel to the motion. Each wheel plays a role in the movement of the whole robot, but imposes constraints on that same motion. This section expresses the robot motion in the global and local robot reference frames (see Figure 2.2). Here, x_r^w and y_r^w constitute the global reference coordinate frame, and y_r and z_r describe the local reference coordinate frame, ϕ is the robot orientation angle (also defined as the angular difference between the global and local reference frames), and ω is the angular velocity of the robot.

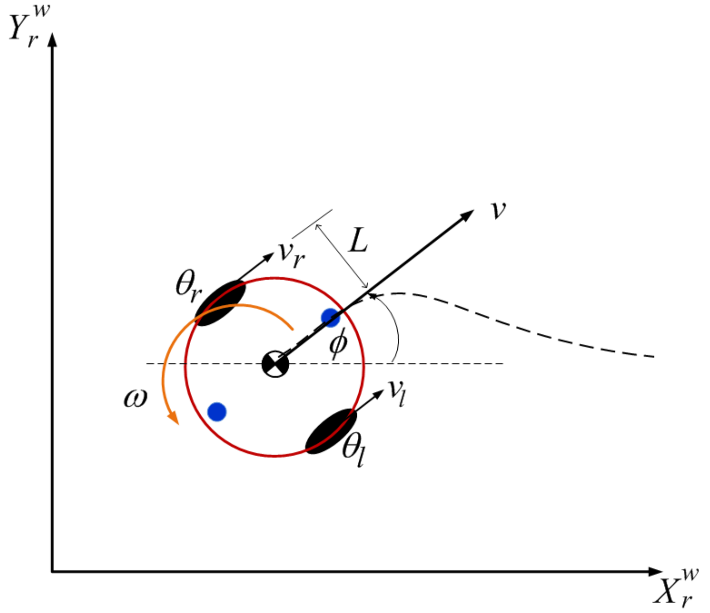
This section also discusses the construction of the robot's forward kinematic model of motion; that is, the movement of the robot as a function of its geometry and individual wheel behavior. Next, we describe the kinematic constraints on the whole robot and define the maneuverability of the robot. The robot is kinematically modeled as a rigid body on wheels, operating on a horizontal plane. The robot chassis on the plane has three dimensions; two for positioning in the plane and one for orientation along the vertical axis, which is orthogonal to the plane.

The motion in the global reference frame (x_r^w, y_r^w) is mapped to the motion in the local reference frame (x_r, y_r) by the orthogonal rotation matrix $(R_w(\phi))$, given as

$$R_w(\phi) = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$



(a) The global reference frame and the local robot reference frame.



(b) A differential drive robot in its global reference frame

Figure 2.2: The representation of robot's coordinate frame

The pose (position and orientation) and angular velocity (ω) of the mobile robot in Figure 2.2b are respectively given by

$$P_r = \begin{bmatrix} x_r \\ y_r \\ \phi \end{bmatrix}, \quad \dot{P}_r = \begin{bmatrix} \dot{x_r} \\ \dot{y_r} \\ \dot{\phi} \end{bmatrix} \tag{2.2}$$

where P_r is the pose of robot, $\dot{P}_r$ is the velocity of the robot, and $\dot{\phi} = \omega$. Therefore the relation between the pose in global reference frame P_r^w and in local reference frame is given by

$$\dot{P}_r = R_w(\phi)\ \dot{P}_r^w \tag{2.3}$$

Suppose that the robot's local reference frame is aligned such that the robot advances by $+X_r$, as shown in Figure 2.2b. We first consider the contribution of the rotational speed to the translational speed of each wheel. Assuming that the wheels roll without slipping, the angular positions and velocities of the left and right wheels are $\left(\theta_l, \dot{\theta}_l\right)$ and $\left(\theta_r, \dot{\theta}_r\right)$ respectively. We let v_l and v_r be the linear velocity of the left and right wheels, respectively, and v be the velocity of the robot. From Figure 2.2b, we get

$$v_l = v + L\dot{\phi}, \qquad v_r = v - L\dot{\phi} \tag{2.4}$$

where is L is the half width of the robot.

Adding and subtracting v_l and v_r in Eq. (2.4) we get

$$v = \frac{1}{2}(v_r + v_l), \qquad 2L\dot{\phi} = (v_l - v_r) \tag{2.5}$$

where, under the non-slippage assumption, we have

$$\dot{x} = R\dot{\theta}\cos\phi, \quad \dot{y} = R\dot{\theta}\sin\phi \tag{2.6}$$

where R is the wheel radius.

As $v_r = R\dot{\theta}_r$ and $v_l = R\dot{\theta}_l$, the axial velocities of the left and right wheels are respectively given by:

$$\dot{x} = v\cos\phi \qquad \dot{y} = v\sin\phi \tag{2.7}$$

Substituting Eqs. (2.7) and (2.6) into Eq. (2.5), the kinematics model of mobile robot in Figure 2.4 can be described by the following::

$$\begin{aligned}\dot{x} &= \frac{R}{2}\left(\dot{\theta}_l \cos\phi + \dot{\theta}_r \cos\phi\right)\\ \dot{y} &= \frac{R}{2}\left(\dot{\theta}_l \sin\phi + \dot{\theta}_r \sin\phi\right)\\ \dot{\phi} &= \frac{R}{2L}\left(\dot{\theta}_l - \dot{\theta}_r\right)\end{aligned} \tag{2.8}$$

Analogously to Eq. (2.2), the kinematics model can be written in matrix form:

$$\dot{P}_r = \begin{bmatrix} \frac{R}{2}\cos\phi \\ \frac{R}{2}\sin\phi \\ \frac{R}{2L} \end{bmatrix} \dot{\theta}_l + \begin{bmatrix} \frac{R}{2}\cos\phi \\ \frac{R}{2}\sin\phi \\ -\frac{R}{2L} \end{bmatrix} \dot{\theta}_r \tag{2.9}$$

or

$$\dot{P}_r = J\dot{\theta} \tag{2.10}$$

where

$$\dot{P}_r = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}, \quad \dot{\theta} = \begin{bmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{bmatrix} \tag{2.11}$$

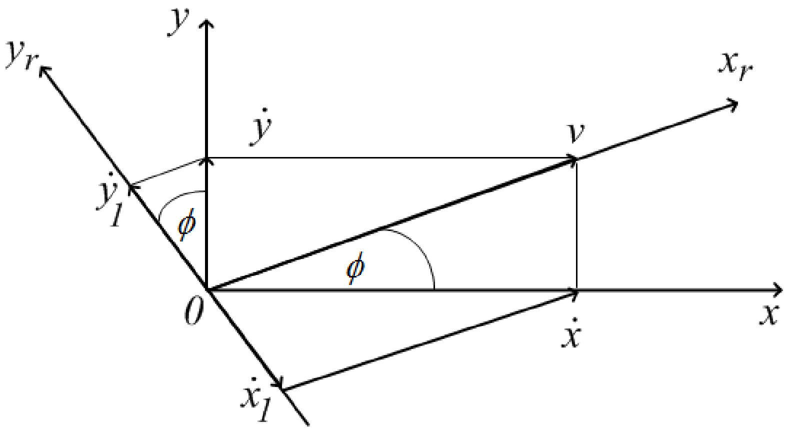


Figure 2.3: Nonholonomic constraint diagram

and J is the robot's Jacobian matrix, given by

$$J = \begin{bmatrix} \frac{R}{2}\cos\phi & \frac{R}{2}\cos\phi \\ \frac{R}{2}\sin\phi & \frac{R}{2}\sin\phi \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \tag{2.12}$$

The two 3-dimensional vectors field are:

$$g_1 = \begin{bmatrix} \frac{R}{2}\cos\phi \\ \frac{R}{2}\sin\phi \\ \frac{R}{2L} \end{bmatrix}, \quad g_2 = \begin{bmatrix} \frac{R}{2}\cos\phi \\ \frac{R}{2}\sin\phi \\ -\frac{R}{2L} \end{bmatrix} \tag{2.13}$$

Fields g_1 and g_2 allow rotations of the right and left wheels, respectively. Eliminating v in Eq. (2.7), can obtain the non-holonomic constraint:

$$\dot{x}\cos\phi - \dot{y}\sin\phi = 0 \tag{2.14}$$

which expresses the movement of the robot along the X_r axis. As shown in Figure 2.3, the velocity along the Y_r axis is zero (no lateral motion); that is,

$$\dot{x}_1 - \dot{y}_1 = 0 \tag{2.15}$$

where $\dot{x}_1 = \dot{x}\cos\phi$ and $\dot{y}_1 = \dot{y}\sin\phi$.

The Jacobian matrix (J) in Eq.(2.12) has three rows and two columns, so is non-invertible. Therefore, solving Eq. (2.10) for $\dot{P}_r$ we obtain

$$\dot{\theta} = J^{*}\dot{P}_r \tag{2.16}$$

where J^{*} is the generalized inverse of J given by

$$J^{*} = \left(J^T J\right)^{-1} J^T \tag{2.17}$$

Here, J^{*} can be computed directly from Eq. (2.4), and from Figure 2.3 we observe that

$$v = \dot{x}\cos\phi + \dot{y}\sin\phi \tag{2.18}$$

Thus, by Eq. (2.4) we obtain:

$$\begin{aligned} R\dot{\theta}_l &= \dot{x}\cos\phi + \dot{y}\sin\phi + L\dot{\phi} \\ R\dot{\theta}_r &= \dot{x}\cos\phi + \dot{y}\sin\phi - L\dot{\phi} \end{aligned} \tag{2.19}$$

$$\begin{bmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{bmatrix} = \frac{1}{R} \begin{bmatrix} \cos\phi & \sin\phi & L \\ \cos\phi & \sin\phi & -L \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \tag{2.20}$$

or $\dot{\theta} = J^* \dot{P}_r$, where:

$$J^* = \frac{1}{R} \begin{bmatrix} \cos\phi & \sin\phi & L \\ \cos\phi & \sin\phi & -L \end{bmatrix} \tag{2.21}$$

Finally, the kinematic relation can be written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \end{bmatrix} \tag{2.22}$$

2.3 Dynamics of Mobile Robot

After modeling the kinematics of the mobile robot, we performed the dynamic modeling, which is presented herein. Dynamic modeling is based on the laws of mechanics and involves three physical elements: inertia, elasticity, and friction, which exist in any real mechanical system [5][7]. The mobile robot in this study was subject to non-holonomic constraints that required particular treatment. Longitudinal and lateral slippage in the wheel movement commonly degrade the stability and control of WMRs. The dynamics of the present mobile robot shown in Figure 2.4 are given by

$$\begin{aligned} I\ddot{\phi} &= (D_r - D_l)L \\ M\dot{v} &= D_r + D_l \end{aligned} \tag{2.23}$$

where I is the moment of inertia of the robot around its center of gravity, M is the mass of the robot, and $\dot{v}$ and $\ddot{\phi}$ are the translational and angular accelerations of the robot respectively. The other symbols were defined in the previous section.

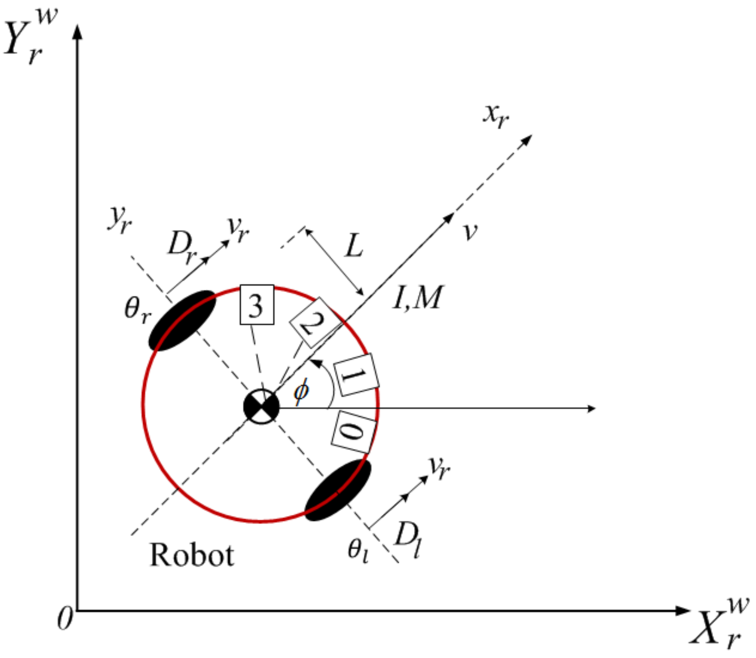


Figure 2.4: Two-wheeled mobile robot

The translational and angular velocity (v and $\dot{\phi}$, respectively), and acceleration ($\dot{v}$ and $\ddot{\phi}$, respectively) in Eq. (2.23) are given by the following equations:

$$\begin{aligned} v &= \frac{R}{2}\left(\dot{\theta}_r + \dot{\theta}_l\right) \quad \dot{v} = \frac{R}{2}\left(\ddot{\theta}_r + \ddot{\theta}_l\right) \\ \dot{\phi} &= \frac{R}{2L}\left(\dot{\theta}_r - \dot{\theta}_l\right) \quad \ddot{\phi} = \frac{R}{2L}\left(\ddot{\theta}_r - \ddot{\theta}_l\right) \end{aligned} \tag{2.24}$$

where $\ddot{\theta}_r$ and $\ddot{\theta}_l$ are the angular acceleration of the right and left wheels respectively.

The motor dynamics of the right and left wheels of the applied robot (see Figure 2.4) are given by

$$\begin{aligned} I_{mt}\ddot{\theta}_r + C\dot{\theta}_r &= KV_r - RD_r \\ I_{mt}\ddot{\theta}_l + C\dot{\theta}_l &= KV_l - RD_l \end{aligned} \tag{2.25}$$

where I_{mt} is the moment of inertia of motors, C is the damping coefficient and K is the driving gain of motor. V_r and V_l are the input voltage to the left and right wheels respectively.

By rewriting the dynamics of mobile robot in Eq. (2.23), we have

$$
\begin{aligned}
D_r &= \frac{R}{2}\left(M\dot{v} + \frac{I}{L}\ddot{\phi}\right) \\
D_l &= \frac{R}{2}\left(M\dot{v} - \frac{I}{L}\ddot{\phi}\right)
\end{aligned}
\tag{2.26}
$$

and rearranging Eq. (2.24), we obtain

$$
\begin{aligned}
\dot{\theta}_r &= \frac{1}{R}\left(v + L\dot{\phi}\right) \quad & \ddot{\theta}_r &= \frac{1}{R}\left(\dot{v} + L\ddot{\phi}\right) \\
\dot{\theta}_l &= \frac{1}{R}\left(v - L\dot{\phi}\right) \quad & \ddot{\theta}_l &= \frac{1}{R}\left(\dot{v} - L\ddot{\phi}\right)
\end{aligned}
\tag{2.27}
$$

Substituting Eq. (2.26) and Eq. (2.27) into Eq. (2.25), we have

$$
\begin{aligned}
\frac{I_{mt}}{R}\left(\dot{v} + L\ddot{\phi}\right) + \frac{C}{R}\left(v + L\dot{\phi}\right) &= KV_r - \frac{R}{2}\left(M\dot{v} + \frac{I}{L}\ddot{\phi}\right) \\
\frac{I_{mt}}{R}\left(\dot{v} - L\ddot{\phi}\right) + \frac{C}{R}\left(v - L\dot{\phi}\right) &= KV_l - \frac{R}{2}\left(M\dot{v} - \frac{I}{L}\ddot{\phi}\right)
\end{aligned}
\tag{2.28}
$$

and Eq. (2.28) can be rearranged as

$$
\begin{aligned}
\left(\frac{R}{2}M + \frac{I_{mt}}{R}\right)\dot{v} + \left(\frac{I_{mt}}{R}L + \frac{IR}{2L}\right)\ddot{\phi} &= Kv_r - \frac{C}{R}v - \frac{C}{R}L\dot{\phi} \\
\left(\frac{R}{2}M + \frac{I_{mt}}{R}\right)\dot{v} - \left(\frac{I_{mt}}{R}L + \frac{IR}{2L}\right)\ddot{\phi} &= Kv_r - \frac{C}{R}v + \frac{C}{R}L\dot{\phi}
\end{aligned}
\tag{2.29}
$$

The dynamics form of Eq. (2.29) is rewritten as

$$
\begin{bmatrix} a_1 & a_2 \\ a_1 & -a_2 \end{bmatrix}
\begin{bmatrix} \dot{v} \\ \ddot{\phi} \end{bmatrix}
=
\begin{bmatrix} -a_3 & -a_3L \\ -a_3 & a_2L \end{bmatrix}
\begin{bmatrix} v \\ \dot{\phi} \end{bmatrix}
+ K \begin{bmatrix} V_r \\ V_l \end{bmatrix}
\tag{2.30}
$$

where

$$
a_1 = \frac{R}{2}M + \frac{I_{mt}}{R} \qquad a_2 = \frac{I_{mt}}{R}L + \frac{IR}{2L} \qquad a_3 = \frac{C}{R}
$$

Therefore the translational and angular acceleration ($\dot{v}$ and $\ddot{\phi}$, respectively) can be derived from Eq.(2.30) as

$$\begin{bmatrix} \dot{v} \\ \ddot{\phi} \end{bmatrix} = \frac{1}{-2a_1a_2} \begin{bmatrix} -a_2 & -a_2 \\ -a_1 & a_1 \end{bmatrix} \begin{bmatrix} -a_3 & -a_3L \\ -a_3 & a_2L \end{bmatrix} \begin{bmatrix} v \\ \dot{\phi} \end{bmatrix} - \frac{K}{2a_1a_2} \begin{bmatrix} -a_2 & -a_2 \\ -a_1 & a_1 \end{bmatrix} \begin{bmatrix} V_r \\ V_l \end{bmatrix} \tag{2.31}$$

Finally, by rearranging Eq. (2.31), we get the following state space dynamic model of the applied WMR's motion:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.32}$$

By substituting and rearranging Eq. (2.31) into the generic form in Eq. (2.32), we can derive the following state space dynamic model of the applied system:

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} v \\ \phi \\ \dot{\phi} \end{bmatrix} &= \begin{bmatrix} \alpha_1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} v \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} \alpha_3 & \alpha_3 \\ 0 & 0 \\ \alpha_4 & \alpha_4 \end{bmatrix} \begin{bmatrix} V_r \\ V_l \end{bmatrix} \\ y &= \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ \phi \\ \dot{\phi} \end{bmatrix} \end{aligned} \tag{2.33}$$

where

$$\begin{aligned} \alpha_1 &= -\frac{a_3}{a_1} = -\frac{\frac{C}{R}}{\frac{R}{2}M + \frac{I_{mt}}{R}} = -\frac{2C}{MR^2 + 2I_{mt}} \\ \alpha_2 &= -\frac{a3}{a2} = -\frac{\frac{C}{R}L}{\frac{R}{2}M + \frac{I_{mt}}{R}} = -\frac{2CL^2}{2I_{mt}L^2 + IR^2} \\ \alpha_3 &= \frac{K}{2a_1} = \frac{K}{2\left(\frac{R}{2}M + \frac{I_{mt}}{R}\right)} = \frac{K}{MR + 2\frac{I_{mt}}{R}} = \frac{KR}{MR^2 + 2I_{mt}} \\ \alpha_4 &= \frac{K}{2a_2} = \frac{K}{2\left(\frac{I_{mt}}{R}L + \frac{IR}{2L}\right)} = \frac{K}{2\frac{I_{mt}}{R}K + \frac{IR}{L}} = \frac{KLR}{2I_{mt}L^2 + IR^2} \end{aligned}$$

2.4 Experimental Setup

The proposed method was evaluated on an experimental testbed designed for that purpose. An overview of the mobile robot experimental system is provided in Figure 2.5. The robot was connected to a personal computer (PC) installed with an analog input board, a counter board, and an analog output board, as shown in Figure 2.6. The PC acted as a controller to these three boards, generating the input and output signals.

The command input voltage was sent to the motor driver via the 12-bit analog output board, and an armature current was generated in each motor. Moreover, the rotation angle of each tire was detected by an encoder board installed in the DC servo motor, and input to the computer through the 24-bit counter board. The data given by the distance sensor were processed by an analog-to-digital (AD) board, creating reference trajectories for the mobile robot. These data were continually updated online. The target to be intercepted and followed by the robot was detected by the Kinect sensor. The target position and orientation data were processed by the PC and used as the input data to move the motors of the robot and to compare the distance between the target and the robot. In this way, the following system was continuously maintained.

Figure 2.7 shows the non-holonomic mobile robot used in this study. The robot had two independently controlled tires powered by two DC servo motors equipped with incremental encoders and was balanced by two castor wheels. The reference trajectory for obstacle avoidance was created by four IR proximity sensors, and the Kinect sensor for tracking and following the moving object was mounted on the robot.

Panels 2.8a and 2.8b of Figure 2.8 show the power sources used in the experimental system and the drivers for the left and right motors, respectively. The case in Figure 2.8b encloses the voltage–current conversion circuit, whose main component is the power operational amplifier. The DC power is input to the motor driver. The motor driver adjusts the current, which is proportional to the voltage output from the digital analog board, flowing to the DC servo motor. Tables 2.1 - 2.5 list the parameters of the experimental system.

The experimental environment was mostly confined to the laboratory but was later extended to the hallway. The experimental environments were altered to fit the various experiment objectives, and are specified in the next chapter.

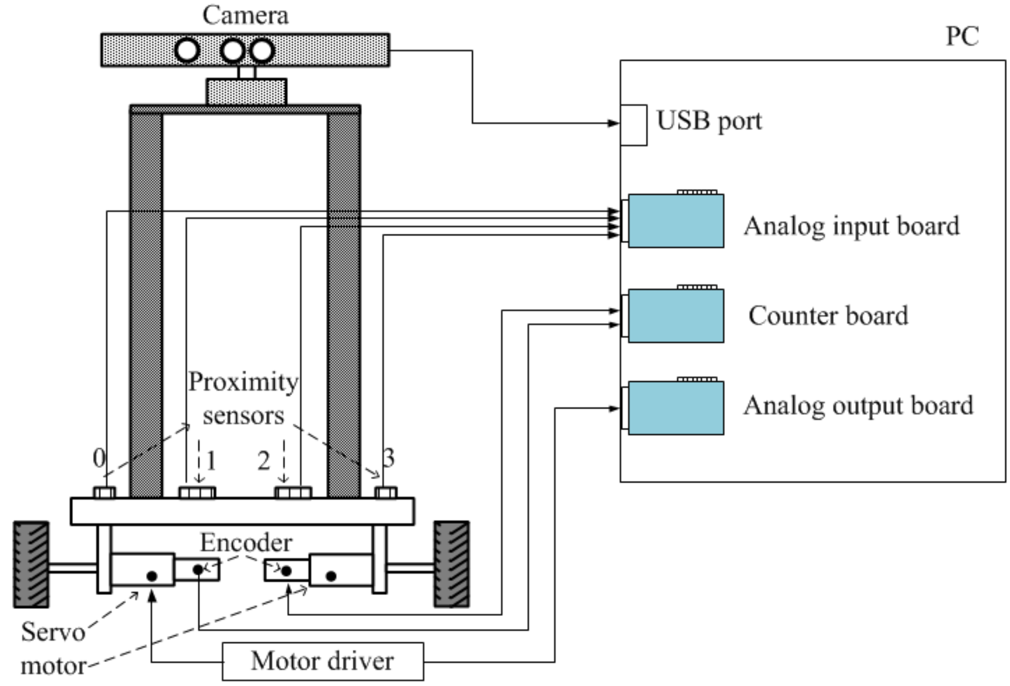


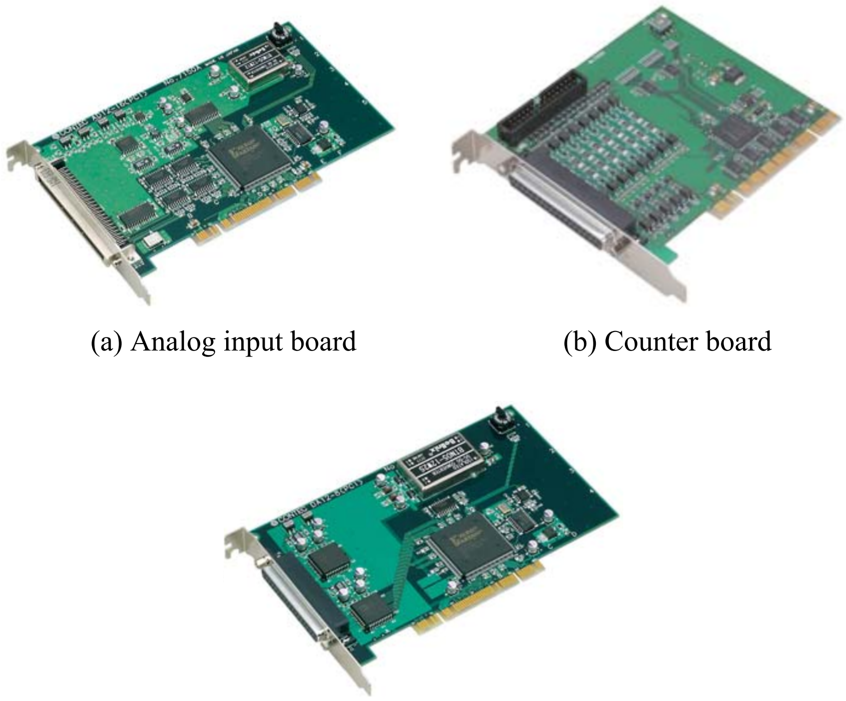
Figure 2.5: Overview of the experimental system

Table 2.1: Components of experimental testbed

Components	Type	Manufacturer
DC servo motors	RH-14D 6002	Harmonic drive LLC
AD converter board	AD 12-16 (PCI)	CONTEC
Counter board	CNT24-4 (PCI)H	CONTEC
DA converter board	DA 12-8(PCI)	CONTEC
IR Proximity Sensors	2Y0A21	SHARP
Vision Sensor	Kinect XBOX 360	Microsoft
PC (0S)	Intel(R) Pentium(R) CPU G630 @ 2.70GHz Windows-7 64 bits	Microsoft

Table 2.2: Parameters of mobile robot

Symbol	Parameters	Value (unit)
M	Mass	8.895 (kg)
I	Moment of inertia robot	0.18 ($\text{kg} \cdot \text{m}^2$)
R	Wheels radius	0.05(m)
L	Radius of mobile robot to the center of gravity	0.17 (m)

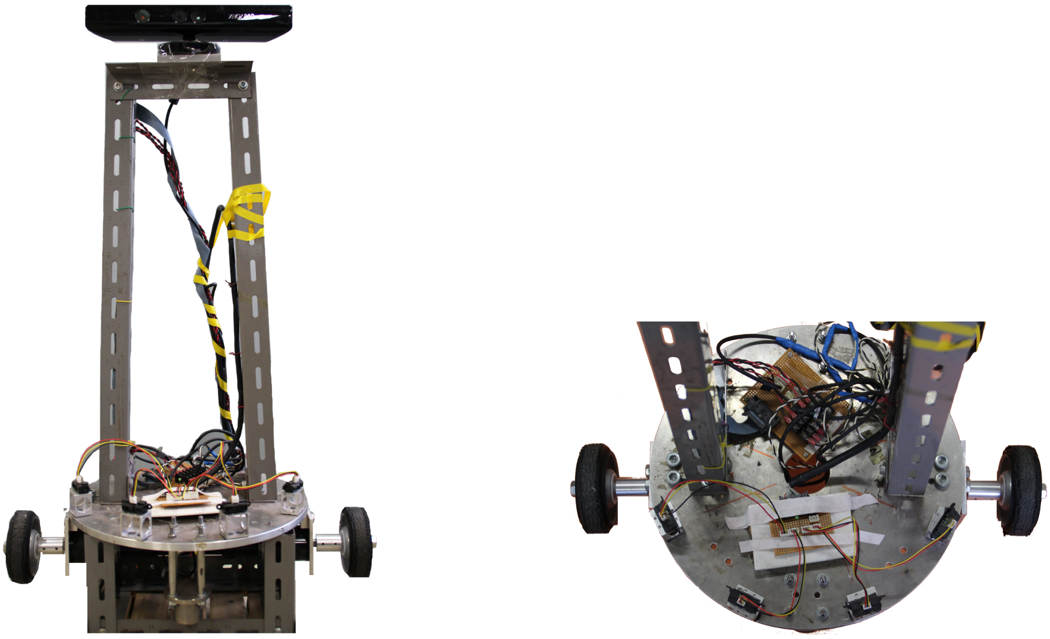


(a) Analog input board

(b) Counter board

(c) Analog output

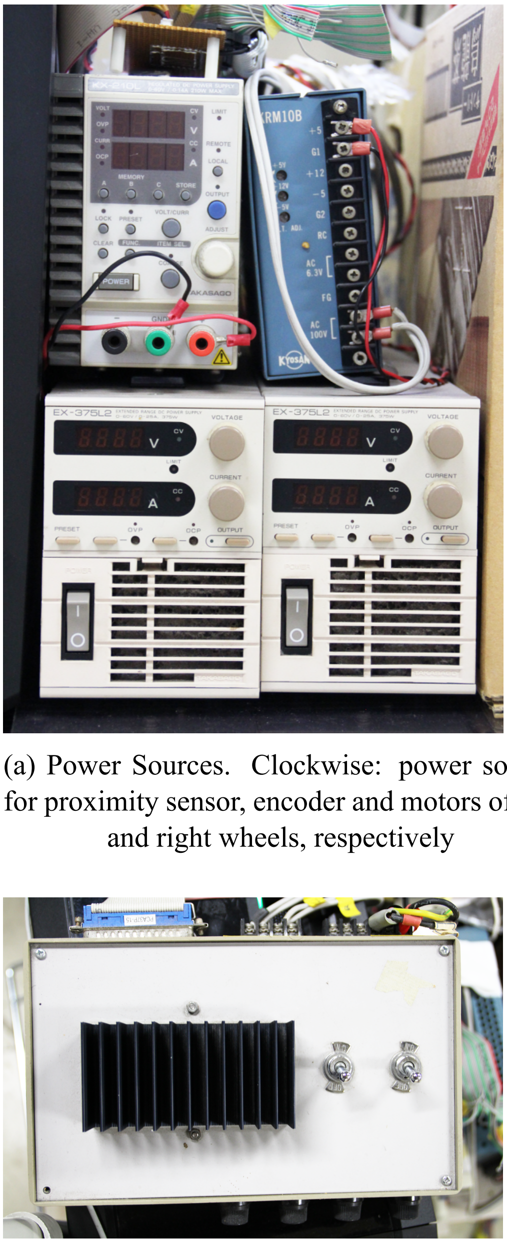
Figure 2.6: Installed board in PC



(a) Wheeled Mobile Robot

(b) Upper view of proximity sensors arrangement

Figure 2.7: Two wheeled mobile robot equipped with proximity sensors and vision sensor



(a) Power Sources. Clockwise: power source for proximity sensor, encoder and motors of left and right wheels, respectively

(b) Upper view of proximity sensors arrangement

Figure 2.8: The representation of robot's coordinate frame

Table 2.3: Parameters of IR proximity sensor

Symbol	Parameters	Value (unit)
V_{cc}	Supply voltage	-0.3 to $+7$ (V)
V_o	Output terminal voltage	-0.3 to $+0.3(V)$
T_{opr}	Operating temperature	-10 to $+60(^0C)$
T_{stg}	Storage temperature	-40 to $+70(^0C)$

Table 2.4: Parameters of Motor

Parameters	Value (unit)
Mass	0.77 (kg)
Maximum momentary torque	$14(\mathrm{N}\cdot\mathrm{m})$
Rated torque	$3.2((\mathrm{N}\cdot\mathrm{m}))$
Maximal rotational speed	$100\ (\mathrm{r/min})$
Rated rotational speed	$60(\mathrm{r/min})$

Table 2.5: Parameters of Kinect Sensor

Parameters	Value (unit)
Horizontal field of view	57 Degrees
Vertical field of view	43 Degrees
Depth sensor range	1.2 m - 3.5 m
Data stream	$320x24016$ bit depth @30 frames/sec

Chapter 3

Collision Avoidance Control for Mobile Robots in a Human Living Environments

3.1 Introduction

Designing the perfect navigation for the mobile robot assigned in human-living environment is one of the most important tasks. Navigation is finding a viable way between a starting point and a goal point, knowing which path is the most efficient with the most effective way. The process of finding the most effective way is typically divide into two approaches, the deliberative and the reactive approaches [66][67].

In the deliberative approach, a path through a map of the world is constructed, followed by the execution of the plan by the robot. The deliberative controllers require high computational power and have slow response time. The reactive controllers respond fast but suffer from the restrictions of not having a global picture of the environment and the path planning task.

One of the reactive approaches is the navigation design based on potential field, originally proposed by [38]. The basic idea is to guide the robot by defining attractive and repulsive forces representing goal and obstacle respectively[39][40][68]. To design an effective motion trajectory, the position control is most likely to be found in application of service mobile robot in which the robot is treated essentially as an isolated system. However, to be applied in human living environment, robot system has to consider the dynamic interaction with the environment [69]. Obstacle avoidance is one of the main functions required in autonomous robot navigation [5][70]. The effectiveness of obstacle avoidance will be improved by installing some sensors simultaneously instead of using only a single sensor. The installment of some sensors and other components in a system is called system integration.

System integration is defined as the process of bringing together the component subsystems into one system and ensuring that the subsystems function together as a system. One of the example of applying system integration by installing some sensors with different characteristics that creates a combination of sensory data that has an inherent redundancy and may provide robust recognition of robot working environment [16][17][71]. The applied sensors in a mobile robot can give partial knowledge on its environment and goal position(s) to encompass the robot reaching its goal position(s) as efficiently and reliable as possible [7].

One of the application of mobile robot is a service robot [24][66] [72]-[75] because it is expected to work in a dynamic environment such as a restaurant environment [2] [76], sport environment [77] and museum environment [4][78]. The application of mobile robot as a service robot can be semi-automatic [79][80] or fully automatic [81].

Swarm robotics is a relative new field of research but since the first pioneering work by [43], who simulated a flock of bird in flight and since then this field has been through many developments applying different approaches for swarm aggregation, navigation, coordination and control. The swarm intelligence paradigm has proven to have very interesting properties such as robustness, flexibility and ability to solve complex problems exploiting parallelism and self-organization. There have been increasing interests in deploying a team of robots, or swarm robots, to fulfill certain complicated tasks. Since swarm robots may accomplish task faster than single robot.

This chapter deals with collision avoidance scheme achieved by model reference control that based on the potential field path planner. The designed controller is used to design a service mobile robot moving in dynamic environment consisting of moving obstacles. This chapter also presents the collision avoidance control for swarm robots moving in a rehabilitation environment that consists of static obstacle and moving obstacles. A method is presented for system integration of sensors to achieve motion in the dynamic environment. The proposed method combines the information obtained by several proximity sensors, an image sensor, and a localization sensors (RFID system). This study applies a leader-follower formation in which the swarm robot team has a leader robot that follows the rehabilitee, while the other robots follow the leader robot. The robot controllers comprise a reference and PI controller. The reference controller generates a robot motion trajectory by referring to sensor information in real-time, and the PI controller makes the robots follow the generated motion trajectory. Various simulation results, which assume the presence of several static and dynamic obstacles in the human living environment, demonstrate the effectiveness of the proposed design. This chapter relates to our work presented in [82] [103].

3.2 Potential Field Method

As the robot navigates in an environment and detects obstacles, it should know how to react to the identified obstacles and knows which direction to move away from the obstacles. Most of the obstacle avoidances algorithms are designed to work in environments that are known to the robot in advance, also called global path planning. However, in the application of the robot that moves around in unknown environment, the algorithm must be able to recalculate the path planning online based on limited sensor information, or local path-planning. One algorithm that can be used for both known and unknown environments is the artificial potential method.

Artificial potential fields for autonomous robot navigation were first proposed by [38]. The main idea is to generate attraction and repulsion forces within the working environment of the robot to guide it to the target. The target point has an attractive influence on the robot and each obstacle tends to push away the robot, in order to avoid collisions. Potential field methods provide an elegant solution to the path finding problem. Since the path is the result of the interaction force fields, the path finding problem becomes a search for optimum filed configuration instead of the direct construction of an optimum path.

The potential field approach has been used extensively for mobile robot path planning due to its ability to find simple mathematical and computational solution. The basic concept of the potential field method is by applying artificial potential field in robot workspace where the goal exerts an attractive force on the robot and the obstacles give repulsive force. The robot is assumed to be a point of mass and moves in a two-dimensional workspace from a high potential point to a low potential point and the applied potential force applied to the robot is updated based on the position of the robot relative to the goal and/or to the obstacles [39] [40] [58][68] [83].

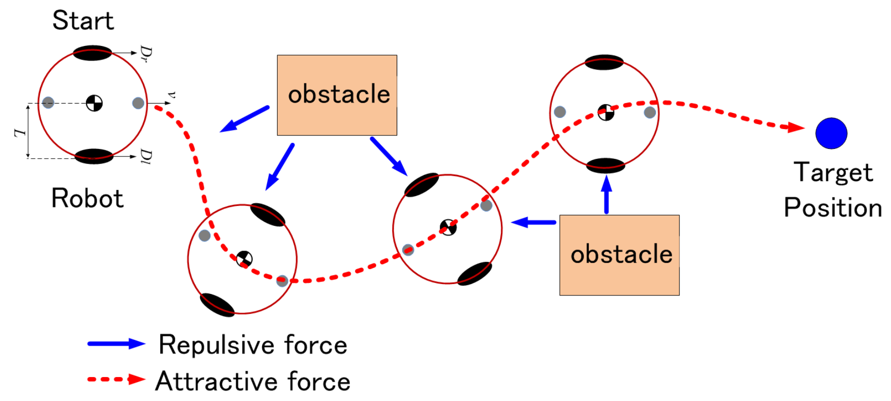


Figure 3.1: Basic principle of artificial potential field method

The potential fields method for autonomous mobile robot navigation basically based on the assignment of an attractive potential field to the goal point and a repulsive potential to each of the obstacles in the environment. The most simple implementation is by considering a known environment where fixed potentials can be assigned to the goal and the obstacles. When the obstacles are unknown, the potential fields have to be adapted as the robot advances and detects the new obstacles.

The potential field function can be constructed as the sum of the attractive and repulsive potential as below:

$$F_{total}\left(P_r^c\right) = F_{att}\left(P_r^c\right) + F_{rep}\left(P_r^c\right) \tag{3.1}$$

where $F_{total}\left(P_r^c\right)$ is the potential field force, $F_{att}\left(P_r^c\right)$ is the attractive potential field force, $F_{rep}\left(P_r^c\right)$ is the repulsive potential field force, and P_r^c is the robot position in camera coordinates frame.

3.2.1 Attractive Potential Field

The attractive potential $F_{att}\left(P_r^c\right)$ monotonously increases with distance from target position P_m^c [7][9]. The simplest choice for this type of behavior is the conic potential, measuring a scaled distance to the goal, $F_{att}\left(P_r^c\right) = \kappa d_{act}\left(P_r^c, P_m^c\right)$, where κ is the positive constant, d_{act} is the distance between the robot and the target and P_m^c is the target position in camera coordinates frame. The attractive gradient is given by

$$\nabla F_{att}\left(P_r^c\right) = \frac{\kappa}{d_{act}\left(P_r^c, P_m^c\right)}\left(P_r^c - P_m^c\right) \tag{3.2}$$

The gradient vector points away from the target with magnitude κ at all points in the configuration space of the robot except the target, where it is undefined. If the robot starts at any point other than the target, by following the negative gradient, it will follow a path towards the goal. In order to avoid the possibility of "chattering" problem due to the discontinuity in the attractive gradient at the origin, a potential function that grows quadratically with distance is normally used, e.g.,

$$F_{att}\left(P_r^c\right) = \frac{1}{2}\kappa d_{act}^2\left(P_r^c, P_m^c\right) \tag{3.3}$$

with the gradient

$$\begin{aligned}\nabla F_{att}\left(P_r^c\right) &= \nabla\left(\frac{1}{2}\kappa d_{act}^2\left(P_r^c, P_m^c\right)\right)\\ &= \frac{1}{2}\kappa\nabla d_{act}^2\left(P_r^c, P_m^c\right)\\ &= \kappa\left(P_r^c - P_m^c\right)\end{aligned} \tag{3.4}$$

which is a vector based at P_r^c, points away from P_m^c, and has a magnitude proportional to the distance from P_r^c to P_m^c. The farther away P_r^c is from P_m^c, the bigger the magnitude of the vector. In other words, when the robot is far away from the goal, the robot quickly approaches it; when the robot is close to the goal, the robot slowly approaches it [9].

In this study, we used attractive potential field for target setting. In chapter 6, this attractive potential field method is developed for visual tracking using vision-based sensor.

3.2.2 Repulsive Potential Field

The repulsive potential force keeps the robot away from the obstacles. The strength of the repulsive potential depends on how close the robot is to the obstacle. The closes the robot is, the stronger the repulsive force should be [7][9]. The repulsive potential is usually defined in terms of distance to the closest obstacle d, defined as:

$$F_{rep}\left(P_r^c\right) = \begin{cases}\frac{1}{2}\eta\left(\frac{1}{d} - \frac{1}{d_{max}}\right)^2, & d \leq d_{max}\\ 0, & d > d_{max}\end{cases} \tag{3.5}$$

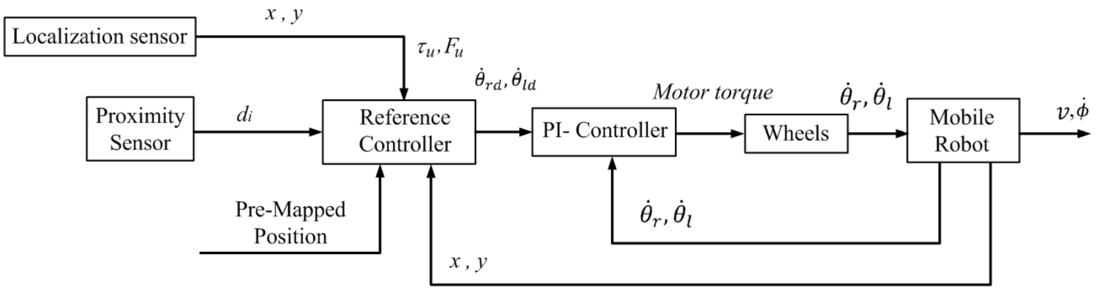


Figure 3.2: Block diagram of proposed method in chapter 3.

whose gradient is

$$\nabla F_{rep}\left(P_r^c\right) = \begin{cases} \eta\left(\frac{1}{d_{max}} - \frac{1}{d}\right)\frac{1}{d^2}\nabla d, & d \le d_{max} \\ 0, & d > d_{max} \end{cases} \tag{3.6}$$

where the factor d_{max} allows the robot to ignore obstacles that are sufficiently far away. And η can be viewed as a gain on the repulsive gradient.

By implementing this algorithm numerically, a path may form that oscillates around points that are equally distant in both directions from the obstacles. To avoid these oscillations, the repulsive potential can be defined in terms of distances to individual obstacles, instead of just the closest obstacle. The each obstacle has its own potential function as:

$$F_{rep_i}\left(P_r^c\right) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d_i} - \frac{1}{d_{i_{max}}}\right)^2, & d_i \le d_{i_{max}} \\ 0, & d_i > d_{i_{max}} \end{cases} \tag{3.7}$$

where $d_{i_{max}}$ defines as the maximum distance to obstacle i. Then the total repulsive field become $\sum_n^{i=1} F_{rep_i}\left(P_r^c\right)$.

Usually when the artificial potential field method is implemented, the robot is either given a pre-defined map of the environment and the potential fields are computed prior to driving. In other cases, when the map is not known, the robot is given an initial position and a goal position, the potential field is then continuously updated and recalculated as the robot encounters obstacles. The potential field concept heavily relies on the environment information provided by the sensor. Position sensor is required for attractive potential field for giving the position information and distance sensor is required for repulsive potential filed for giving distance information of the robot to the obstacles.

In this study, we develop the concept of repulsive potential field to be the reference controller to create a reference trajectory and apply Proportional Integral controller to move the robot toward the reference controller.

3.3 Reference Controller Design

The reference controller used in this study is derived by using potential field method concept. The environmental information required to create the reference trajectory is provided by integrating multiple sensors: a Kinect sensor, four proximity sensors and a position sensor system. We assume that the position sensor system indicates the position of target, that is indicated by the pre-defined point.

If there is no obstacle between the robot and the blue circle mark, the input torque and force for the robot in Figure 2.4, τ_u and F_u, are given as follows:

$$\begin{aligned} I\ddot{\phi}_d &= \tau_u \\ M\dot{v}_d &= F_u \end{aligned} \tag{3.8}$$

where $\ddot{\phi}_d$ and $\dot{v}_d$ are the desired angular and translational accelerations.

When the robot encounter obstacles in its workspace, the repulsion virtual torques and forces (τ_1, τ_2, F_1, and F_2) are generated. In an environment where obstacles are present, we use the following reference model to generate the reference trajectory.

$$\begin{aligned} I\ddot{\phi}_d + C_\phi \dot{\phi}_d &= \tau_u + \tau_1 + \tau_2 \\ M\dot{v}_d + C_v v_d &= F_u - F_1 - F_2 \end{aligned} \tag{3.9}$$

where C_ϕ and C_v are virtual damping coefficients for increasing stability, $\dot{\phi}_d$ and v_d are the desired angular and translational velocities of the robot generated by a reference controller, and τ_i and F_i are the virtual torques and forces to avoid collision (i = 1, 2)

The damping coefficients are not considered in Eq. (2.23) because they are usually very small. However, to ensure stable robot motion, C_ϕ and C_v are added virtually to the robot dynamics.

The reference trajectory is created by adding the virtual torques and forces (τ_1, τ_2, F_1, and F_2) to the dynamics of a mobile robot in Eq. (3.9). The virtual torques and forces are

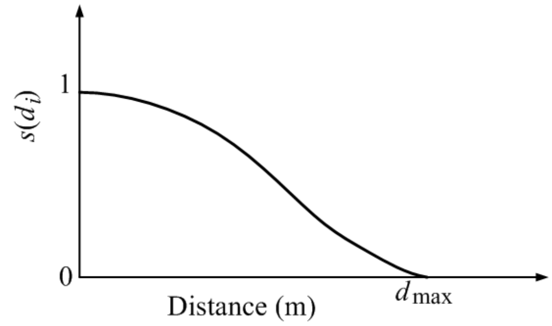


Figure 3.3: Shape function of the distance between proximity sensor and obstacle.

calculated by considering the combined data input from the position sensor and proximity sensors.

Torques τ_1 and τ_2 are designed for collision avoidance and to keep the robot parallel to the virtual wall of passages, respectively. The force F_1 gives the deceleration effect according to the distance of the robot to the obstacles, and F_2 is used to give the deceleration effect according to the approaching speed of dynamic obstacles. These four parameters used to adjust the magnitude of virtual external forces/torques, are calculated as below:

$$\tau_1 = \operatorname{sgn}(v)\,\alpha_1 \left(\sum_{i=0,1} s(d_i) - \sum_{i=2,3} s(d_i) \right) \tag{3.10}$$

$$\tau_2 = \alpha_2 \left(\sum_{i=0,1} s(d_i) - \sum_{i=2,3} s(d_i) \right) \tag{3.11}$$

$$F_1 = \beta_1 \left(\sum_{i=1,2} s(d_i) \right) \tag{3.12}$$

$$F_2 = \beta_2 \left(\sum_{i=1,2} \min\left(\dot{d}_i, 0\right) \right) \tag{3.13}$$

where d_i is the distance to the obstacles measured by ith sensor shown in Figure 2.4, $\operatorname{sgn}(v)$ is the sign function of the translational velocity of the robot, α_j and β_j are the adjustable constants to provide the effective collision avoidance (j = 1, 2), $\dot{d}_i$ is the time derivative of distance (d_i) to obstacles, which corresponds with the robot speed when approaching obstacles; and $s(d_i)$ is the shape function representing the relationship between the virtual external force and the distance to the obstacle.

The shape function is given as follows:

$$\begin{cases} s\left(d_i\right)=1, & \text{if } d_i<\bar{d} \\ s\left(d_i\right)=\frac{1}{a}\left[\exp\left\{-\frac{\left(d_i-\bar{d}\right)^2}{2\sigma^2}\right\}-b\right], & \text{if } \bar{d}<d_i<d_{max} \\ s\left(d_i\right)=1, & \text{if } d_i>d_{max} \end{cases}$$
$$b=\exp\left\{-\frac{\left(d_{\max}-\bar{d}\right)^2}{2\sigma^2}\right\}, \quad a=1-b \tag{3.14}$$

where σ and $\bar{d}$ are design parameters for defining the shape function of the virtual external force and d_{max} is the maximum distance that can be measured by a proximity sensor.

The virtual external torque and force increase as the distance to the obstacle decreases as shown in Figure 3.3. The virtual external force will be zero when the distance of the robot from the obstacle is greater than $d_{\max}$, which is a design parameter based on sensor specifications.

3.4 PI Controller Design

A PI controller is employed to move the robot on the desired trajectory. In this study, the PI control works by calculating the difference between the desired velocity and the actual velocity, as follows:

$$\begin{aligned} \tau_{mr} &= K_P\left(\dot{\theta}_{rd}-\dot{\theta}_r\right)+K_I\int\left(\dot{\theta}_{rd}-\dot{\theta}_r\right) \\ \tau_{ml} &= K_P\left(\dot{\theta}_{ld}-\dot{\theta}_l\right)+K_I\int\left(\dot{\theta}_{ld}-\dot{\theta}_l\right) \end{aligned} \tag{3.15}$$

where τ_{mr} and τ_{ml} are the motor torques for the right and left wheels, respectively, K_P is the proportional gain, and K_I is the integral gain.

The reference velocities ($\dot{\theta}_{rd}$, $\dot{\theta}_{ld}$) for the right and left wheels, respectively, are calculated by

$$\begin{aligned} \dot{\theta}_{rd} &= \frac{v_d+L\dot{\phi}_d}{R} \\ \dot{\theta}_{ld} &= \frac{v_d-L\dot{\phi}_d}{R} \end{aligned} \tag{3.16}$$

where v_d and $\dot{\phi}_d$ are the desired translational and angular velocities.

This section also discussed the derivation of PI Controller design for simulation. We apply again Eq. (2.25) to design the appropriate Proportional Integral (PI) controller for computer simulation to validate the effectiveness of the propose method.

Considering

$$\begin{aligned} KV_r &= K_P\left(\dot{\theta}_{rd} - \dot{\theta}_r\right) + K_I \int \left(\dot{\theta}_{rd} - \dot{\theta}_r\right) dt \\ KV_l &= K_P\left(\dot{\theta}_{ld} - \dot{\theta}_l\right) + K_I \int \left(\dot{\theta}_{ld} - \dot{\theta}_l\right) dt \end{aligned} \tag{3.17}$$

where K is motor gain.

By substituting Eq. (3.17) into Eq. (2.25), the following dynamics is obtained

$$\begin{aligned} I_{mt}\ddot{\theta}_r + C\dot{\theta}_r &= K_P\left(\dot{\theta}_{rd} - \dot{\theta}_r\right) + K_I \int \left(\dot{\theta}_{rd} - \dot{\theta}_r\right) dt - RD_r \\ I_{mt}\ddot{\theta}_l + C\dot{\theta}_l &= K_P\left(\dot{\theta}_{ld} - \dot{\theta}_l\right) + K_I \int \left(\dot{\theta}_{ld} - \dot{\theta}_l\right) dt - RD_l \end{aligned} \tag{3.18}$$

Substituting Eqs. (2.23) and (2.24) into Eq. (3.18) results in

$$\begin{aligned} \frac{I_{mt}}{R}\left(\dot{v} + L\ddot{\phi}\right) + \frac{C}{R}\left(v + L\dot{\phi}\right) =& \frac{K_P}{R}\left\{\left(v_d + L\dot{\phi}_d\right) - \left(v + L\dot{\phi}\right)\right\} \\ &+ K_I S_r - \frac{R}{2}\left\{M\dot{v} + \frac{I}{L}\ddot{\phi}\right\} \end{aligned} \tag{3.19}$$

where S_r is the integral of the tracking error of the right wheel given by

$$S_r = \int \left(\dot{\theta}_{rd} - \dot{\theta}_r\right) dt \tag{3.20}$$

Substituting Eq. (3.23) into Eq. (3.19), we have the following dynamics for the right wheel

$$\begin{aligned} &\left(\frac{I_{mt}}{R} + \frac{MR}{2}\right)\dot{v} + \left(\frac{I_{mt}L}{R} + \frac{IR}{2L}\right)\ddot{\phi} \\ &\quad = \frac{1}{R}\left(C + K_P\right)v - \frac{L}{R}\left(C + K_P\right)\dot{\phi} + K_I S_r + \frac{K_P}{R}v_d + \frac{K_P}{R}L\dot{\phi}_d \end{aligned} \tag{3.21}$$

Similarly, we have the following dynamics for the left wheel

$$\begin{aligned}&\left(\frac{I_{mt}}{R}+\frac{MR}{2}\right)\dot{v}-\left(\frac{I_{mt}L}{R}+\frac{IR}{2L}\right)\ddot{\phi}\\&\qquad=-\frac{1}{R}\left(C+K_P\right)v+\frac{L}{R}\left(C+K_P\right)\dot{\phi}+K_IS_l+\frac{K_P}{R}v_d-\frac{K_P}{R}L\dot{\phi}_d\end{aligned}\tag{3.22}$$

since S_l is given by

$$S_l=\int\left(\dot{\theta}_{ld}-\dot{\theta}_l\right)dt\tag{3.23}$$

Eq. (3.21) and Eq. (3.22) give

$$\begin{aligned}&\begin{bmatrix}\frac{I_{mt}}{R}+\frac{MR}{2} & \frac{I_{mt}L}{R}+\frac{IR}{2}\\ \frac{I_{mt}}{R}+\frac{MR}{2} & -\left(\frac{I_{mt}L}{R}+\frac{IR}{2}\right)\end{bmatrix}\begin{bmatrix}\dot{v}\\ \ddot{\phi}\end{bmatrix}=\begin{bmatrix}-\frac{I}{R}\left(C+K_P\right) & -\frac{L}{R}\left(C+K_P\right)\\ -\frac{I}{R}\left(C+K_P\right) & \frac{L}{R}\left(C+K_P\right)\end{bmatrix}\begin{bmatrix}v\\ \dot{\phi}\end{bmatrix}\\&\qquad+\begin{bmatrix}\frac{I}{R}K_P & \frac{L}{R}K_P\\ \frac{I}{R}K_P & -\frac{L}{R}K_P\end{bmatrix}\begin{bmatrix}v_d\\ \dot{\phi}_d\end{bmatrix}+\begin{bmatrix}K_I & 0\\ 0 & K_I\end{bmatrix}\begin{bmatrix}S_r\\ S_l\end{bmatrix}\end{aligned}\tag{3.24}$$

Then, we have

$$\begin{aligned}\begin{bmatrix}\dot{v}\\ \ddot{\phi}\end{bmatrix}&=\begin{bmatrix}-\frac{2}{MR^2+2I_{mt}}\left(C+K_P\right) & 0\\ 0 & -\frac{2}{IR^2+2I_{mt}L^2}\left(C+K_P\right)\end{bmatrix}\begin{bmatrix}v\\ \dot{\phi}\end{bmatrix}\\&\quad+\begin{bmatrix}\frac{2}{MR^2+2I_{mt}}K_P & 0\\ 0 & \frac{2}{IR^2+2I_{mt}L^2}K_P\end{bmatrix}\begin{bmatrix}v_d\\ \dot{\phi}_d\end{bmatrix}+\begin{bmatrix}\frac{R}{MR^2+2I_{mt}}K_I & \frac{R}{MR^{2L^2}+2I_{mt}}K_I\\ \frac{RL}{IR^2+2I_{mt}L^2}K_I & -\frac{RL}{IR^2+2I_{mt}L^2}K_I\end{bmatrix}\begin{bmatrix}S_r\\ S_l\end{bmatrix}\end{aligned}\tag{3.25}$$

Defining the state vector by $x=\begin{bmatrix}v & \dot{\phi} & S_r & S_l\end{bmatrix}^T$, input vector by $u=\begin{bmatrix}v_d & \dot{\phi}_d\end{bmatrix}^T$, and output vector by $y = \begin{bmatrix} S_r & S_l \end{bmatrix}^T$ in Eq. (3.25), we employed the following linear dynamics for simulation:

$$
\begin{aligned}
&\dot{x} = Ax + Bu + D \\
&y = Cx \\
&A = \begin{bmatrix} -a_1 & 0 & a_3 & a_3 \\ 0 & -a_2 & a_4 & -a_4 \\ -a_5 & -a_6 & 0 & 0 \\ -a_5 & a_6 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \\ a_5 & a_6 \\ a_5 & -a_6 \end{bmatrix}, \\
&C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} -d_1 & -d_1 \\ -d_2 & d_2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}
\tag{3.26}
$$

where

$$
\begin{aligned}
a_1 &= -\frac{2}{MR^2 + 2I_{mt}}(C + K_P) & a_2 &= -\frac{2L^2}{IR^2 + 2I_{mt}L^2}(C + K_P) & a_3 &= \frac{R}{MR^2 + 2I_{mt}}K_I \\
a_4 &= \frac{RL}{IR^2 + 2I_{mt}L^2}K_I & a_5 &= \frac{1}{R} \\
b_1 &= -\frac{2}{MR^2 + 2I_{mt}}K_P & b_2 &= -\frac{2L^2}{IR^2 + 2I_{mt}L^2}K_P \\
d_1 &= \frac{R}{MR^2 + 2I_{mt}} & d_2 &= \frac{RL}{IR^2 + 2I_{mt}L^2}
\end{aligned}
$$

3.5 Swarm Robot

Swarm optimization, swarm intelligence and swarm robotics are the fields considering a group of relatively simple individuals able to cooperate to perform complex tasks, in decentralized manner. The inspiration is found in the first line within animal societies, such as bird, ants and bees. Social insects exhibit successful behavior in performing complex tasks on the level of the group. These swarms are robust, able to adapt to constant environmental changes in conditions of limited communications among members and lack of global data. Swarm robotics are inspired by the flocking of animal behavior in their living environment to accomplish a task.

Swarm robots application have improved in the past few year since despite of many complicated artificial algorithms proposed to improve the correctness and efficiency in decision making of a single robot, the application of a team of robots has more advantages, such as the ability to accomplish task faster than single robot [47][50][51][104].

In 1986, Reynold [43] introduced three heuristic rules that lead to creation of the first computer animation of flocking. The three flocking rules of Reynolds are:

- Flock centering: attempt to stay close to nearby flock-mates
- Collision avoidance: avoid collision with nearby flock-mates
- Velocity matching: attempt to match velocity with nearby flock-mates

These three rules are also known as cohesion, separation and alignment that are subject to broad interpretation that complicates objective analysis and implementation of Reynold rules [43]. A number of recent studies on motion control for swarm robots suffer from common drawbacks such as the use of unbounded forces for collision avoidance, lack of scalability and irregular fragmentation and collapse [44][46][104].

In order to apply multiple robots in various human coexisting environments, the motion planning strategies is very important, such as obstacle avoidance in many different navigation map, for example through a narrow hall or inspecting a wide space. The obstacles considered are not only the static ones such as tables or chairs but also the dynamics ones such as human or other robots. Depending upon the classification, magnitude, and safety degree against obstacles, an allowable approaching distance needs to be determined.

Usually, a swarm robot system consists of many identical robots. For navigation of a swarm robot system, each mobile robot should know its location with respect to the global reference frame, as well as the position of obstacles and other mobile robot [54][105] [51]. The first objective of navigation will be moving to the goal point, and the the second objective will be avoiding obstacles, and the third objective will be avoiding other robot and human [106] [107] [108].

3.6 Swarm Service Mobile Robot

According to the International Federation of Robotics (IFR), service robot is a robot which operates semi or fully autonomously to perform services useful to well being of human and equipment, excluding manufacturing operation [2][66][70][72][73][75]. A current trend in robotics is integrating different types of sensors having different characteristics to improve the performance of the robot system and also benefit from the reduced cost of sensors.

System integration is defined as the process of bringing together the component subsystems into one system and ensuring that the subsystems function together as a system. This result of linking the process of different systems is expected to improve the performance of a robot system. One of the examples of system integration is by installing some different sensors with different characteristics. By designing a system with the combination of sensory data that has inherent redundancy and may provide robust recognition of a robot's working environment [16][17][18]. The application of this different type of sensors installment enables the robot to be applied in dynamic environments such as human living environment. One of the applications of service robots is rehabilitation environment. The primary objectives of the rehabilitation robots are to either fully or partially perform tasks that benefit the disabled people and support a rehabilitee's manipulative function [1][31][32].

Conventional, rehabilitation programs relied heavily on the experience and manual manipulation of the rehabilitator. Because rehabilitation must be conducted carefully and the number of rehabilitee continues to increase, a well-designed service robot may prove effective in providing the support required for careful rehabilitation. Although most service robots developed thus far are based on a single robot application, to provide better service, multiple service robots working together to create a swarm robot team is preferred, particularly for human services and rehabilitation purposes. The swarm robot may not only support the rehabilitee's movement but also conduct various tasks such as collecting or transporting certain objects during rehabilitation.

A swarm service mobile robot operating in the human living environment needs to cope with its dynamic changes. Hence, the fundamental function of the robot is to avoid static and dynamic obstacles. Particularly, mobile robots in the swarm team must maintain their velocity and avoid collisions with other swarm mates [11][44]-[51]. Existing studies have employed proximity [45] and vision sensors [48][52] for swarm robot obstacle avoidance, and the radio frequency identification (RFID) for localization and navigation purposes [52]-[55].

This study presents the collision avoidance control for swarm robots moving in a dynamic environment of moving obstacles. A method is presented for installing different types of sensors to achieve motion in dynamic environment. The proposed method combines the information obtained by several proximity sensors, an image sensor and a localization sensor. The application of leader-follower formation is considered, in which the swarm robot team has a leader that follows the rehabilitee, while the other robot follows the leader robot.

This study considers the two wheeled differential drive mobile robot discussed in chapter 2 as shown in Figure 2.4 with dynamics in Eq. (2.23). From the dynamic of mobile robot in Eq. (2.23) and based on repulsive potential field, a Reference Controller, Eq. (3.9), is derived to create the reference trajectory. The proposed method is shown in Figure 3.2. The position of the rehabilitee and the leader robot are considered achieved from a localization sensor attached to them. The swarm robot discussed in this study is concentrating on non-communicative swarming, where there is no communication between the leader and follower but the follower keeps on following the leader robot based on leader robot position detection. This method is also known as communication through environment where the leader robot leaves its "traces" in the environment and follower robot senses the trace without direct communication between them [119].

3.7 Stability Analysis

The stability analysis for obstacle avoidance in this study is by considering the case where robot is situated inside four walls spaces as shown in Figure 3.4. The objective of this arrangement is to investigate the effect of the walls to the robot in order to confirm the stability of the robot system.

The distance from the sensors to the walls in Figure 3.4 is given by

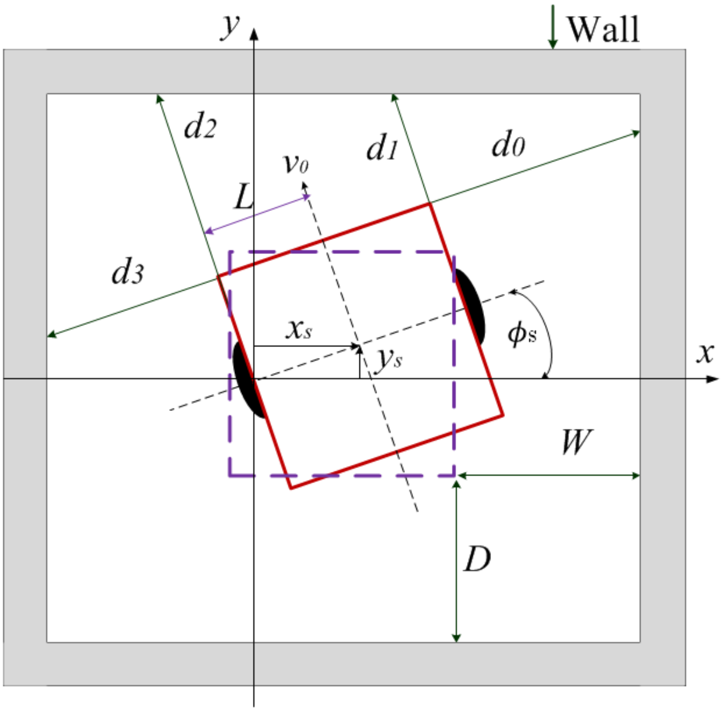


Figure 3.4: Stability analysis model

$$\begin{aligned} d_0 &= \frac{W+L-x_s}{\cos\phi_s} - L + L\tan\phi_s \\ d_1 &= \frac{D+L-y_s}{\cos\phi_s} - L\tan\phi_s + L \\ d_2 &= \frac{D+L-y_s}{\cos\phi_s} + L\tan\phi_s + L \\ d_3 &= \frac{W+L+x_s}{\cos\phi_s} - L - L\tan\phi_s \end{aligned} \tag{3.27}$$

where W, D are the distance from the mobile robot to the wall, x_s, y_s are the center position of the robot, L is the half width of the robot, and ϕ_s is the robot orientation.

The shape function in Eq.(3.14) is approximated as follows:

$$\begin{aligned} s(d_0) &= \frac{1}{a}\left[\exp\left\{-\frac{\left(d_0-\bar{d}\right)^2}{2\sigma^2}\right\}-b\right] \\ &\cong \frac{1}{a}\left[\exp\left\{-\frac{\left(W-\bar{d}\right)^2}{2\sigma^2}\right\}-b\right] \\ &\quad -2L\frac{W-\bar{d}}{a\sigma^2}\exp\left\{-\frac{\left(W-\bar{d}\right)^2}{2\sigma^2}\right\}\phi_s \\ &\quad +\frac{W-\bar{d}}{a\sigma^2}\exp\left\{-\frac{\left(W-\bar{d}\right)^2}{2\sigma^2}\right\}x_s \\ &\cong s_W - Lp_W\phi_s + p_W x_s \end{aligned} \tag{3.28}$$

where s_W and p_W are constants for stability analysis and given by

$$s_W = \frac{1}{a}\left[\exp\left\{-\frac{\left(W-\bar{d}\right)^2}{2\sigma^2}\right\}-b\right] \tag{3.29}$$

$$p_W = \frac{W-\bar{d}}{a\sigma^2}\exp\left\{-\frac{\left(W-\bar{d}\right)^2}{2\sigma^2}\right\} \tag{3.30}$$

$$\begin{aligned} s(d_1) &= \frac{1}{a}\left[\exp\left\{-\frac{\left(d_1-\bar{d}\right)^2}{2\sigma^2}\right\}-b\right] \\ &\cong \frac{1}{a}\left[\exp\left\{-\frac{\left(D-\bar{d}\right)^2}{2\sigma^2}\right\}-b\right] \\ &\quad +L\frac{D-\bar{d}}{a\sigma^2}\exp\left\{-\frac{\left(D-\bar{d}\right)^2}{2\sigma^2}\right\}\phi_s \\ &\quad +\frac{D-\bar{d}}{a\sigma^2}\exp\left\{-\frac{\left(D-\bar{d}\right)^2}{2\sigma^2}\right\}y_s \\ &\cong s_D + Lp_D\phi_s + p_D y_s \end{aligned} \tag{3.31}$$

while s_D and p_D are constants for stability analysis and given by

$$s_D = \frac{1}{a}\left[\exp\left\{-\frac{\left(D-\bar{d}\right)^2}{2\sigma^2}\right\}-b\right] \tag{3.32}$$

$$p_D = \frac{D-\bar{d}}{a\sigma^2}\exp\left\{-\frac{\left(D-\bar{d}\right)^2}{2\sigma^2}\right\} \tag{3.33}$$

In similar manner, we have

$$\begin{aligned} s\left(d_2\right) &= \frac{1}{a}\left[\exp\left\{-\frac{\left(d_2-\bar{d}\right)^2}{2\sigma^2}\right\}-b\right] \\ &\cong s_D - Lp_D\phi_s + p_D y_s \end{aligned} \tag{3.34}$$

$$\begin{aligned} s\left(d_3\right) &= \frac{1}{a}\left[\exp\left\{-\frac{\left(d_3-\bar{d}\right)^2}{2\sigma^2}\right\}-b\right] \\ &\cong s_W + Lp_W\phi_s - p_W y_s \end{aligned} \tag{3.35}$$

where $\bar{d} \leq W \leq d_{\max}$ and $\bar{d} \leq W \leq d_{\max}$ are assumed.

Considering the dynamics in Eq. (2.23) and related equations, we have

$$\begin{aligned} I\ddot{\phi} + C_\phi\dot{\phi} &= \tau_u + \operatorname{sgn}\left(v_0\right)\alpha_1\left\{\sum_{i=0,1} s\left(d_i\right) - \sum_{j=2,3} s\left(d_j\right)\right\} \\ &+ \alpha_2\left\{\sum_{i=0,2} s\left(d_i\right) - \sum_{j=1,3} s\left(d_j\right)\right\} \end{aligned} \tag{3.36}$$

where v_0 is the velocity of the robot. Substituting Eqs. (3.28), (3.31), (3.34) and (3.35) into Eq. (3.36) results in

$$\begin{aligned} I\ddot{\phi} + C_\phi\dot{\phi} =& \tau_u - 2\left[\operatorname{sgn}\left(v_0\right)\alpha_1\left\{Lp_W - Lp_D\right\} + \alpha_2\left\{Lp_W - Lp_D\right\}\right]\phi_s \\ &+ \operatorname{sgn}\left(v_0\right)2\alpha_1 p_W x_s + 2\alpha_2 p_W x_s \end{aligned} \tag{3.37}$$

By notating that

$$Q = \operatorname{sgn}\left(v_0\right)\alpha_1\left\{Lp_W - Lp_D\right\} + \alpha_2\left\{Lp_W - Lp_D\right\} \tag{3.38}$$

we have the following dynamics from Eq. (3.37)

$$I\ddot{\phi} + C_\phi\dot{\phi} = \tau_u - 2Q\phi_s + \operatorname{sgn}\left(v_0\right)2\alpha_1 p_W x_s + 2\alpha_2 p_W x_s \tag{3.39}$$

From figure 3.4, we have

$$\dot{x_s} = -v_0 \sin\phi_s \tag{3.40}$$

where ϕ_s is the orientation of the robot in Figure 3.4 and considered to have small magnitude to analyze the stability in a typical linear manner.

Notating that $z = \begin{bmatrix} \phi_s & \dot{\phi}_s & x_s \end{bmatrix}^T$, we have

$$\dot{z} = A_\phi z + \tau_u \tag{3.41}$$

where A_ϕ is a matrix derived from Eq. (3.39) as follows:

$$A_\phi = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{2Q}{I} & \frac{C_\phi Q}{I} & \frac{\mathrm{sgn}(v_0)2\alpha_1 p_W + 2\alpha_2 p_W}{I} \\ -(v_0) & 0 & 0 \end{bmatrix} \tag{3.42}$$

The determinant $\det(sI - A)$ is given by

$$\left|sI - A_\phi\right| = s^3 + \frac{C_\phi Q}{I}s^2 + \frac{2\alpha_1 p_W + 2\alpha_2 p_W}{I}\left|v_0\right| \tag{3.43}$$

Because p_W is positive, if the conditions below are satisfied

$$\begin{aligned} & Q > 0 \\ & \frac{C_\phi Q}{I} - \alpha_1 p_W \left|v_0\right| > 0 \end{aligned} \tag{3.44}$$

then the system in Eq. (3.41) is stable [109].

Next, we consider the translational motion of mobile robots. Considering Eqs. (2.23), (3.9) and related equations, we have

$$M\dot{v} + C_v v = F_u - \beta_1\left(\sum_{i=1,2} s\left(d_i\right)\right) - \beta_2\left(\sum_{i=0,3} \min\left(\dot{d}_i, 0\right)\right) \tag{3.45}$$

Substituting Eqs. (3.28), (3.31), (3.34) and (3.35) into Eq. (3.37) results in

$$M\dot{v} + C_v v = F_u - \beta_1 p_D y_s - \beta_2 p_D \dot{y}_s \tag{3.46}$$

From

$$\dot{y_s} = v_s \tag{3.47}$$

notating that $z = \begin{bmatrix} y_s & \dot{y_s} \end{bmatrix}^T$, we have

$$\dot{z} = A_v z + F_u \tag{3.48}$$

where A_v is a matrix derived from Eq. (3.46) as follows:

$$A_v = \begin{bmatrix} 0 & 1 \\ -\frac{2\beta_1 p_D}{M} & -\frac{C_v + 2\beta_2}{M} \end{bmatrix} \tag{3.49}$$

The determinant $\det(sI - A_v)$ is

$$|sI - A_v| = s^2 + \frac{C_v + 2\beta_2}{M} s + \frac{2\beta_1 p_D}{M} \tag{3.50}$$

If $p_D, \beta_1, \beta_2 > 0$ then the system in Eq. (3.48) is stable [109].

3.8 Results

To prove the effectiveness of the proposed method in Figure 3.2, experiment and simulation are conducted. The simulation and experiment in subsection 3.8.1 are conducted to prove the effectiveness of the proposed method in reaching via points despite the existence of obstacles. In subsection 3.8.2, the simulation is conducted to see the feasibility of the proposed method in creating a swarm robot system in following a human rehabilitee in an rehabilitation environment where the static and dynamic obstacles are considered.

3.8.1 Obstacle Avoidance for a Mobile Robot Moving through Via Points

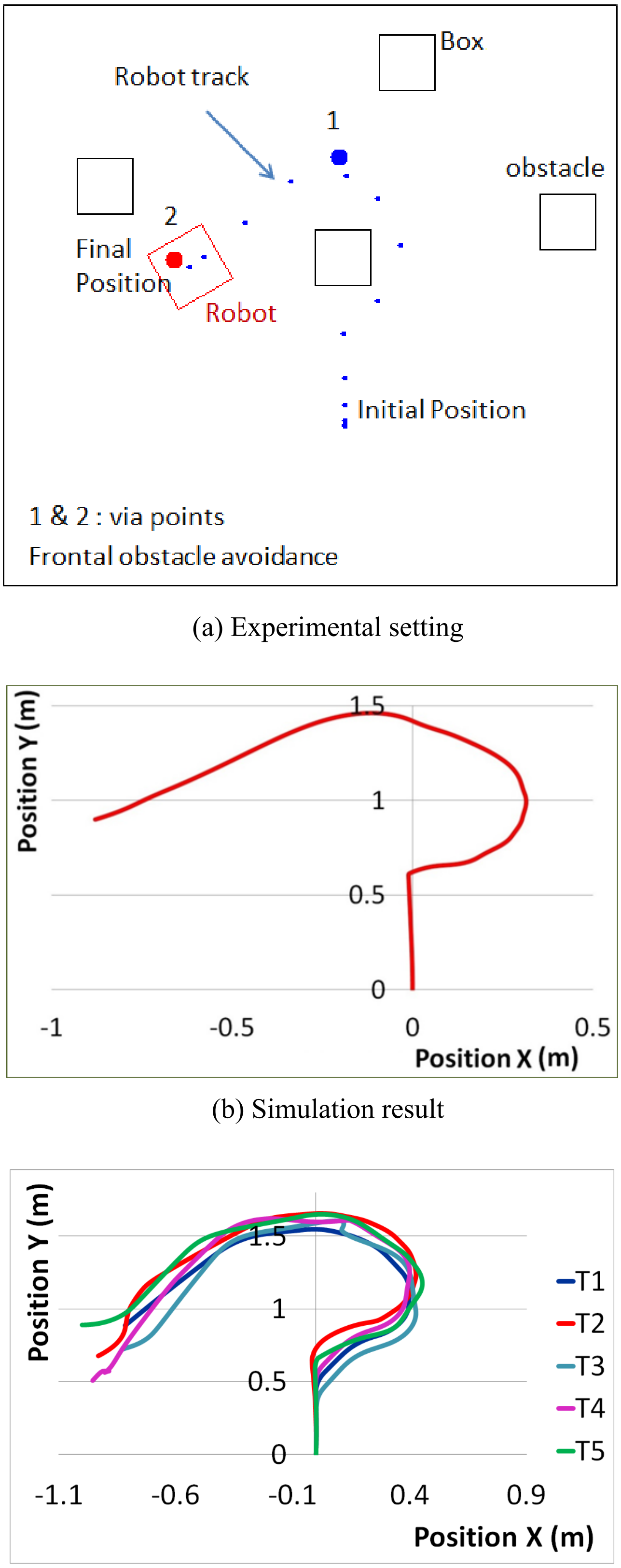
This chapter deals with collision avoidance control for a service mobile robot moving in a dynamic environment consisting of moving obstacles. A method for system integration is presented to achieve the motion in a dynamic environment. The proposed method combines the information obtained by several proximity sensors and an image sensor. The robot controllers consist of a reference controller and a PI controller. The reference controller generates a robot motion trajectory by referring sensor information in real-time, and the PI controller moves the robots to follow the generated motion trajectory. Various simulation and experimental results, which assume static and dynamic objects in the environment, demonstrate the effectiveness of the proposed design.

The experiment in this subsection is only to test the ability of the robot to move via points while avoiding obstacles; the static one and stop while encountering the dynamic one(s). Via points are considered pre-defined achieved from a localization sensor. The mobile robot applied in this study has four IR-proximity sensors installed on it and a Kinect sensor mounted on top of it. The algorithm and hardware design are kept simple so that the proposed control can be applied to a variety of service robots.

Figure 3.5 and 3.7 are the comparison of robot position and orientation of simulation results and experimental results, (a) is the experimental setting, and via points are indicated by numbers, (b) is the simulation and (c) is the experimental results. Two via points setting in Figure 3.5 is to show that the robot can reach target even the trajectory is blocked by obstacle, while many via points setting in Figure 3.7 is to show the possibility of robot's continuous tracking ability. We did the experiment five times for each via points setting shown in Figure 3.5c and 3.7c, where T is trajectory. Figure 3.6 and 3.8 are the video captures results of Figure 3.5 and 3.7, respectively which show the robot can reach all via points while avoiding obstacles.

The application of Kinect sensor is to know the existence of dynamic obstacle (passing human). In the current application, when "seeing" human and human reaches the minimum allowed distance, robot stops as shown in Figure 3.9. Kinect sensor is also used for the obstacle position's higher or undetectable by IR-proximity sensor.

The results obtained in this chapter will be used to develop a service mobile robot applied in human co-existing environment, and the stability analysis for obstacle avoidance in this study is given in chapter 5.



(a) Experimental setting

(b) Simulation result

(c) Experiment result

Figure 3.5: Two via points case

3.8.2 Swarm Robot Control for Human Services and Moving Rehabilitation by Sensor Integration

Computer simulations were performed to verify the effectiveness of the proposed method. Figure 3.10 shows the initial condition of the human living environment in which several static and dynamic objects exist. It includes steps low enough for the rehabilitee to step over, although these steps are considered to be static obstacles for robots. The passing humans are considered to be the dynamic obstacles that have to be avoided by the robots.

This simulation applies the Kinect sensor to enable the robot to “see” static and dynamic obstacles. Figure 3.11 shows the simulation result of the Kinect sensor application. The simulation was conducted by taking the distance data input from a real Kinect sensor detecting approaching human. The distance data is also compared with that from the proximity sensors. The application of Kinect sensor in Figure 3.11 shows that when the passing human approaches the robots, and the distance from the robot to the human is smaller than the allowed distance, the robots stop when the dwelling period reaches approximately 50s.

Figures 3.12a - 3.19a show the simulation screenshot, and Figures 3.12b - 3.19b show the simulation results that demonstrate the effectiveness of the proposed system. Figure 3.12 shows the rehabilitee and robots move in obstacle free environment. This setting is to verified that the robots can follow the rehabilitee's trajectory. The resulting graph (Figure 3.12b) shows no ripples (evidence of obstacle avoidance) since the mobile robots only consider the rehabilitee – robot distance and the distance between the robots, the rehabilitee just walks around, and robots determine the distance to static and dynamic obstacles. Figures 3.13a and 3.14a show the setting where the rehabilitee and the robots are moving around static obstacles (steps) while avoiding dynamic obstacles (passing humans). Figure 3.15 shows the setting where the rehabilitee and the robots are moving among statics obstacles and avoiding dynamics ones, therefore Figure 3.15a shows some ripples caused by obstacle avoidances. Figures 3.16 and 3.17 show the setting where the rehabilitee steps on the steps, therefore the robots have to follow the rehabilitee and at the same time avoiding the steps and the passing humans. Figures 3.18 and 3.19 show the environment in which the rehabilitee moves randomly. Figures 3.18b and 3.19b show trajectory results in random setups, in which more ripples are shown. In all these cases, the swarm robots successfully follow the rehabilitee while avoiding obstacles.

3.9 Conclusion

This chapter presents simple obstacle avoidance for a mobile robot moving through via points. The application of this robot presented in chapters 5 is a service mobile robot for human living environment. Simulation results show that the robot moves via points smoothly with two different via point settings. Experimental system was constructed and the effectiveness of the proposed system was confirmed. This chapter also presents the design of a collision avoidance control system for swarm robots moving in an environment that includes moving obstacles. The swarm robots follow the rehabilitee to provide support in performing his/her tasks in a dynamic environment. This study applies a reference and PI controller. The reference controller creates the reference trajectory for the PI controller based on the integrated sensor information obtained from the Kinect, proximity sensors and tracking device system. The obstacle avoidance trajectory is generated by the reference controller, and the stability of the overall system is analytically verified. Various computer simulations are performed to verify the effectiveness of the proposed method. The rehabilitee was successfully followed by the swarm robots in all situations.

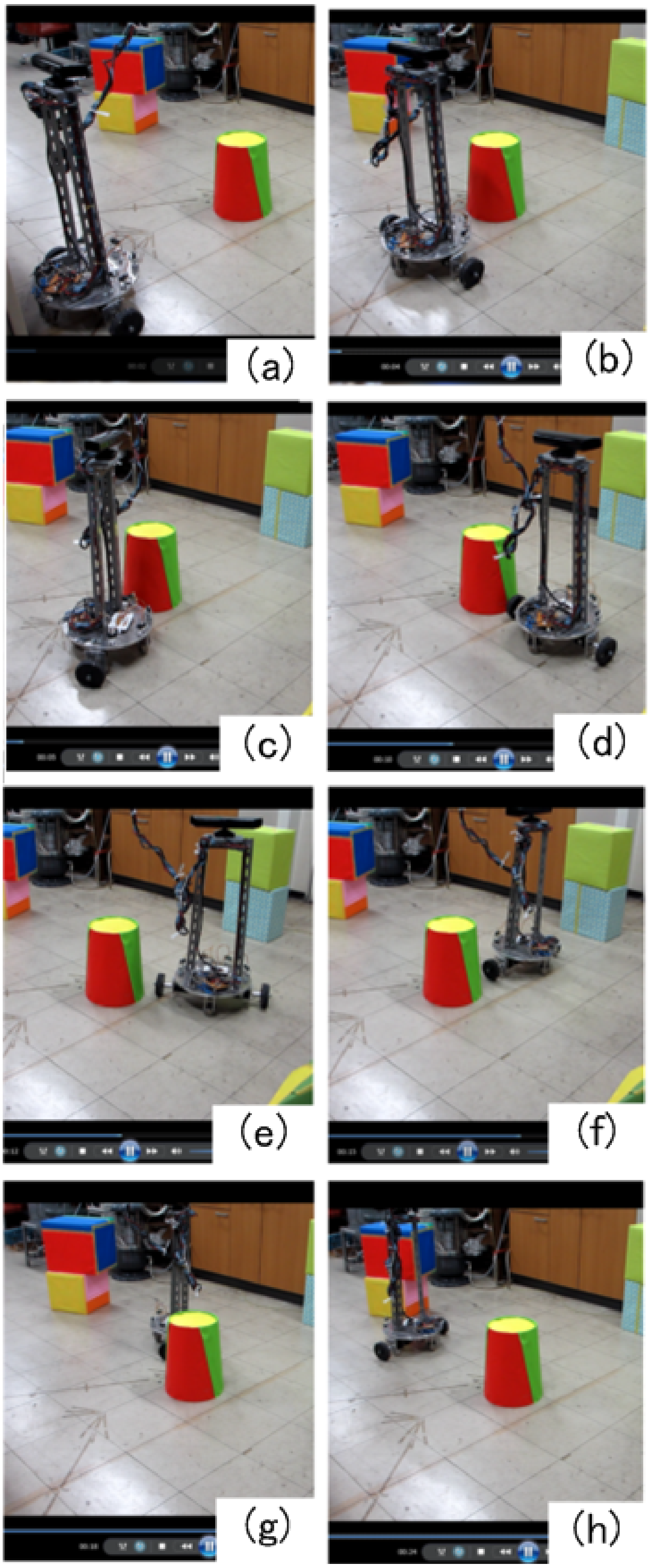
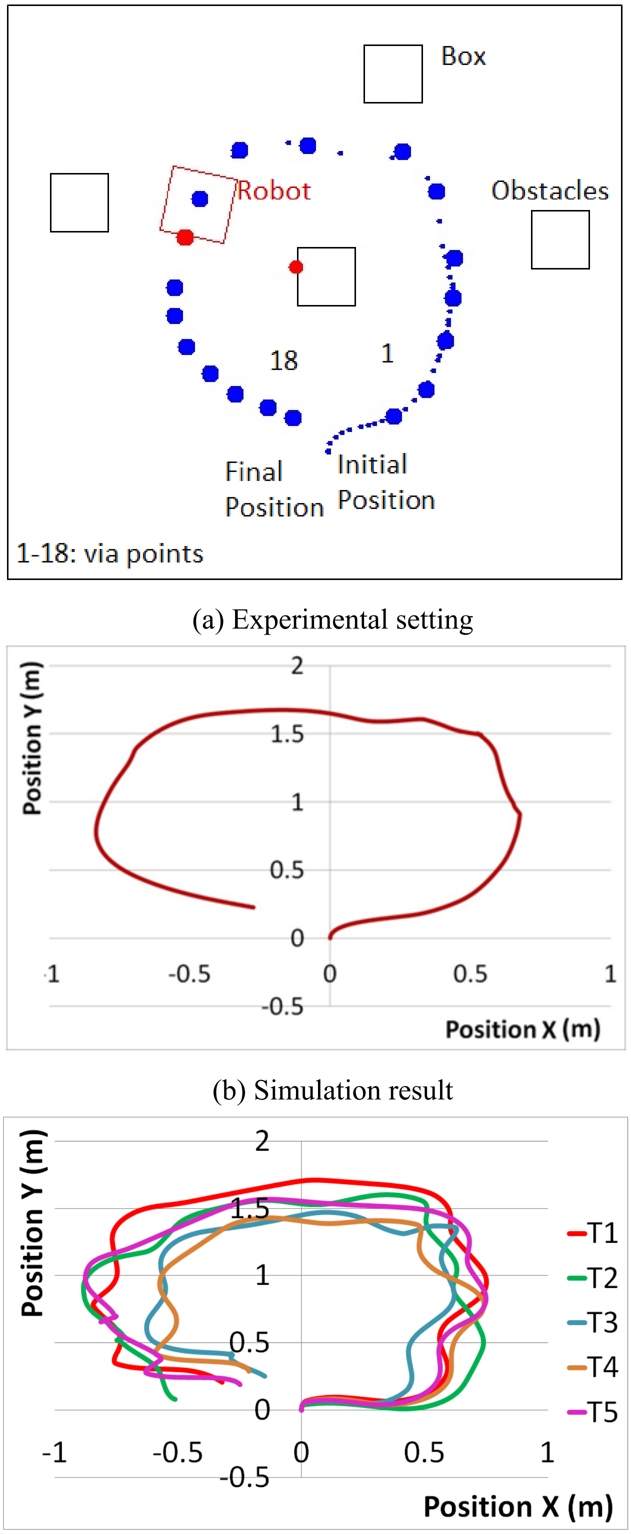


Figure 3.6: Video captures for results in Figure 3.5



(a) Experimental setting

(b) Simulation result

(c) Experiment result

Figure 3.7: Many via points case

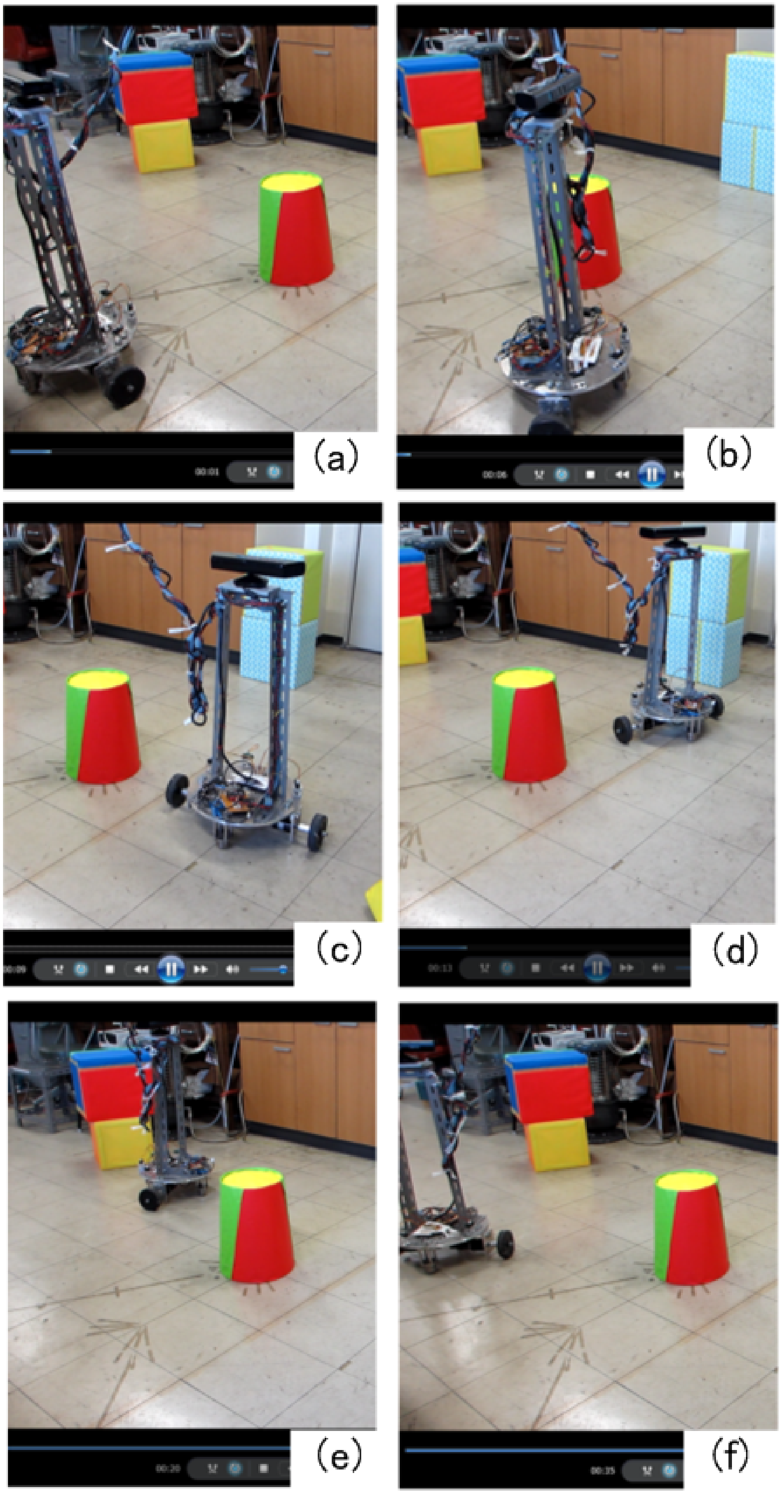


Figure 3.8: Video captures for result in Figure 3.7

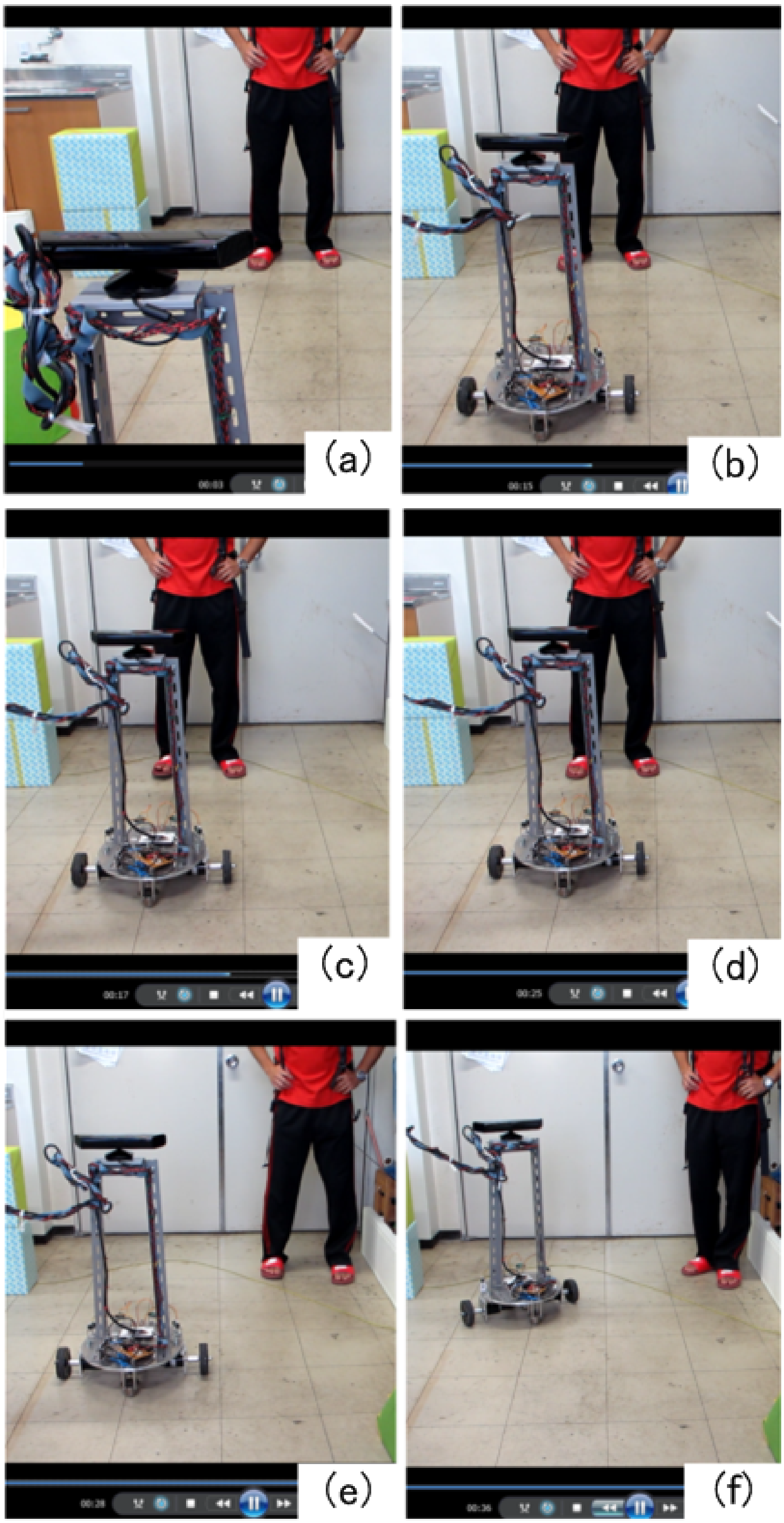


Figure 3.9: Dynamic obstacle detection

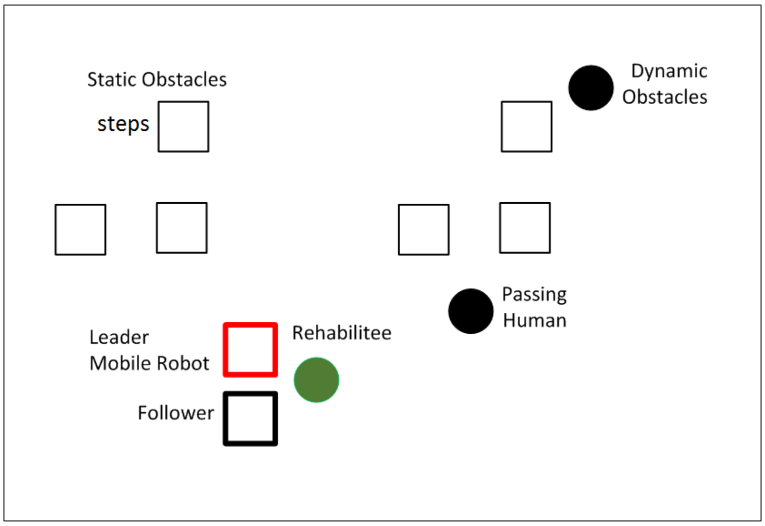


Figure 3.10: The initial condition of the human living environment

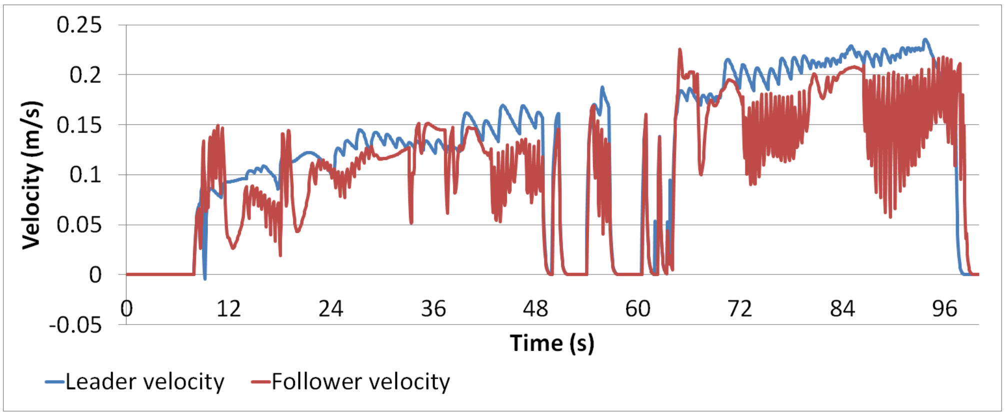
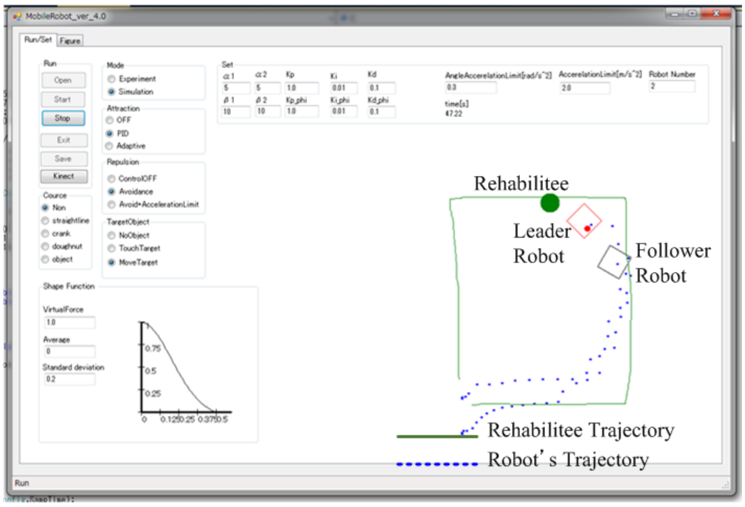


Figure 3.11: Simulation result of the Kinect sensor application



MobileRobot_ver_4.0

Mode: ○ Experiment ● Simulation

Attraction: ○ OFF ● PID ○ Adaptive

Repulsion: ○ ControlOFF ● Avoidance ○ Avoid+AccelerationLimit

TargetObject: ○ NoObject ○ TouchTarget ● MoveTarget

Cource: ● Non ○ straightline ○ crank ○ doughnut ○ object

Set

α1	α2	Kp	Ki	Kd	AngleAccerelationLimit[rad/s^2]	AccerelationLimit[m/s^2]	Robot Number
5	5	1.0	0.01	0.1	0.3	2.0	2

β1	β2	Kp_phi	Ki_phi	Kd_phi	time[s]
10	10	1.0	0.01	0.1	47.22

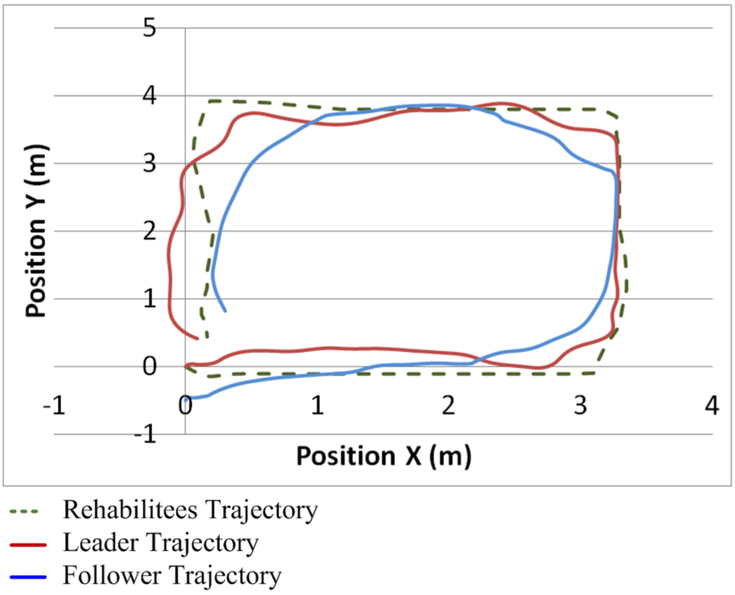
Shape Function

VirtualForce: 1.0

Average: 0

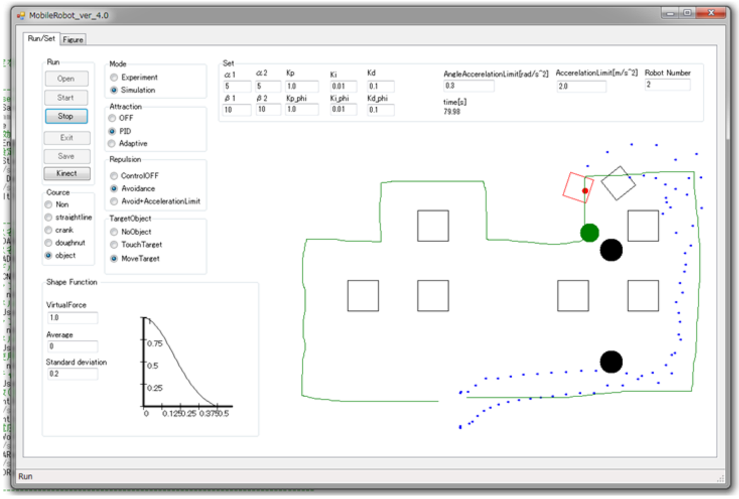
Standard deviation: 0.2

(a) Computer simulation screenshot



(b) Trajectories from the simulation

Figure 3.12: Simulation result for obstacle free environment



Run

Mode
○ Experiment
● Simulation

Attraction
○ OFF
● PID
○ Adaptive

Repulsion
○ ControlOFF
● Avoidance
○ Avoid+AccelerationLimit

Cource
○ Non
○ straightline
○ crank
○ doughnut
● object

TargetObject
○ NoObject
○ TouchTarget
● MoveTarget

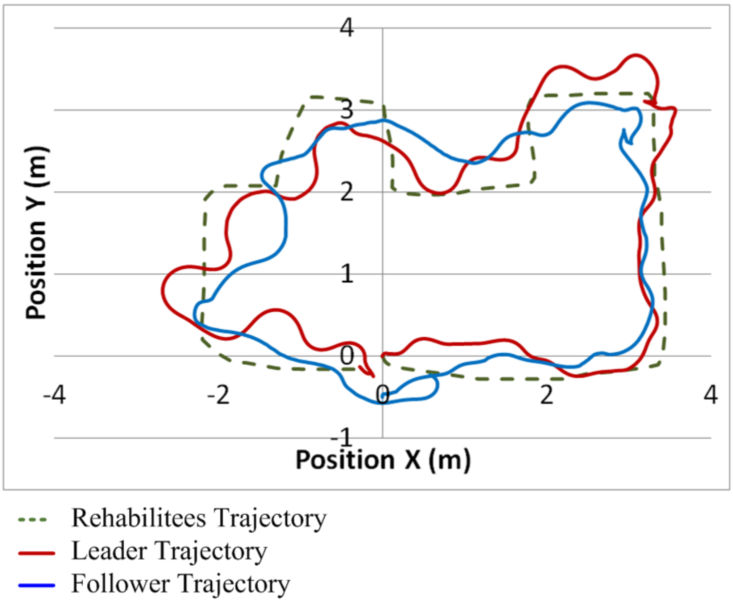
Set

α1	α2	Kp	Ki	Kd
5	5	1.0	0.01	0.1
β1	β2	Kp_phi	Ki_phi	Kd_phi
10	10	1.0	0.01	0.1

AngleAccerelationLimit[rad/s^2] 0.3
AccerelationLimit[m/s^2] 2.0
Robot Number 2
time[s] 79.98

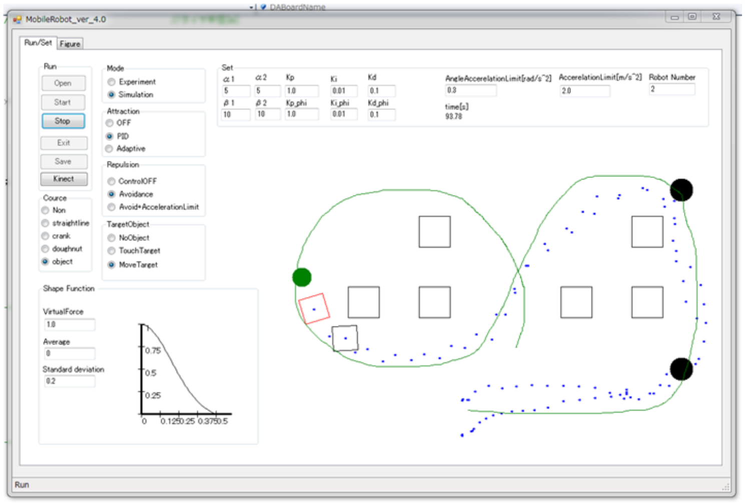
Shape Function
VirtualForce 1.0
Average 0
Standard deviation 0.2

(a) Computer simulation screenshot



(b) Trajectories from the simulation

Figure 3.13: Simulation result of first environment setting where human and the robots are moving around the obstacles and avoiding the passing human, referring to figure 3.10



Run

Mode
○ Experiment
● Simulation

Attraction
○ OFF
● PID
○ Adaptive

Repulsion
○ ControlOFF
● Avoidance
○ Avoid+AccelerationLimit

Cource
○ Non
○ straightline
○ crank
○ doughnut
● object

TargetObject
○ NoObject
○ TouchTarget
● MoveTarget

Set

α1	α2	Kp	Ki	Kd
5	5	1.0	0.01	0.1
β1	β2	Kp_phi	Ki_phi	Kd_phi
10	10	1.0	0.01	0.1

AngleAccerelationLimit[rad/s^2]
0.3

AccerelationLimit[m/s^2]
2.0

Robot Number
2

time[s]
93.78

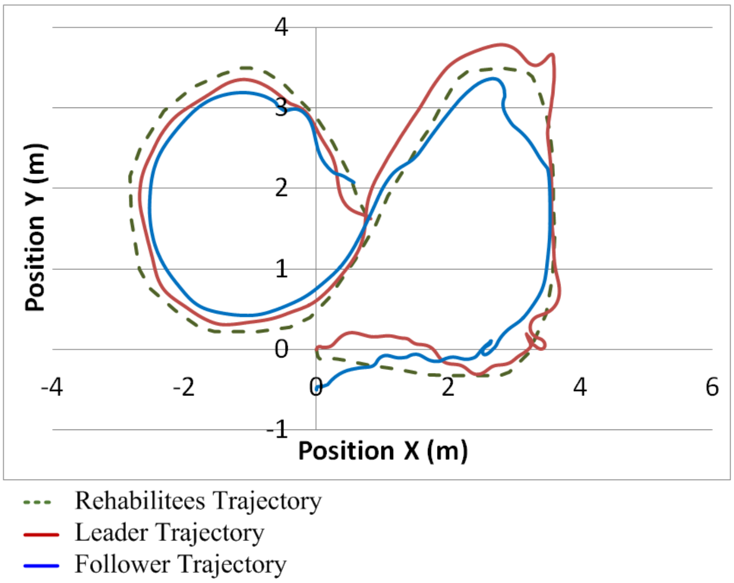
Shape Function

VirtualForce
1.0

Average
0

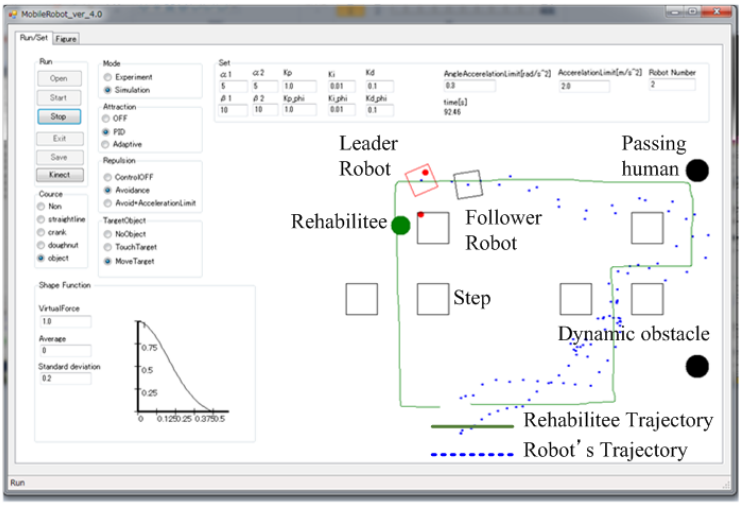
Standard deviation
0.2

(a) Computer simulation screenshot



(b) Trajectory for the simulation

Figure 3.14: Simulation result of second environment setting where human and the robots are moving around the obstacles and avoiding the passing human



Mode: ○ Experiment ● Simulation

Attraction: ○ OFF ● PID ○ Adaptive

Repulsion: ○ ControlOFF ● Avoidance ○ Avoid+AccelerationLimit

TargetObject: ○ NoObject ○ TouchTarget ● MoveTarget

Cource: ○ Non ○ straightline ○ crank ○ doughnut ● object

Set

α1	α2	Kp	Ki	Kd
5	5	1.0	0.01	0.1
β1	β2	Kp_phi	Ki_phi	Kd_phi
10	10	1.0	0.01	0.1

AngleAccerelationLimit[rad/s^2]: 0.3

AccerelationLimit[m/s^2]: 2.0

Robot Number: 2

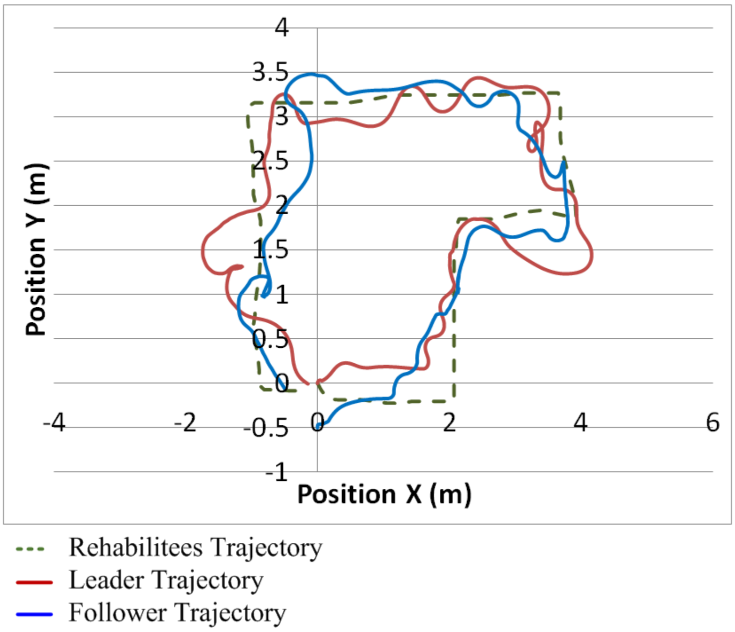
time[s]: 92.46

VirtualForce: 1.0

Average: 0

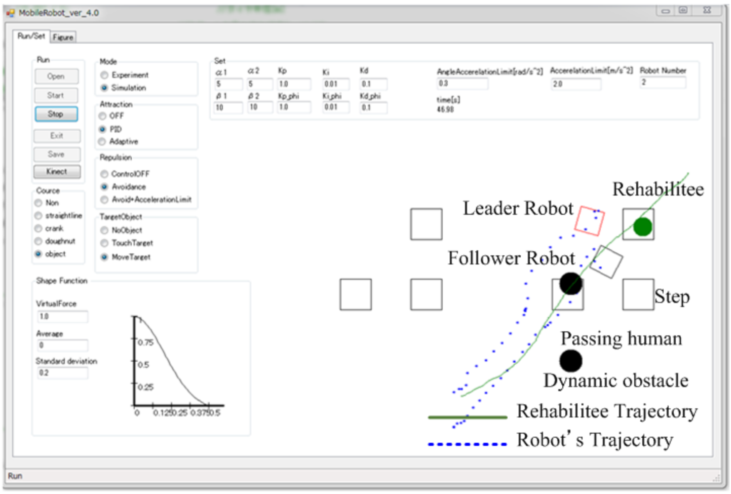
Standard deviation: 0.2

(a) Computer simulation screenshot



(b) Trajectories from the simulation

Figure 3.15: Simulation result for obstacle existing environment, where the rehabilitee and the robots are moving among static obstacles and avoiding the dynamic ones



Mode: ○ Experiment ● Simulation

Attraction: ○ OFF ● PID ○ Adaptive

Repulsion: ○ ControlOFF ● Avoidance ○ Avoid+AccelerationLimit

Cource: ○ Non ○ straightline ○ crank ○ doughnut ● object

TargetObject: ○ NoObject ○ TouchTarget ● MoveTarget

Set

α1	α2	Kp	Ki	Kd	AngleAccerelationLimit[rad/s^2]	AccerelationLimit[m/s^2]	Robot Number
5	5	1.0	0.01	0.1	0.3	2.0	2
β1	β2	Kp_phi	Ki_phi	Kd_phi	time[s]		
10	10	1.0	0.01	0.1	46.98		

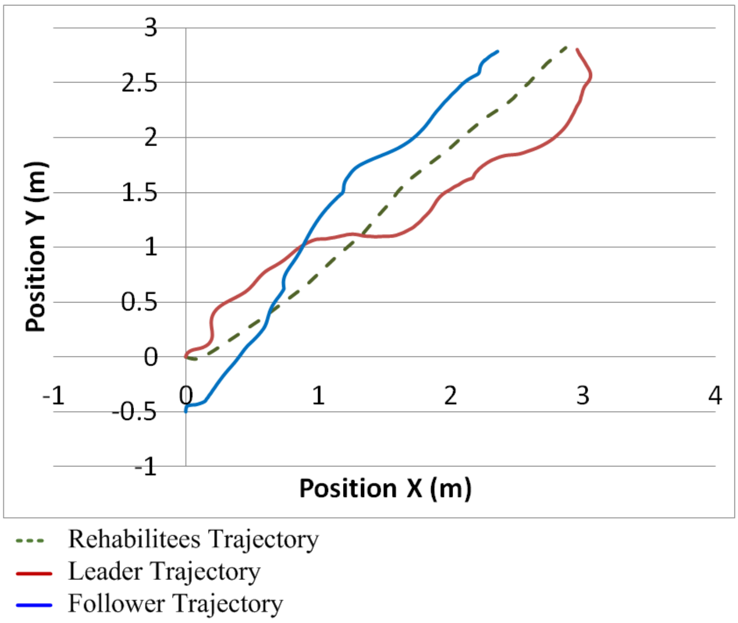
Shape Function

VirtualForce: 1.0

Average: 0

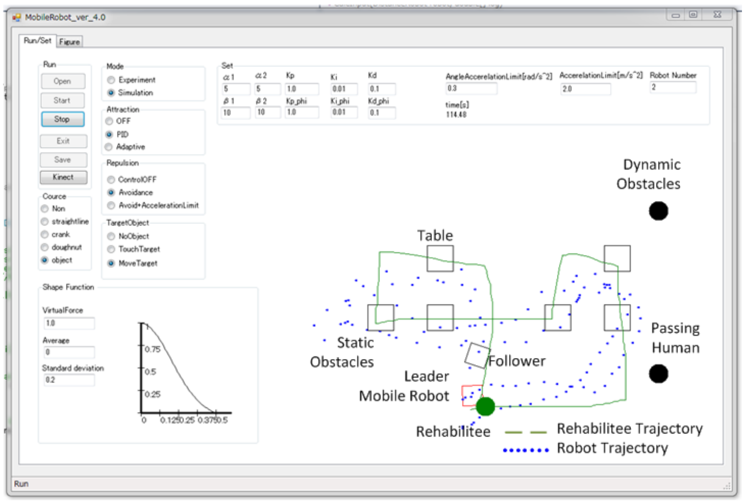
Standard deviation: 0.2

(a) Computer simulation screenshot



(b) Trajectories from the simulation

Figure 3.16: Simulation result of first experiment setting where rehabilitee are stepping on steps and the robots are avoiding the steps and passing human



Mode
○ Experiment
● Simulation

Attraction
○ OFF
● PID
○ Adaptive

Repulsion
○ ControlOFF
● Avoidance
○ Avoid+AccelerationLimit

Cource
○ Non
○ straightline
○ crank
○ doughnut
● object

TargetObject
○ NoObject
○ TouchTarget
● MoveTarget

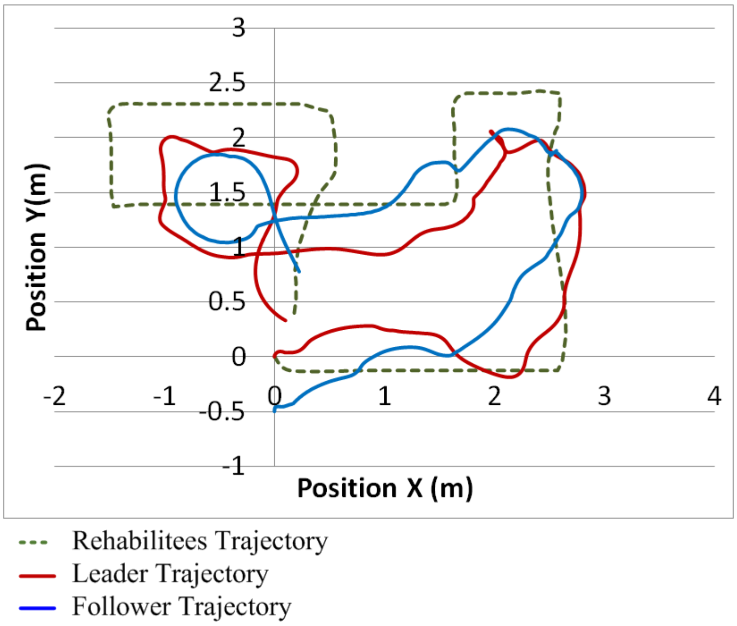
Set

α1	α2	Kp	Ki	Kd
5	5	1.0	0.01	0.1
β1	β2	Kp_phi	Ki_phi	Kd_phi
10	10	1.0	0.01	0.1

AngleAccerelationLimit[rad/s^2] 0.3
AccerelationLimit[m/s^2] 2.0
Robot Number 2
time[s] 114.48

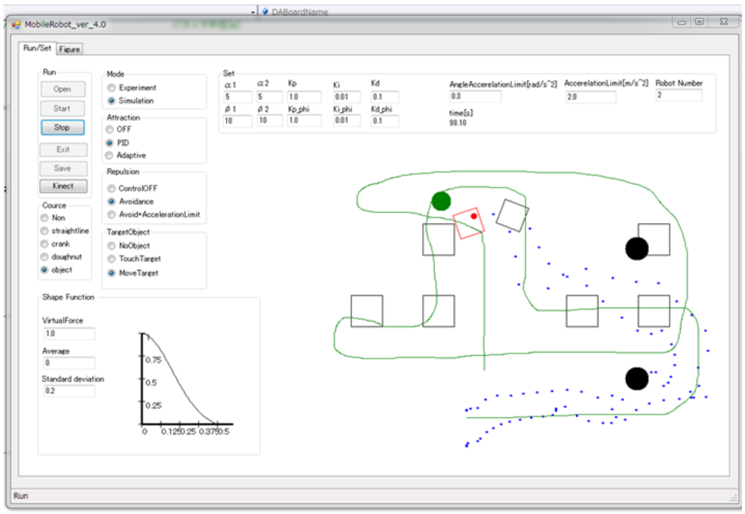
Shape Function
VirtualForce 1.0
Average 0
Standard deviation 0.2

(a) Computer simulation screenshot



(b) Trajectories from the simulation

Figure 3.17: Simulation result of second experiment setting where rehabilitee are stepping on steps and the robots are avoiding the steps and passing human



Mode
○ Experiment
● Simulation

Attraction
○ OFF
● PID
○ Adaptive

Repulsion
○ ControlOFF
● Avoidance
○ Avoid+AccelerationLimit

Cource
○ Non
○ straightline
○ crank
○ doughnut
● object

TargetObject
○ NoObject
○ TouchTarget
● MoveTarget

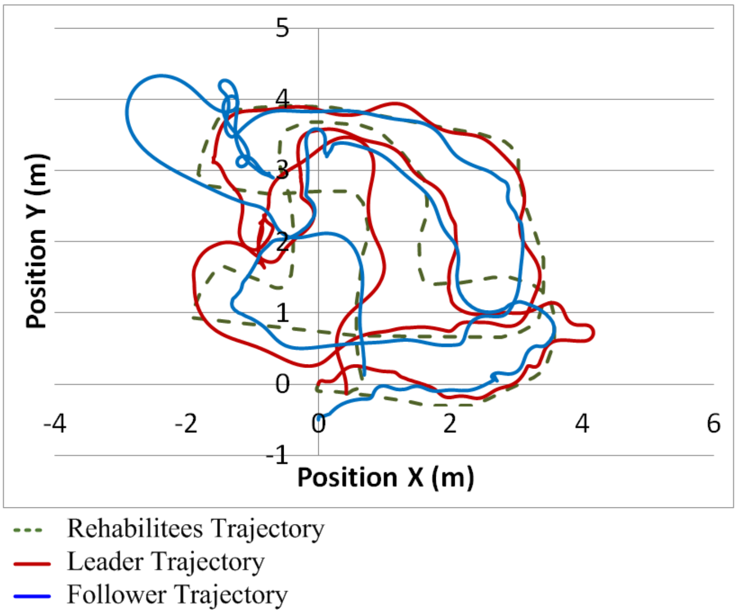
Set

α1	α2	Kp	Ki	Kd
5	5	1.0	0.01	0.1
β1	β2	Kp_phi	Ki_phi	Kd_phi
10	10	1.0	0.01	0.1

AngleAccerelationLimit[rad/s^2]: 0.3
AccerelationLimit[m/s^2]: 2.0
Robot Number: 2
time[s]: 98.10

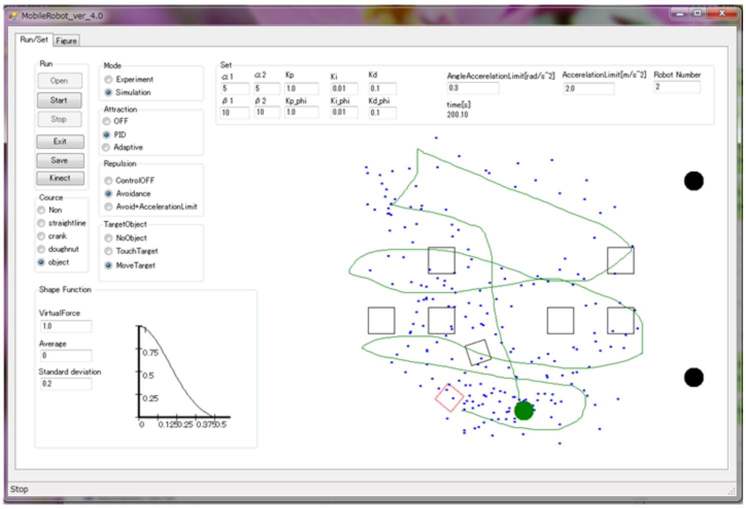
Shape Function
VirtualForce: 1.0
Average: 0
Standard deviation: 0.2

(a) Computer simulation screenshot



(b) Trajectories from the simulation

Figure 3.18: Simulation result of first experiment setting where rehabilitee are moving in random motion, stepping on steps and the robots are avoiding the steps and passing human



Mode: ○ Experiment ● Simulation

Attraction: ○ OFF ● PID ○ Adaptive

Repulsion: ○ ControlOFF ● Avoidance ○ Avoid+AccelerationLimit

TargetObject: ○ NoObject ○ TouchTarget ● MoveTarget

Cource: ○ Non ○ straightline ○ crank ○ doughnut ● object

α1	α2	Kp	Ki	Kd
5	5	1.0	0.01	0.1
β1	β2	Kp_phi	Ki_phi	Kd_phi
10	10	1.0	0.01	0.1

AngleAccerelationLimit[rad/s^2]: 0.3

AccerelationLimit[m/s^2]: 2.0

Robot Number: 2

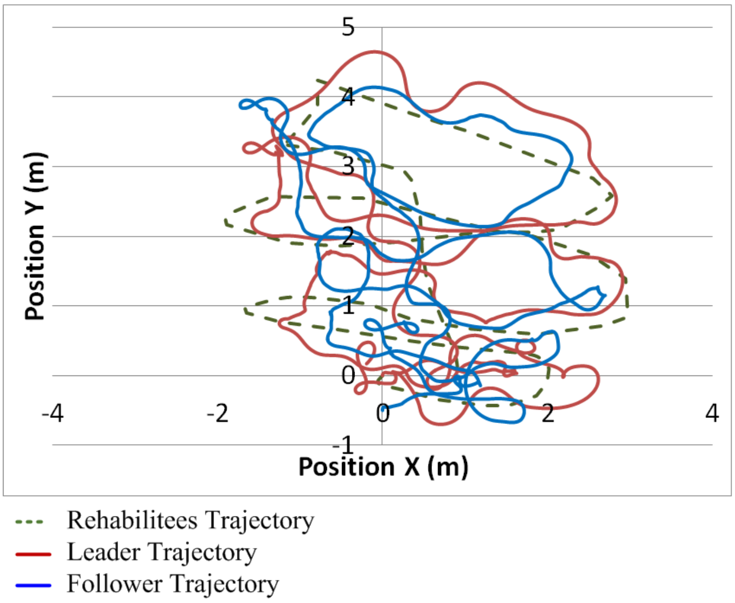
time[s]: 200.10

VirtualForce: 1.0

Average: 0

Standard deviation: 0.2

(a) Computer simulation screenshot



(b) Trajectories from the simulation

Figure 3.19: Simulation result of second experiment setting where rehabilitee are moving in random motion, stepping on steps and the robots are avoiding the steps and passing human

Chapter 4

Collision Avoidance Control for a Human-Operated Four-Wheeled Mobile Robot

4.1 Introduction

This chapter related to the work presented in [84] as the extended of the method in chapter 3. In this chapter, the two-wheeled mobile robot is replaced with four-wheeled mobile robot.

Fully automated robots are desirable to support household chores, nursing and welfare work, and industrial tasks performed by skilled workers. However, from the viewpoint of cost efficiency, it is impractical to produce such robots using the currently available technology. Human-operated robotic systems are a suitable solution, and hence, widely studied. The objectives of human-operated robots include extending human mechanical power [85][86], providing precise and smooth operations in difficult physical tasks [87][88], and executing missions in remote or hazardous environments [89][90]. In human-operated robotic systems, controllers are required to incorporate the human operator commands and compensate for operator's mistakes without hampering the ease of operation. Collision avoidance function are necessary for easy and safe operation of a robot operated by an elderly or disabled person. We consider a collision avoidance control for human-operated four wheeled mobile robots that are widely used in common vehicle systems.

Much research has been conducted on obstacle avoidance for mobile robots [38][41][91] [92][93]. The potential field method based on the idea of imaginary forces acting on a robot is one of the most popular approaches to obstacle avoidance. This approach has been extended by many studies. Because the four-wheeled robot is a nonholonomic system, the obstacle avoidance function must consider this dynamic property. The robot manipulator dynamics is considered and decoupled in the implementation of the obstacle avoidance function presented in [38], however, this decoupling approach cannot be applied to the nonholonomic four-wheeled mobile robot. The dynamic window approach is one of the most efficient approaches that consider the nonholonomic constraint and can be applied to unknown environments [94][95]. In this approach, the mobile robot destination is given and the robot motion is generally determined by optimizing a certain cost function such as the distance to the destination.

In the field of autonomous vehicle control, Reichardt and Schick [96] proposed the concept of risk map to achieve human-like behaviour. A risk map is an egocentric map of potentials reflecting the risk at a certain position in the environment. Gerdes and Rossetter [97][98] proposed an approach based on the concept of artificial potential fields, which ensures safe motion in the absence of driver inputs. Wolf and Burdick [99] presented a set of potential function components to assist automated and semi-automated vehicles. However, these approaches require computational effort and expensive sensors to construct and employ the risk map and artificial potential fields. This chapter aims to present a simple approach that employs inexpensive distance sensors.

The social force model, which has been used to explain pedestrian motion [100][101], considers the dynamics of a pedestrian and the imaginary social forces acting on him/her in order to avoid collisions with other people or walls. Based on this concept, we propose a control approach for collision avoidance in which the control input signal is modified according to the distance sensor information. The proposed control system is an extension of that in [79] to a four-wheeled robot. A stability analysis is performed to validate the proposed approach. The effectiveness of the proposed approach is demonstrated by experimental results obtained when several unskilled operators control the four-wheeled robot in a corridor-like environment.

4.2 Controller Design for Collision Avoidance

4.2.1 Social Force Model

Helbing and Molnar [100] first introduced the social force model to explain pedestrian motion. The social forces are considered to act a pedestrian in order to avoid collisions with other people or walls and to enable motion in a specific direction at a given speed. The social forces for collision avoidance are modeled as repulsion forces from obstacles such as other people or walls. Follow-up studies on this concept have been conducted [101][102]. This subsection briefly explains the concept of the social force model. The social force model is defined as follows:

$$\frac{dw}{dt} = F + F_l \tag{4.1}$$

where w is the pedestrian velocity vector, F is the social force vector, and F_l is the fluctuation vector. The social force vector F defined in Eq. (4.2) consists of the attractive force from the desired position F_α, the repulsive force from other pedestrians and walls F_β, and attractive force from the objects of interest F_γ:

$$F = F_\alpha + F_\beta + F_\gamma \tag{4.2}$$

Helbing and Molnar [100] conducted computer simulations of interacting pedestrians and showed that the social force model can describe the pedestrian behavior including obstacle avoidance. The following section applies this concept to the robot vehicle control.

4.2.2 Dynamics and Control of the Four-Wheeled Mobile Robot

Figure 4.1 shows a schematic of the four-wheeled mobile robot. The dynamics of the four-wheeled mobile robot is represented as follows [97]:

$$m\dot{u}_x = f_{xr} + f_{xf}\cos\delta - f_{yf}\sin\delta + m\omega u_y \tag{4.3}$$

$$m\dot{u}_y = f_{yr} + f_{xf}\sin\delta + f_{yf}\cos\delta - m\omega u_x \tag{4.4}$$

$$I\dot{\omega} = -af_{xf}\sin\delta + af_{yf}\cos\delta - bf_{yr} + \frac{d}{2}\{f_{xrr} - f_{xlr} + (f_{xrf} - f_{xlf})\cos\delta\}, \tag{4.5}$$

$$f_{xf} = f_{xrf} + f_{xlf},\ f_{xr} = f_{xrr} + f_{xlr},\ f_{yf} = f_{yrf} + f_{ylf},\ f_{yr} = f_{yrr} + f_{ylr},$$

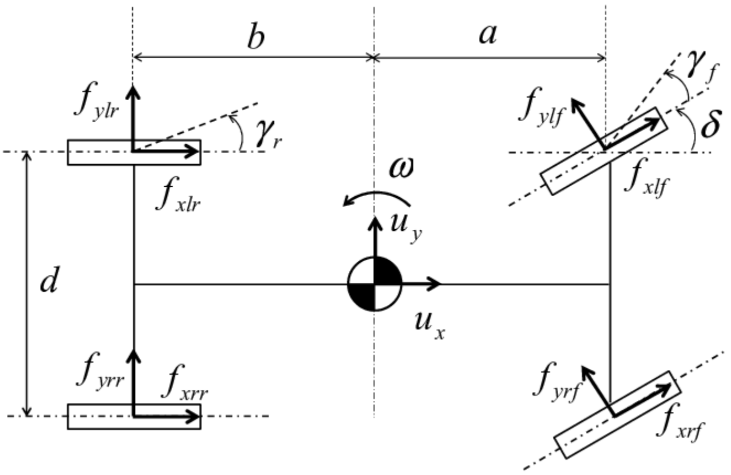


Figure 4.1: Four-wheeled robot model

where u_x: vehicle velocity in front-rear direction, u_y: vehicle velocity in the lateral direction, ω: vehicle angular velocity, f_{ijk}: force acting on each wheel $\{i$: force direction (x or y), j: right(r) or left(l) wheel, k: front(f) or real(r) wheel$\}$, δ: steering angle, m: vehicle mass, a, b: distance between the centre of gravity and the rear or front wheel, d: distance between rear wheels (front wheels), and $(\dot{\ })$: time derivative.

We assume that f_{xr} and δ are inputs provided by an operator, $f_{xrr} = f_{xlr}$, f_{xrf} and f_{xlf} are zero (i.e., rear-wheel drive) and vehicle parameters m, I, a, b and d are known and constant. The forces f_{yf} and f_{yr} are approximated as follows Gerdes2001:

$$f_{yf} \simeq -c_f \gamma_f = -c_f \tan^{-1}\left(\frac{u_y + \omega a}{u_x}\right) - \delta \tag{4.6}$$

$$f_{yr} \simeq -c_r \gamma_r = -c_r \tan^{-1}\left(\frac{u_y - \omega b}{u_x}\right), \tag{4.7}$$

where c_f and c_r are the cornering stiffness, and γ_f and γ_r are the sliding angles of the front and rear wheels. From Eqs. (4.3) - (4.7), we have the following dynamics:

$$m\dot{u}_x = f_{xr} + c_f \left\{ \tan^{-1}\left(\frac{u_y + \omega a}{u_x}\right) - \delta \right\} \sin\delta + m\omega u_y \tag{4.8}$$

$$m\dot{u}_y = -c_r \tan^{-1}\left(\frac{u_y - \omega b}{u_x}\right) - c_f \left\{ \tan^{-1}\left(\frac{u_y + \omega a}{u_x}\right) - \delta \right\} \cos\delta - m\omega u_x \tag{4.9}$$

$$I\dot{\omega} = -ac_f \left\{ \tan^{-1}\left(\frac{u_y + \omega a}{u_x}\right) - \delta \right\} \cos\delta + bc_r \tan^{-1}\left(\frac{u_y - \omega b}{u_x}\right), \tag{4.10}$$

This study assumes that several distance sensors are located on the robot. Figure 4.2

shows an example of a sensor location for the rectangular shaped robot. Because the distance information to obstacles is available, we include this information in steering angles and driving force generated by rear wheels for collision avoidance as follows:

$$\delta = \delta_d + \sum_{i=1}^{m} g_{ri} - \sum_{i=1}^{m} g_{li} \tag{4.11}$$

$$f_{xr} = f_d - \sum_{i=1}^{m} h_{ri} - \sum_{i=1}^{m} h_{li} \tag{4.12}$$

where δ_d and f_d are the steering angle and driving force designated by an operator, respectively, and f_d corresponds to the accelerating or braking force of a typical vehicle. The virtual steering angles g_{ri} and g_{li} are assumed to be proportional to the distance measurement at each sensor location as follows:

$$g_{ki} = -p_i d_{ki} + q_i, \; k = l, r \tag{4.13}$$

where the subscript l or r denote that the corresponding sensor is located on the left or right side of the robot body, i the sensor number and p_i and q_i are positive constants. In this study, g_{ri} and g_{li} are assumed to be positive. Eq. (4.11) indicates that the controller steers to the left when the distance to the obstacle measured by the sensor located at the right-hand side of the robot becomes small and vice versa.

Because we cannot directly apply the social force to the dynamics in Eqs (4.3)-(4.5), we propose to include the similar effect in the steering angel and driving force, as shown in Eqs. (4.11) and (4.12), which are common control variables in four-wheeled vehicle systems. This controller design has not been presented as far as author's knowledge.

For simplicity, the virtual forces h_{ri} and h_{li} in Eq. (4.12) are assumed to be proportional to the distance measurement at each sensor location as follows:

$$h_{ki} = -\hat{p}_i d_{ki} + \hat{q}_i, k = l, r \tag{4.14}$$

where $\hat{p}_i$ and $\hat{q}_i$ are positive constants. From Eq. (4.12), it can be seen that the smaller the distance, the larger the braking force provided by the controller. The effect of g_{ki} and h_{ki} can be interpreted as in the social force model, in which the social force is modeled as a virtual repulsive force to avoid collisions with obstacles.

4.2.3 Stability Analysis

Using the test case presented in Figure 4.2, we consider the validity of the proposed method for realizing the collision avoidance function in the human-operated robot. This subsection is devoted for analyzing the validity of the control in Eqs. (4.11) and (4.12), and the measurement of rotational deviation and forward speed as well as actual parameter values is not required for the control and the analysis in this subsection. Although the vehicle system has nonlinear dynamics in Eqs. (4.3)-(4.5), we apply a linear analysis at a certain operating point that is generally effective to predict the fundamental property of the control system. Experiments were conducted to further verify the effectiveness, and their results are shown in Section 3. For simplicity, the robot is assumed to have a rectangular shape. It is further assumed that the human operator intends to move the robot along the centerline between two parallel walls. Because of operational mistakes, the robot deviates from the centerline as shown in Figure 4.2. The lateral and rotational deviations are denoted by x and ϕ, respectively. The position in the vertical direction is denoted by y. In addition, we assume that all distance sensors are located symmetrically with respect to the centerline and only above the robot's centre of gravity, as shown in the Figure 4.1. The number of sensors located at the left or right half side of the robot is denoted by N.

The distance between each sensor and the walls are given as follows:

$$d_{ri} = \frac{L-x}{\cos\phi} + l_i \tan\phi - B, \tag{4.15}$$

$$d_{li} = \frac{L+x}{\cos\phi} - l_i \tan\phi - B, \tag{4.16}$$

where L is the half distance between the walls and B is the half width of the robot. l_i is the distance from the robot's centre of gravity to the ith distance sensor along the robot's centre line.

From Figure 4.2, we obtain the following relations:

$$\dot{x} = -u_x \sin\phi - u_y \cos\phi, \tag{4.17}$$

$$\dot{y} = u_x \cos\phi - u_y \sin\phi, \tag{4.18}$$

$$\dot{\phi} = \omega, \tag{4.19}$$

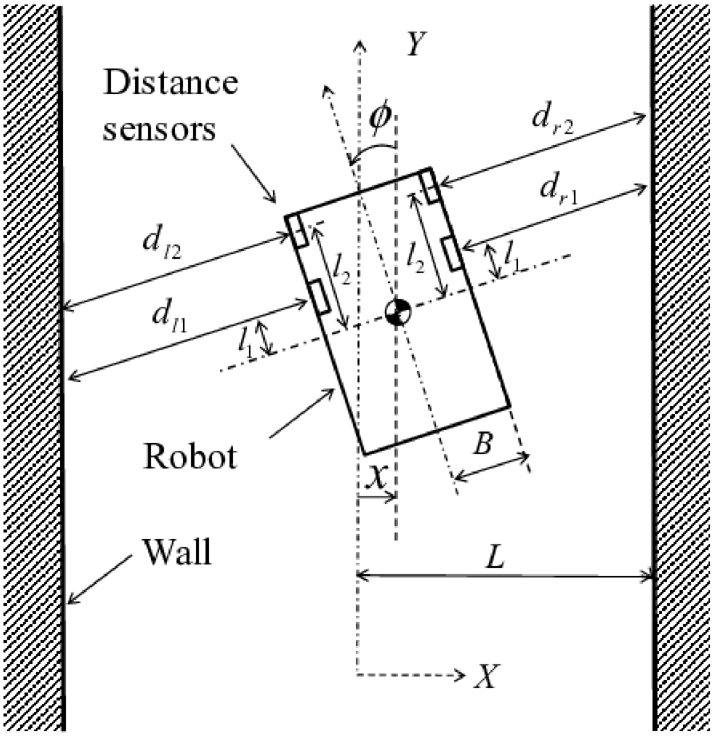


Figure 4.2: Robot motion between walls

Assuming that the robot moves with approximately the constant speed (operating point) $(u_x, u_y, \omega) = (u_{x0}, 0, 0)$ as follows:

$$u_x = u_{x0} + u_{xs},\ u_y = u_{ys},\ \omega = \omega_s \tag{4.20}$$

where $(u_{xs},\ u_{ys},\ \omega_s)$ is the deviation from the operating point and the steering angle δ is small. Linearizing Eqs. (4.8)-(4.10), we obtain the following linearized dynamics:

$$m\dot{u}_{xs} = f_{xr} \tag{4.21}$$

$$m\dot{u}_{ys} = -c_r\frac{u_{ys} - b\omega_s}{u_{x0}} - c_f\frac{u_{ys} + a\omega_s}{u_{x0}} + c_f\delta - m\omega_s u_{x0} \tag{4.22}$$

$$I\dot{\omega}_s = -ac_f\frac{u_{ys} + a\omega_s}{u_{x0}} + ac_f\delta + bc_r\frac{u_{ys} - b\omega_s}{u_{x0}} \tag{4.23}$$

Furthermore, assuming that the angle ϕ is small and substituting Eqs. (4.17)-(4.19) after linearization, controller Eqs. (4.11) and (4.12) and distance Eq. (4.15) and (4.16) into

Eqs. (4.21)- (4.23) yields the following dynamics:

$$\ddot{y} = \frac{f_{xr}}{m} \tag{4.24}$$

$$\ddot{x} = -\frac{c_r + c_f}{mu_{x0}}\dot{x} - \frac{2c_f \sum_{i=1}^{N} p_i l_i}{m}x + \left(u_{x0} - \frac{bc_r - ac_f}{mu_{x0}}\right)\dot{\phi} + \frac{2c_f \sum_{i=1}^{N} p_i l_i}{m}\phi \tag{4.25}$$

$$\ddot{\phi} = -\frac{ac_f - bc_r}{Iu_{x0}}\dot{x} - \frac{2ac_f \sum_{i=1}^{N} p_i l_i}{I}x - \frac{a^2 c_f + b^2 c_r}{Iu_{x0}}\dot{\phi} - \frac{2ac_f \sum_{i=1}^{N} p_i l_i}{I}\phi \tag{4.26}$$

where we assume the desired steering angle $\delta_d = 0$. The breaking force effect appears as in Eq. (4.24), and it is obvious that the motion is decelerated when f_{xr} is negative. Because the control objective is to reduce the deviation in the x and ϕ directions, we only consider Eqs. (4.25) and (4.26) for the stability analysis. It should be noted that owing to the nonholonomic constraint, the robot can not move instantaneously in the x direction. Hence, we only consider the ϕ-dynamics in Eq. (4.26) for the stability analysis.

To validate the proposed method, we simply consider the case that the vehicle's front and rear sides and the cornering stiffness satisfy $a \simeq b$ and $c_f \simeq c_r$, respectively. Then, we can rewrite eq. (4.26) as follows:

$$\ddot{\phi} + c_1\dot{\phi} + c_2\phi = c_3 x, \tag{4.27}$$

where $c_1 \sim c_3$ are positive constants. Eq. (4.27) is a stable system with respect to ϕ. In addition, the positive value of x provides a positive steady state value for ϕ, which makes the robot turn left and reduces the magnitude of x. Similarly, when x has a negative value, the negative steady state value for ϕ causes the robot to turn right and reduces the magnitude of x. Hence, this approach is expected to provide the appropriate collision avoidance function.

4.3 Experiments

Figure 4.3 shows the experimental robot equipped with distance sensors and the controller for human operators. The measurable range of the distance sensor is 10 - 80 [cm]. Rotary encoders (500 [PPR]) attached to the motors are used for measuring the robot position and orientation by assuming that the wheel slip is negligible. The proposed controller design is verified in the environment shown in Figure 4.4, where the robot controlled by six operators is expected to move from the start position to the destination.

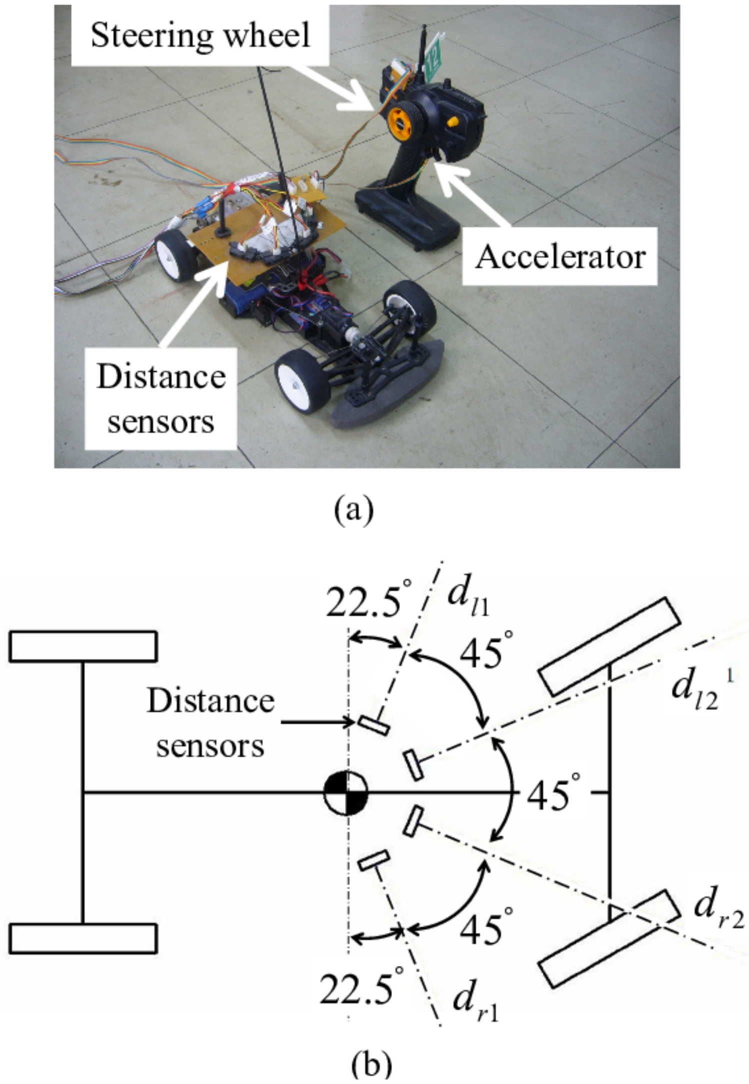


Figure 4.3: Experimental robot system

In order to achieve the effective collision avoidance, it is reasonable to employ a function that provides a larger steering angle and breaking force near obstacles compared with Eqs. (4.13) and (4.14). We consider the following nonlinear functions instead of Eqs. (4.13) and (4.14).

$$g_{ki} = \frac{\alpha_i}{\sqrt[n]{d_{ki}}}, k = l, r \tag{4.28}$$

$$h_{ki} = \frac{\beta_i}{\sqrt[n]{d_{ki}}}, k = l, r \tag{4.29}$$

Figure 4.5 shows the profile of these functions in which the parameters are set as $\alpha_i = 0.5$ and $n = 1, 2, 5$. Regarding to the stability analysis, linearizing Eqs. (4.28) and (4.29)

Table 4.1: Experimental Parameters.

Parameter	Value	Parameter	Value
m	$1.33\ \left(\mathrm{kg}\cdot\mathrm{m}^2\right)$	α_1	$2.0\times10^{-3}\ \left(\mathrm{rad}\cdot\mathrm{m}^3\right)$
I	$0.02\ \left(\mathrm{kg}\cdot\mathrm{m}^2\right)$	α_2	$4.0\times10^{-3}\ \left(\mathrm{rad}\cdot\mathrm{m}^3\right)$
a	0.09 (m)	β_1	$0.4\ \left(\mathrm{N}\cdot\mathrm{m}^3\right)$
b	0.07 (m)	β_2	$0.5\ \left(\mathrm{N}\cdot\mathrm{m}^3\right)$
C_f	15.0 (N/rad)	n	3
C_r	15.0 (N/rad)		

Table 4.2: Number of collisions occurred for each operator

Operator No	Manual	Proposed
1	3	0
2	2	0
3	1	0
4	3	0
5	1	0
6	2	0

leads to Eqs. (4.13) and (4.14), and hence the linear analysis assuming a certain operating point in Section 2.3 is still valid for linearized equations of Eqs. (4.28) and (4.29). Table 4.1 lists the parameters used in the experiment. Each operator operates the robot under the following conditions:

(a1) Without the collision avoidance function (if the robot collides with the wall, the operator operates the robot from the start position again).

(a2) With the collision avoidance function.

In case (a2), we consider the worst case that the operator can control only on/off of the translational motion, and the breaking and the steering are controlled automatically.

Table 4.2 shows the number of collisions for each operator. No collisions occurred while operating the robot with the collision avoidance function.

Figure 4.6 compares the time required for each operator to reach the goal. Because the robot collided the wall during the trial of all operators, they performed several trials and the required time was reduced in the last trial. The Figure 4.6 shows the required time recorded in the last trial of the manual control case. On an average, there is no significant difference in the required time to reach the goal with and without the proposed method.

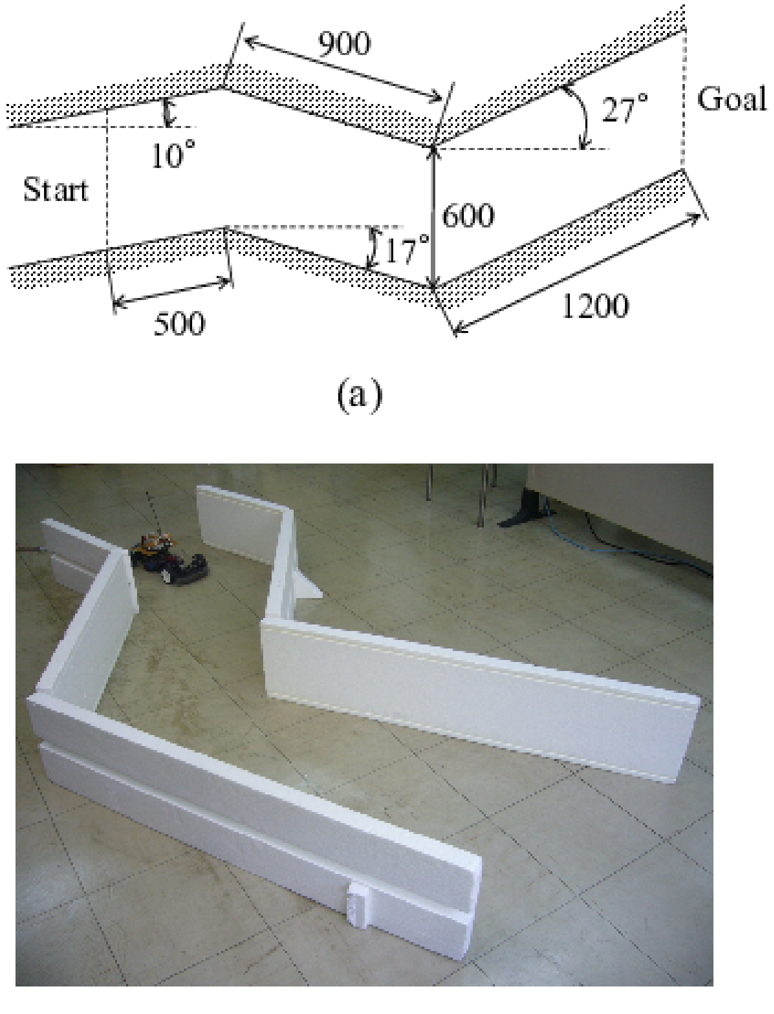
The average times were 9.42 s for the manual control case and 10.12 s for the proposed method.

Figures 4.7 and 4.8 compare the operator control with and without the proposed method. Figures 4.7(a) and 4.8(a) show the control input voltage commanded by the operator. The control input voltage has the following relation to the steering angle δ_d and acceleration force f_d in Eqs. (4.11) and (4.12), respectively:

$$\delta_d = \frac{\pi}{180} \times \{16.2 \times (V_\delta - V_0)\} \text{ [rad]} \tag{4.30}$$

$$f_d = 0.23 \times (V_f - V_0) \text{ [N]} \tag{4.31}$$

where V_δ and V_f are control input voltages commanded by the human operators and



(a)

(b)

Figure 4.4: Experimental environment

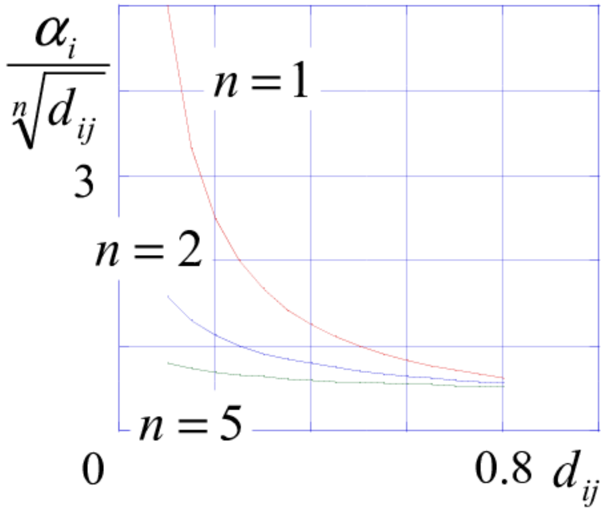


Figure 4.5: Function profile used for collision avoidance

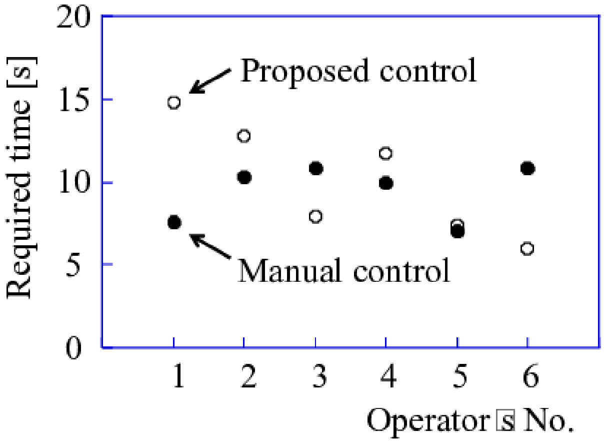
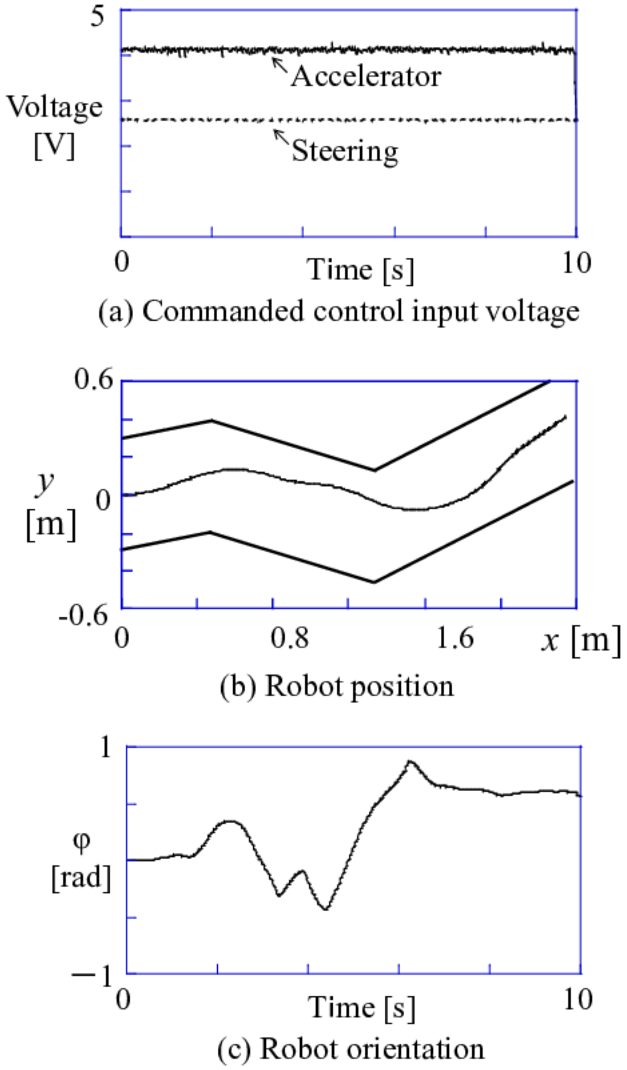


Figure 4.6: Comparison of required time to reach the goal

$V_0 = 2.65$ [V]. In Figure 4.7, although the operator does not steer the robot, it successfully moves to the goal by automatically adjusting the steering angle ϕ. In Figure 4.8, the operator frequently adjusts both the steering wheel and accelerator. However, a collision occurs at approximately $x = 1.8$ [m] in Figure 4.8(b). These results confirm the effectiveness of the proposed controller design using inexpensive distance sensors and simple control input calculations.

Experimental results show that the proposed control in Eqs. (4.11) and (4.12) can provide successful collision avoidance for the worst case that the operator can control only on/off of the translational motion. For the case that the operator can control the speed and the steering, the effect of functions g_{ki} and h_{ki} in Eqs. (4.11) and (4.12) may be tuned by changing the values α_i and n in Fig. 4.5. If the operator is skillful, the effect



(a) Commanded control input voltage

(b) Robot position

(c) Robot orientation

Figure 4.7: Proposed control results. (a) Commanded control input voltage, (b) Robot position, and (c) Robot orientation

should be reduced, otherwise, it should be increased. Hence, the proposed control may be useful for any level of operators in collision avoidance.

4.4 Conclusions

This chapter presents a new approach to collision avoidance for four-wheeled human-operated mobile robots using inexpensive IR proximity sensors. Because the proposed method considers the nonholonomic constraint of a mobile robot, it provides practical collision avoidance control. The effectiveness of the proposed approach is demonstrated by the results of the experiment, in which all unskilled operators could manoeuvre the

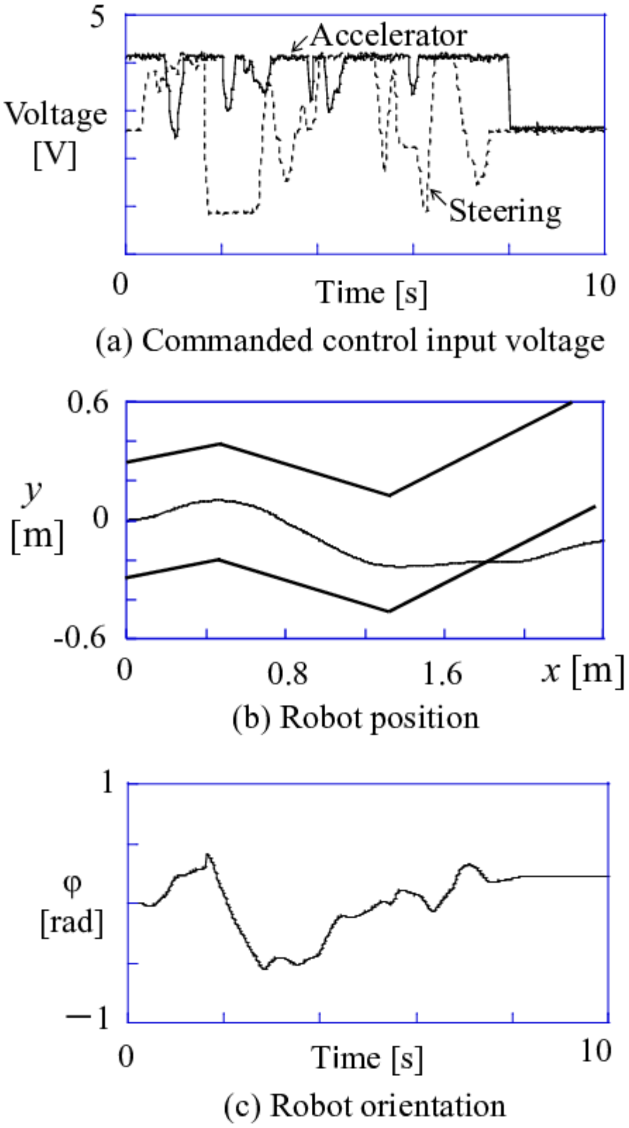


Figure 4.8: Manual control results. (a) Commanded control input voltage, (b) Robot position, and (c) Robot orientation

robot to the destination without collisions. In future studies, the presented linear analysis will be extended to more general cases and the proposed robot system will be applied to more complex environments.

Chapter 5

Vision-based Object Recognition and Tracking Control of a Service Mobile Robot

5.1 Introduction

The application range of robotics as a service robot in a human environment is getting wider as we are able to combine more sensors in a single robot. This improves the performance of the robot system, and also takes advantage of the reduced cost of sensors [16][17][71]. A service robot that operates either semi or fully autonomously to perform services for a human in daily life must be able to continuously track the human. This tracking is made possible by the application of a visual sensor and range finder [2][5][11][24][66] [73][75].

Many researchers who investigated human detection and tracking mainly focused on a vision-based method since it can be used for environmental sensing without physical contact [21]-[25][26]. Vision robot control is a feedback control method that uses one or more visual sensors to control the motion of the robot [5][6][110]. The control inputs for the robot actuators are produced by processing image data relative to a target object [27][57].

Vision, as our most powerful sense, provides us with an enormous amount of information and enables intelligent interaction with a dynamic environment. Therefore, considerable effort has been devoted to develop a robot with sensors that mimic the capabilities of the human vision system. The first application of a vision system was based on the stereo vision of a pin-hole camera fitted to a mobile robot or the end-effector of an arm robot [13][15]. This application has improved rapidly of late as the recent technology enables the combination of multiple camera features such as IR sensors, image detection, and depth measurement [11][12][13][22][27][62].

Visual odometry is a method of estimating the motion of a robot only by visual input. The basic principle involves conducting iterations of two structures resulted from motions using a stereo camera [13]. A stereo camera can be substituted by a single RGB-D (Red Green Blue-Depth) camera [22], since the estimation of both distance to the detected object and motion can be predicted and measured using depth mapping [23][27][28][29] [111].

One of the most used RGB-D cameras is the Kinect sensor, a motion sensing device developed by Microsoft Corporation. The advantages of the Kinect sensor are that it not only enables users to develop open-source drivers and programs but is also relatively low in cost. Using the Kinect sensor, the detection of images, distance, and depth sensing is easier compared with other types of camera [21][23]-[30].

The potential field method for path planning is effective because of its simplicity, and its basic mechanism is borrowed from nature (e.g., the principle of particle navigation in a magnetic field). In robotics, this phenomenon is emulated by developing an artificial potential field that attracts the robot to the target. The visual potential field is developed by setting the robot navigation to a sequence of images and not by only referring to the configuration of the obstacles [40][58][59].

In this study, we combine the potential field method with visual sensing to create a human-following robot, i.e, the robot is enabled to track and follow an object attached to the human. The attached object in this study is a blue circle mark. The vision sensor mounted on the robot is a Kinect sensor, and the system is developed utilizing RGB image detection to detect the position of the blue circle mark and depth mapping for distance calculation.

This chapter related to the work presented in [112][113]. The algorithm applied is the same as in previous studies implements the potential field method for path planning. The method of setting the target position in chapter 3 is changed to object detection. The effectiveness of the proposed method is validated by developing a human-following robot system, wheelchair following system, leader robot following system, and extending the experiment into the hallway outside the laboratory testbed environmental system.

5.2 Developed System Overview

This chapter presents a vision robot control system for a human and wheelchair following robot. This is done by creating a desired trajectory and then moving the robot to track that desired trajectory. As an extension of previous chapter 3, the via points target is changed into a moving target detection and the target is set using a blue circle mark. The robot is driven to the mark using an attractive potential field while it avoids obstacles using a repulsive potential field.

The system developed is summarized as follows:

1. The target is set using a blue circle mark to be intercepted that is attached to a human target, wheelchair and leader robot in which the robot will follow continuously.
2. The RGB image is processed to identify the human target with the blue circle mark, depth mapping is used to calculate the distance from the robot to the human, and a camera is used to track the detected mark.
3. A desired (reference) trajectory is set by applying an attractive virtual force to the target, and a repulsion virtual force to the obstacles.
4. A proportional integral (PI) controller is used to move the robot through the desired trajectory.

5.3 Object Detection and Tracking

In this study, the object to be detected and tracked is a blue circle mark, and the algorithm is designed to keep the mark close to the center of the image plane of the camera.

5.3.1 Image Processing for Object Detection

Image processing is required for a robot to track a desired trajectory. The purpose of image processing in this study is to locate the projected target position on an image plane. The image processing for detection is conducted by filtering the blue color from

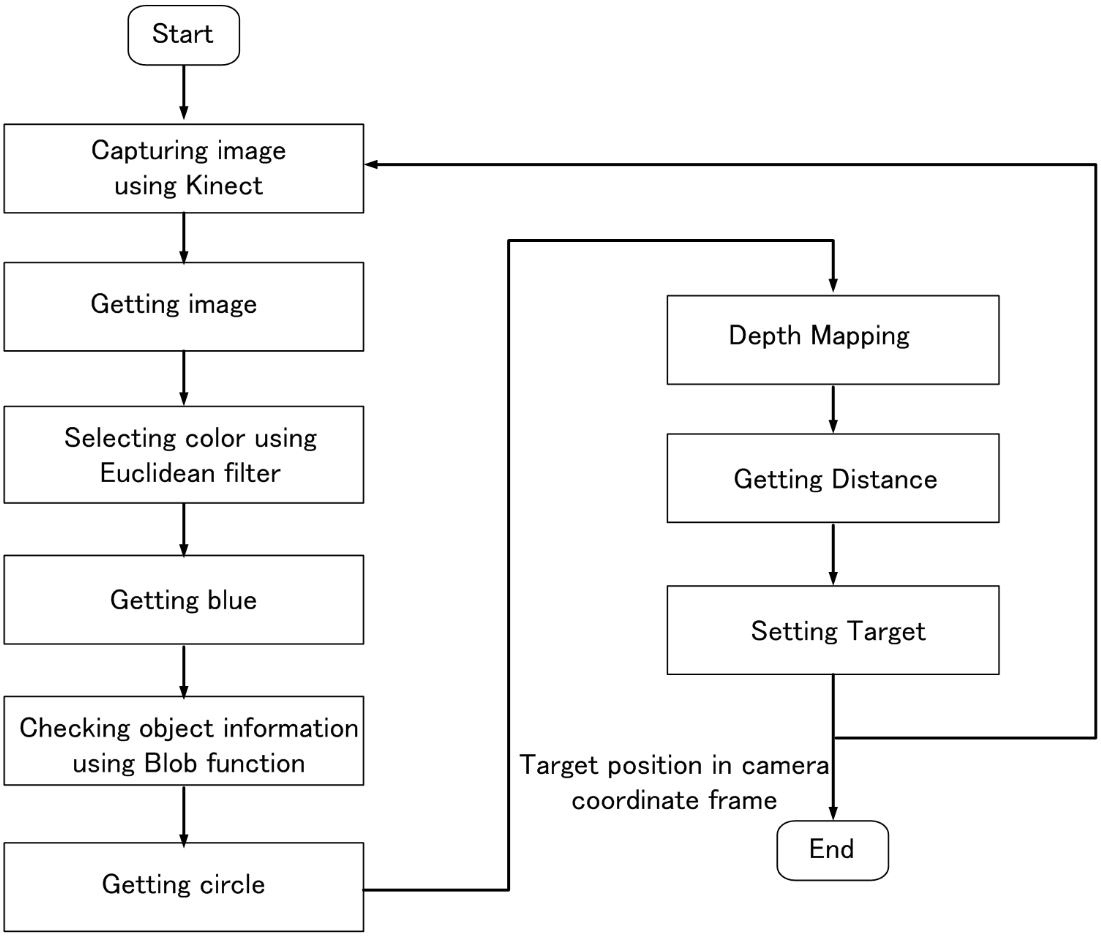

Start
Capturing image using Kinect
Getting image
Selecting color using Euclidean filter
Getting blue
Checking object information using Blob function
Getting circle
Depth Mapping
Getting Distance
Setting Target
Target position in camera coordinate frame
End

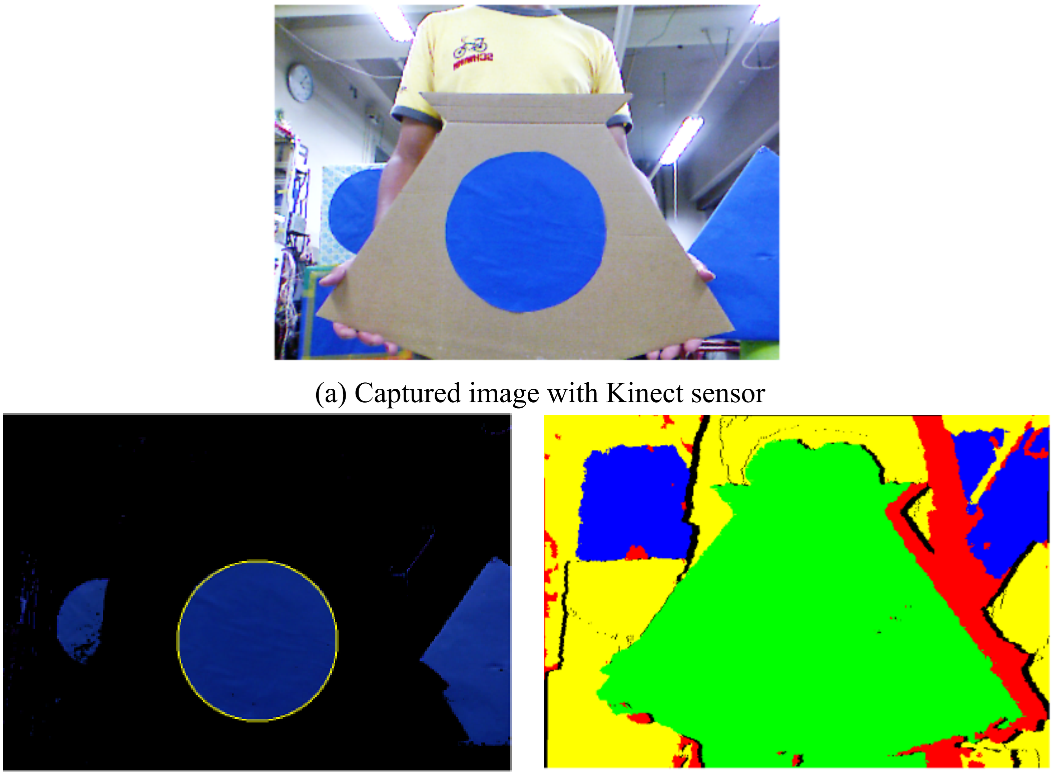
Figure 5.1: Blue circle detection flowchart

all the background colors, detecting and counting all the edge pixels to locate the circle mark, and finally obtaining the coordinates of the projected target on the image plane.

To obtain the distance between the camera and the blue circle mark, depth mapping is applied. Since the camera is mounted on the top of the robot, the distance between the robot and the blue circle mark is the same as the distance between the camera and the blue circle mark. The flowchart showing the steps of image processing and depth mapping for the blue circle mark is given in Figure 5.1. The result of the detected blue circle using image processing is shown in Figure 5.2b and the depth mapping is shown in Figure 5.2c.

5.3.2 Camera - Object Tracking Modeling

The reference trajectory for object tracking is obtained by applying an attractive potential field that is defined as a function of the relative distance between the robot and



(a) Captured image with Kinect sensor

(b) Image procesing result

(c) Depth mapping for target with distance 60cm from Kinect sensor

Figure 5.2: Image processing and depth mapping result

the target (a blue circle mark attached to a human target, wheelchair, or leader robot). The attractive potential field is then continuously updated and recalculated as the robot approaches the target.

The analysis for the blue circle mark position detection is performed using a perspective projection method, as shown in Figure 5.4, where P_m^c is the position of the blue circle mark, P_{pm}^c is the projected point of the blue circle mark in the image plane (the superscript (^{c}) indicating that the positions are relative to the camera coordinate frame), P_c^c is the position of the Kinect sensor (camera), and P_r^c is the position of the robot. The other notations in Figure 5.4 are ε (the focal length of the Kinect sensor), d_{act} (the actual distance from the camera to the blue circle mark), and d_{pm}(the distance from the camera to the image plane).

5.3.2.1 Analysis of Visual Servoing Methods

There are two basic ways to approach the problem of vision-based control and these are distinguished by the way how the data provided by the vision system is used. The first approach to vision-based control is known as position-based visual servo control. With this approach, the vision data is used to build a partial 3D representation of the world. The second method known as image-based visual servo control directly uses the image data to control the robot motion [5][6][8]. The most common approach has been to use easily detected points on an object as the feature points [56][63].

The error function is then the vector difference between the desired and measured locations of these points in the image [8]. The objectives of vision based control is to minimize error that is defined by

$$e(t) = P_m^c - \tilde{P}_m^c \tag{5.1}$$

P_m^c indicates the real vector image features being tracked and $\tilde{P}_m^c$ is the desired features of the image. If the single point is used as the projected image feature, then P_m^c can be defined in term of image plane coordinates of such

$$P_{pm}^c = \begin{bmatrix} x_{pm}^c \\ y_{pm}^c \end{bmatrix} \tag{5.2}$$

The time derivative of P_{pm}^c is the image feature velocity and linearly related to the camera velocity. If the camera velocity ($\dot{P}_c^c$) is represented by

$$\dot{P}_c^c = \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{5.3}$$

where v is the linear velocity of the origin of the camera and ω are the angular velocity of the camera about z axis of camera coordinate frame, then the relationship between the image feature velocity and the camera velocity is given by

$$\dot{P}_m^c = J_{pm}\left(P_m^c, P_c^c\right)\dot{P}_c^c \tag{5.4}$$

The matrix J is the image Jacobian matrix or interaction matrix as the function of image features and position of the camera. In order to derive the interaction matrix which relates the velocity of the camera ($\dot{P}_m^c$) to the time derivatives of the coordinates of the projection of a 3D fixed point of P_m^c in the image plane.

5.3.2.2 Velocity of a Fixed Point Relative to a Moving Camera

In this study the camera is moving or also called eye-in-hand system. The camera is mounted on the robot, therefore the motion of the robot does not affect the camera motion. In this way, the camera can observe the motion of object and without occlusion as the robot moves through the workspace. One of the disadvantage of this method is that the geometric relationship between the camera and the workspace changes as the robot moves. The field of view can change drastically for even small motion of the manipulator, particularly if the robot change orientation [5][6][8].

When considering the moving camera, it is necessary to find out an expression of the velocity of point P_m^c with respect to the moving camera. Using homogeneous transformation equations, the relationship between the coordinates of point P_m^c with respect to world frame and moving camera is derived below [8].

If P_m^w is the coordinate of point P_m^c relative to world coordinate frame and does not vary with time since P_m^w is fixed with respect to world coordinate frame. Then the relationship between the point P_m^c relative to the moving camera frame at time t is given by

$$P_m^w = R_c^w P_m^c + O_c^w \tag{5.5}$$

where R_c^w is the rotation matrix to transform from camera coordinates frame to world coordinates frame and O_c^w is the origin of the camera coordinates frame relative to world coordinate frame.

The time derivative of $\dot{R}_c^w$ is given by

$$\dot{R}_c^w = S(\omega) R_c^w \tag{5.6}$$

where S is a skew symmetric matrix that can be represented as $S(\omega)$ for a unique vector ω. The vector ω is the angular velocity of the rotating frame with respect to the fixed frame at the time t.

Thus, at time t we can solve for the coordinates of P_m^c relative to the camera frame by

$$P_m^c = R_w^c P_m^w - R_w^c O_c^w \tag{5.7}$$

In the reminder of this section, to simplify notation we will define $R = R_w^c$ and $O = O_w^c$.

Now, to find the velocity of the P_m^c relative to the moving camera frame, we can differentiate eq. (5.7) as follow

$$\begin{aligned}\frac{d}{dt}P_m^c &= \frac{d}{dt}\left[R^T P_m^w\right] - \frac{d}{dt}\left[R^T O\right] \\ &= \frac{d}{dt}\left[R^T\right]P_m^w - \frac{d}{dt}\left[R^T\right]O - R^T\frac{d}{dt}O \\ &= \frac{d}{dt}\left[R\right]^T\left(P_m^w - O\right) - R^T\dot{O}\end{aligned} \tag{5.8}$$

where $(P_m^w - O)$ is the vector from the origin of the moving frame to the fixed point P_m^w, expressed in coordinates relative to the fixed frame, and thus $R^T\left(P_m^w - O\right) = P_m^c$ is the vector from the origin of the moving frame to the point P_m^c expressed relative to the moving frame.

By using eq. (5.6), we can write eq. (5.8) as

$$\begin{aligned}\dot{P}_m^c &= \left[S(\omega)R\right]^T\left(P_m^w - O\right) - R^T\dot{O} \\ &= R^T S^T(\omega)\left(P_m^w - O\right) - R^T\dot{O} \\ &= R^T\left[S(-\omega)\left(P_m^w - O\right)\right] - R^T\dot{O} \\ &= -R^T\omega \times R^T\left(P_m^w - O\right) - R^T\dot{O}\end{aligned} \tag{5.9}$$

The vector ω gives the angular velocity vector for the moving frame expressed relative to the fixed frame, i.e., $\omega = \omega^w$. Therefore, $R^T\omega = R_w^{c\,T}\omega^w = \omega_m^c$ gives the angular velocity of the moving frame. Similarly, note that $R^T\dot{O} = \dot{O}_m^c$ gives the translational velocity of the moving frame.

Using this derivation, we can write the equation for the velocity of $\dot{P}_m^c$ relative to moving camera frame

$$\dot{P}_m^c = -\omega_m^c \times P_m^c - v_m^c \tag{5.10}$$

where $\dot{P}_m^c = \begin{bmatrix}\dot{x}_m^c & \dot{y}_m^c & \dot{z}_m^c\end{bmatrix}^T$. By denoting that the coordinates for the angular velocity vector by $\omega_m^c = \begin{bmatrix}\omega_{mx}^c & \omega_{my}^c & \omega_{mz}^c\end{bmatrix}^T = R_w^{c\,T}\omega_w^c$ and $\dot{O}_m^c = \begin{bmatrix}v_{mx}^c & v_{my}^c & v_{mz}^c\end{bmatrix}^T$, we can write eq. (5.10)

$$\begin{bmatrix}\dot{x}_m^c \\ \dot{y}_m^c \\ \dot{z}_m^c\end{bmatrix} = -\begin{bmatrix}\omega_{mx}^c \\ \omega_{my}^c \\ \omega_{mz}^c\end{bmatrix} \times \begin{bmatrix}x_m^c \\ y_m^c \\ z_m^c\end{bmatrix} - \begin{bmatrix}v_{mx}^c \\ v_{my}^c \\ v_{mz}^c\end{bmatrix} \tag{5.11}$$

which can be written as the system of three equations

$$\begin{aligned}\dot{x}^c_m &= y^c_m \omega^c_{mz} - z^c_m \omega^c_{my} - v^c_{mx}\\ \dot{y}^c_m &= z^c_m \omega^c_{mx} - x^c_m \omega^c_{mz} - v^c_{my}\\ \dot{z}^c_m &= x^c_m \omega^c_{my} - y^c_m \omega^c_{mx} - v^c_{mz}\end{aligned} \tag{5.12}$$

5.3.2.3 Velocity of a Moving Point Relative to a Fixed Camera

With a fixed camera configuration, the camera is positioned so that it can observe the robot and any objects around the robot. One of the advantages of this approach is that since the camera position is fixed, the field of view does not change as the robot moves. The geometric relationship between the camera and the workspace is fixed, and can be calibrated offline. The disadvantage to this approach is that the robot moves through the workspace, it can occlude the camera's field of view.

This study requires the determination of the velocity of a moving point relative to a fixed camera. However, as mentioned at the previous subsection, the use of a camera requires us to interpret the motion of a 3D object through 2D images provided by the camera. The primary problem is that 3D information has to be compressed and nonlinearly transformed into 2D transformation; hence, some techniques has to be developed to obtained 3D information despite the fact that only 2D information is available.

In this study, we apply an algorithm for a moving point relative to a moving camera. In previous subsection, we derive the velocity of a fixed point relative to a moving camera as mentioned in [5][8], and in this subsection, we derive the velocity of a moving point relative to a fixed camera. The relationship between the velocity of projected moving object in camera coordinates plane ($\dot{P}^c_w$) and the velocity of the robot in world coordinates frame ($\dot{P}^w_m$) is given by

$$\dot{P}^c_m = R^c_w \dot{P}^w_m \tag{5.13}$$

where R^w_c is the rotation matrix as below

$$R^c_w = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \tag{5.14}$$

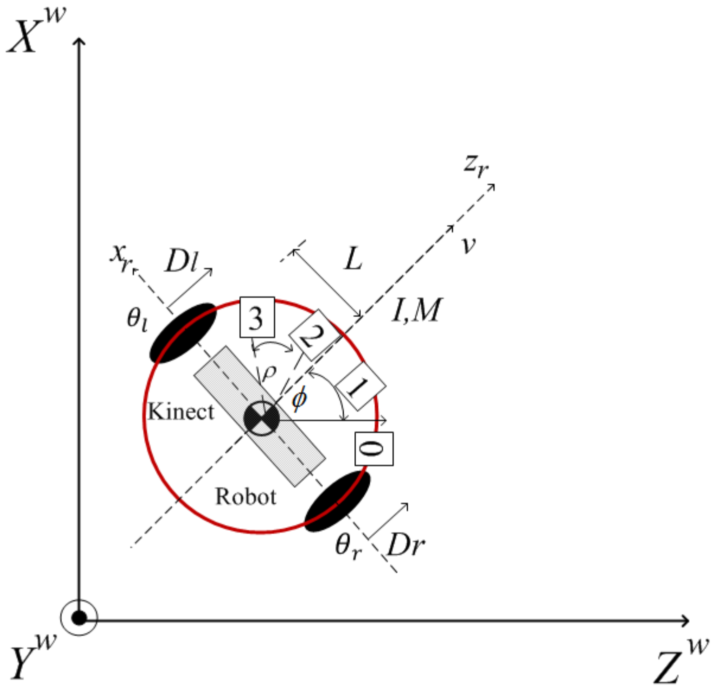


Figure 5.3: Robot position in camera coordinates frame relative to world coordinates frame

Eq. (5.13) can be rewritten as

$$\begin{bmatrix} \dot{x}_m^c \\ \dot{y}_m^c \\ \dot{z}_m^c \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} \dot{x}_m^w \\ \dot{y}_m^w \\ \dot{z}_m^w \end{bmatrix} \tag{5.15}$$

By using linear algebra, from eq. (5.15), we have the following relation

$$\begin{aligned} \dot{x}_m^c &= \dot{x}_m^w \cos\phi - \dot{z}_m^w \sin\phi \\ \dot{y}_m^c &= \dot{y}_m^w \\ \dot{z}_m^c &= \dot{x}_m^w \sin\phi + \dot{z}_m^w \cos\phi \end{aligned} \tag{5.16}$$

5.3.2.4 Velocity of a Moving Point Relative to a Moving Camera

This thesis considers the velocity of a moving point relative to a moving camera. In this subsection, we combine the derivation of the velocity of a moving point relative to fixed point and the velocity of a fixed point relative to a moving camera. The proposed algorithm relies on the availability of a reference frame of the target object, captured at a known orientation relative to the camera. For the general case where both the camera and the object are in motion relative to an inertial frame, a single geometric length on the object is assumed to be known. In this study, this information is provided by the depth information from a Kinect Sensor.

In order to get the Jacobian matrix for relating the moving camera and moving point, we combine eq. (5.12) and eq. (5.16), we have

$$\begin{aligned}
\dot{x}^c_m &= y^c_m \omega^c_{mz} - z^c_m \omega^c_{my} - v^c_{mx} + \dot{x}^w_m \cos\phi - \dot{z}^w_m \sin\phi \\
\dot{y}^c_m &= z^c_m \omega^c_{mx} - x^c_m \omega^c_{mz} - v^c_{my} + \dot{y}^w_m \\
\dot{z}^c_m &= x^c_m \omega^c_{my} - y^c_m \omega^c_{mx} - v^c_{mz} + \dot{x}^w_m \sin\phi + \dot{z}^w_m \cos\phi
\end{aligned} \tag{5.17}$$

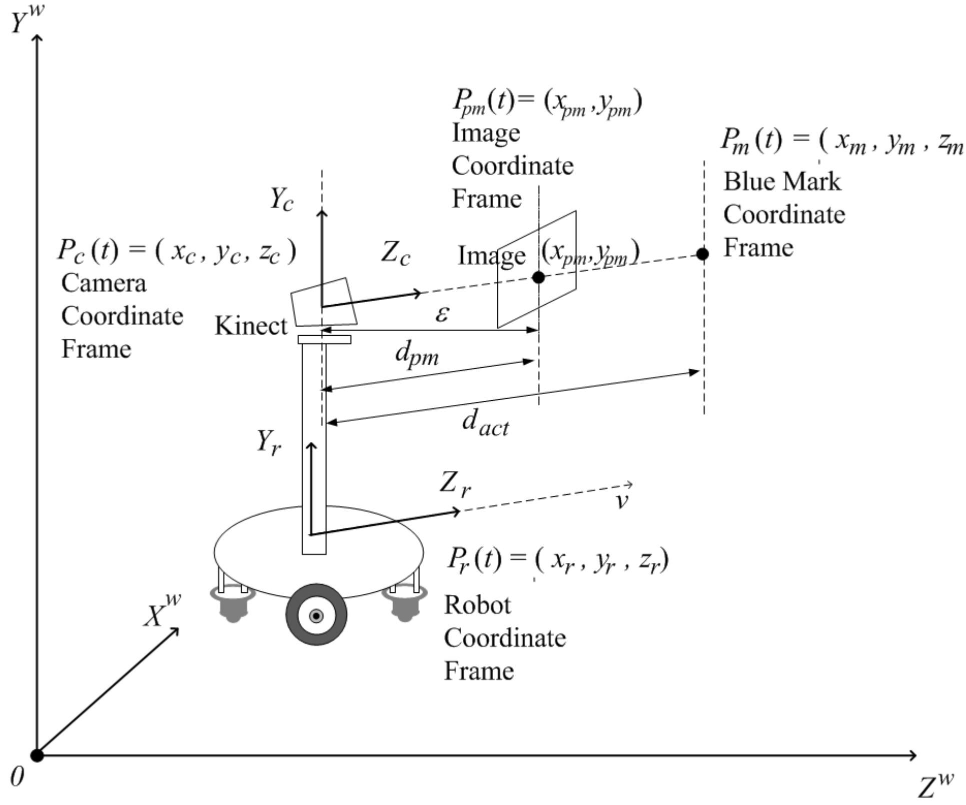


Figure 5.4: Robot and camera position in camera coordinates frame relative to world coordinates frame

From figure 5.4, we have the relation between the target pose in target coordinate frame (P_m) and the projection of target pose in image plane (P_{pm}) below

$$\begin{aligned} &\varsigma = \frac{\varepsilon}{z^c_m} & & \\ &\varepsilon x^c_m = x^c_{pm} z^c_m & &\varepsilon y^c_m = y^c_{pm} z^c_m \\ &x^c_m = \frac{x^c_{pm} z^c_m}{\varepsilon} & &y^c_m = \frac{y^c_{pm} z^c_m}{\varepsilon} \end{aligned} \tag{5.18}$$

where ς is the unknown positive constant.

By substituting eq. (5.18) into eq. (5.17), we have

$$\begin{aligned} \dot{x}^c_m &= \frac{y^c_{pm} z^c_m}{\varepsilon} \omega^c_{mz} - z^c_m \omega^c_{my} - v^c_{mx} + \dot{x}^w_m \cos\phi - \dot{z}^w_m \sin\phi \\ \dot{y}^c_m &= z^c_m \omega^c_{mx} - \frac{x^c_{pm} z^c_m}{\varepsilon} \omega^c_{mz} - v^c_{my} + \dot{y}^w_m \\ \dot{z}^c_m &= \frac{x^c_{pm} z^c_m}{\varepsilon} \omega^c_{my} - \frac{y^c_{pm} z^c_m}{\varepsilon} \omega^c_{mx} - v^c_{mz} + \dot{x}^w_m \sin\phi + \dot{z}^w_m \cos\phi \end{aligned} \tag{5.19}$$

By differentiating eq.(5.18), we have the velocity of the projected target in image plane. The velocity with respect to x-axis in camera coordinate frame is given by

$$\dot{x}^c_{pm} = \frac{d}{dt}\left(\varepsilon \frac{x^c_m}{z^c_m}\right) = \frac{\varepsilon}{(z^c_m)^2}\left(z^c_m \dot{x}^c_m - x^c_m \dot{z}^c_m\right) \tag{5.20}$$

$$\begin{aligned} \dot{x}^c_{pm} = &\frac{\varepsilon}{(z^c_m)^2}\left[z^c_m\left(\frac{y^c_{pm} z^c_m}{\varepsilon}\omega^c_{mz} - z^c_m \omega^c_{my} - v^c_{mx} + \dot{x}^w_m \cos\phi - \dot{z}^w_m \sin\phi\right)\right] \\ &- \frac{\varepsilon}{(z^c_m)^2}\left[x^c_m\left(\frac{x^c_{pm} z^c_m}{\varepsilon}\omega^c_{my} - \frac{y^c_{pm} z^c_m}{\varepsilon}\omega^c_{mx} - v^c_{mz} + \dot{x}^w_m \sin\phi + \dot{z}^w_m \cos\phi\right)\right] \end{aligned}$$

$$\begin{aligned} \dot{x}^c_{pm} = &\frac{\varepsilon}{(z^c_m)^2}\left[\frac{y^c_{pm}\left(z^c_m\right)^2}{\varepsilon}\omega^c_{mz} - \left(z^c_m\right)^2 \omega^c_{my} - z^c_m v^c_{mx} + z^c_m \dot{x}^w_m \cos\phi - z^c_m \dot{z}^w_m \sin\phi\right] \\ &- \frac{\varepsilon}{(z^c_m)^2}\left[\frac{\left(x^c_{pm}\right)^2 z^c_m}{\varepsilon}\omega^c_{my} - \frac{x^c_m y^c_{pm} z^c_m}{\varepsilon}\omega^c_{mx} - x^c_m v^c_{mz} + x^c_m \dot{x}^w_m \sin\phi + x^c_m \dot{z}^w_m \cos\phi\right] \end{aligned}$$

$$\begin{aligned}\dot{x}^c_{pm} = &\left[y^c_{pm}\omega^c_{mz} - \varepsilon\omega^c_{my} - \frac{\varepsilon}{z^c_m}v^c_{mx} + \frac{\varepsilon}{z^c_m}\dot{x}^w_m\cos\phi - \frac{\varepsilon}{z^c_m}\dot{z}^w_m\cos\phi\right] \\ &- \left[\frac{x^c_m x^c_{pm}}{z^c_m}\omega^c_{my} - \frac{x^c_m x^c_{pm}}{z^c_m}\omega^c_{mx} - \frac{\varepsilon x^c_m}{\left(z^c_m\right)^2}v^c_{mz} + \frac{\varepsilon x^c_m \dot{x}^w_m}{\left(z^c_m\right)^2}\sin\phi + \frac{\varepsilon x^c_m \dot{z}^w_m}{\left(z^c_m\right)^2}\cos\phi\right]\end{aligned}$$

$$\begin{aligned}\dot{x}^c_{pm} = &- \frac{\varepsilon}{z^c_m}v^c_{mx} + 0v^c_{my} + \frac{\varepsilon x^c_m}{\left(z^c_m\right)^2}v^c_{mz} + \frac{x^c_m y^c_{pm}}{z^c_m}\omega^c_{mx} - \frac{\varepsilon^2 + x^c_{pm}}{\varepsilon}\omega^c_{my} + y^c_{pm}\omega^c_{mz} \\ &+ \left(\frac{\varepsilon}{z^c_m}\cos\phi - \frac{\varepsilon x^c_m}{\left(z^c_m\right)^2}\sin\phi\right)\dot{x}^w_m - \left(\frac{\varepsilon}{z^c_m}\sin\phi + \frac{\varepsilon x^c_m}{\left(z^c_m\right)^2}\cos\phi\right)\dot{z}^w_m\end{aligned} \tag{5.21}$$

With the same manner, we have the velocity with respect to *y*-axis in camera coordinate frame as below

$$\dot{y}^c_{pm} = \frac{d}{dt}\left(\varepsilon\frac{y^c_m}{z^c_m}\right) = \frac{\varepsilon}{\left(z^c_m\right)^2}\left(z^c_m\dot{y}^c_m - y^c_m\dot{z}^c_m\right) \tag{5.22}$$

$$\begin{aligned}\dot{y}^c_{pm} = &\frac{\varepsilon}{\left(z^c_m\right)^2}\left[z^c_m\left(z^c_m\omega^c_{mx} - x^c_m\omega^c_{mz} - v^c_{my} + \dot{y}^w_m\right)\right] \\ &- \frac{\varepsilon}{\left(z^c_m\right)^2}\left[y^c_m\left(x^c_m\omega^c_{my} - y^c_m\omega^c_{mx} - v^c_z + \dot{x}^w_m\sin\phi + \dot{z}^w_m\cos\phi\right)\right]\end{aligned}$$

$$\begin{aligned}\dot{y}^c_{pm} = &\frac{\varepsilon}{\left(z^c_m\right)^2}\left[z^c_m\left(z^c_m\omega^c_{mx} - \frac{x^c_{pm}z^c_m}{\varepsilon}\omega^c_{mz} - v^c_{my} + \dot{y}^w_m\right)\right] \\ &- \frac{\varepsilon}{\left(z^c_m\right)^2}\left[y^c_m\left(\frac{x^c_{pm}z^c_m}{\varepsilon}\omega^c_{my} - \frac{y^c_{pm}z^c_m}{\varepsilon}\omega^c_{mx} - v^c_{mz} + \dot{x}^w_m\sin\phi + \dot{z}^w_m\cos\phi\right)\right]\end{aligned}$$

$$\begin{aligned}\dot{y}^c_{pm} = &\frac{\varepsilon}{\left(z^c_m\right)^2}\left[\left(z^c_m\right)^2\omega^c_{mx} - \frac{x^c_{pm}\left(z^c_m\right)^2}{\varepsilon}\omega^c_{mz} - z^c_m v^c_{my} + z^c_m\dot{y}^w_m\right] \\ &- \frac{\varepsilon}{\left(z^c_m\right)^2}\left[\frac{x^c_{pm}y^c_m z^c_m}{\varepsilon}\omega^c_{my} - \frac{y^c_{pm}y^c_m z^c_m}{\varepsilon}\omega^c_{mx} - v^c_{mz} + y^c_m\dot{x}^w_m\sin\phi + y^c_m\dot{z}^w_m\cos\phi\right]\end{aligned}$$

$$
\begin{aligned}
\dot{y}^c_{pm} = {} & \left[\varepsilon \omega^c_{mx} - x^c_{pm} \omega^c_{mz} - \frac{\varepsilon}{z^c_m} v^c_{my} + \frac{\varepsilon \dot{y}^w_m}{z^c_m} \right] \\
& - \left[\frac{x^c_{pm} y^c_m}{z^c_m} \omega^c_{my} - \frac{y^c_{pm} y^c_m}{z^c_m} \omega^c_{mx} - \frac{\varepsilon y^c_m}{(z^c_m)^2} v^c_{mz} + \frac{\varepsilon y^c_m \dot{x}^w_m}{(z^c_m)^2} \sin\phi + \frac{\varepsilon y^c_m \dot{z}^w_m}{(z^c_m)^2} \cos\phi \right]
\end{aligned}
$$

$$
\begin{aligned}
\dot{y}^c_{pm} = {} & 0 v^c_{mx} - \frac{\varepsilon}{z^c_m} v^c_{my} + \frac{y^c_{pm}}{z^c_m} v^c_{mz} + \frac{\varepsilon^2 + y^c_{pm}}{\varepsilon} \omega^c_{mx} - \frac{x^c_{pm} y^c_m}{z^c_m} \omega^c_{my} - x^c_{pm} \omega^c_{mz} \\
& + \frac{\varepsilon y^c_m \dot{x}^w_m}{(z^c_m)^2} \sin\phi + \frac{\varepsilon}{z^c_m} + \frac{\varepsilon y^c_m \dot{x}^w_m}{(z^c_m)^2} \cos\phi
\end{aligned}
\tag{5.23}
$$

5.3.3 Moving Object Detection and Tracking

In this study, the object to be detected and tracked is a blue circle mark, and the algorithm is designed to keep the mark close to the center of the image plane of the camera. In this study, we apply the assumptions below:

1. The Kinect sensor is mounted on the top of the robot and the blue circle mark is attached to a human target, wheelchair, or leader robot; therefore, both the camera and the mark are moving, and for that we are using the analysis of a moving point relative to a moving camera.
2. The image formation is modeled by the pinhole lens approximation, where the lens is considered to be an ideal pinhole located in the focal center of the lens.
3. Under the pinhole assumption, P^c_m , P^c_{pm} , and the origin of the camera coordinates frame are collinear, as shown in figure 5.4.

The next step to get the relation between the robot velocity and target velocity is by deriving the image Jacobian matrixes J_{pm}. From eq. (5.4), the image Jacobian matrix is defined. It is a nonlinear map between the moving frame and the robot's coordinates in the camera image. Eq. (5.4) provide an interface for robot velocity inputs, most image based visual control strategies are based on the relation of the robot's velocity and the image features. By rearranging eq. (5.21) and eq. (5.23) in the matrix form, we have

$$\begin{bmatrix} \dot{x}^c_{pm} \\ \dot{y}^c_{pm} \end{bmatrix} = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} v^c_{mx} \\ v^c_{my} \\ v^c_{mz} \\ \omega^c_{mx} \\ \omega^c_{my} \\ \omega^c_{mz} \\ \dot{x}^w_m \\ \dot{y}^w_m \\ \dot{z}^w_m \end{bmatrix} \tag{5.24}$$

where

$$A = \begin{bmatrix} -\frac{\varepsilon}{z^c_m} & 0 & \frac{x^c_{pm}}{z^c_m} & \frac{x^c_{pm} y^c_{pm}}{\varepsilon} & -\left(\frac{\varepsilon^2 + \left(x^c_{pm}\right)^2}{\varepsilon}\right) & y^c_{pm} \\ 0 & -\frac{\varepsilon}{z^c_m} & \frac{y^c_{pm}}{z^c_m} & \frac{\varepsilon^2 + \left(y^c_{pm}\right)^2}{\varepsilon} & -\frac{x^c_{pm} y^c_{pm}}{\varepsilon} & -x^c_{pm} \end{bmatrix}$$

and

$$B = \begin{bmatrix} \frac{\varepsilon}{z^c_m}\cos\phi - \frac{x^c_{pm}}{z^c_m}\sin\phi & 0 & -\left(\frac{\varepsilon}{z^c_m}\sin\phi + \frac{x^c_{pm}}{z^c_m}\cos\phi\right) \\ \frac{y^c_{pm}}{z^c_m}\sin\phi & \frac{\varepsilon}{z^c_m} & \frac{y^c_{pm}}{z^c_m}\cos\phi \end{bmatrix}$$

The first three columns are dependent on the image plane coordinate $\left(x^c_{pm}, y^c_{pm}\right)$ and the depth, z, of the 3D point relative to the camera frame. Since the velocity of the moving camera is $v\left(t\right) = \begin{bmatrix} v_x & v_y \end{bmatrix}^T$ and $\dot{\phi}\left(t\right) = \omega_{my}$, the relationship between the velocity of the moving projected point $\begin{bmatrix} \dot{x}^c_{pm} & \dot{y}^c_{pm} \end{bmatrix}^T$ in the camera coordinate frame and the velocity of the robot $\begin{bmatrix} \dot{x}^w_m & \dot{y}^w_m & \dot{z}^w_m \end{bmatrix}^T$ in the world coordinate frame are given by:

$$\begin{bmatrix} \dot{x}^c_{pm} \\ \dot{y}^c_{pm} \end{bmatrix} = J_{pm}\left(x^c_{pm}, y^c_{pm}, z^c_m, \varepsilon, \phi\right) \begin{bmatrix} \dot{x}^c_r \\ \dot{y}^c_r \\ \dot{\phi} \\ \dot{x}^w_m \\ \dot{y}^w_m \\ \dot{z}^w_m \end{bmatrix} \tag{5.25}$$

J_{pm} is the image Jacobian given by:

$$J_{pm}\left(x_{pm}^{c}, y_{pm}^{c}, z_{m}^{c}, \varepsilon, \phi\right) = \begin{bmatrix} C & B \end{bmatrix} \tag{5.26}$$

where

$$C = \begin{bmatrix} -\frac{\varepsilon}{z_m^c} & \frac{x_{pm}^c}{z_{pm}^c} & -\left(\frac{\varepsilon^2+\left(x_{pm}^c\right)^2}{\varepsilon}\right) \\ 0 & \frac{y_{pm}^c}{z_m^c} & -\frac{x_{pm}^c y_{pm}^c}{\varepsilon} \end{bmatrix}$$

By considering the actual distance d_{act} in $+Z$ direction, we have

$$\begin{aligned} z_m^c &= \frac{d_{act}\,\varepsilon}{d_{pm}} \\ d_{pm} &= \sqrt{x_{pm}^2 + y_{pm}^2 + \varepsilon^2} \end{aligned} \tag{5.27}$$

Substituting eq. (5.27) into eq. (5.26), we have the following image Jacobian that considers the distance between the robot and the blue circle mark.

$$\bar{J}_{pm}\left(x_{pm}^{c}, y_{pm}^{c}, z_{m}^{c}, \varepsilon, d_{act}, d_{pm}\right) = \begin{bmatrix} E & F \end{bmatrix} \tag{5.28}$$

where

$$E = \begin{bmatrix} \frac{-d_{pm}}{d_{act}} & \frac{x_{pm}^c d_{pm}}{d_{act}\varepsilon} & -\left(\frac{\varepsilon^2+\left(x_{pm}^c\right)^2}{\varepsilon}\right) \\ 0 & \frac{y_{pm}^c d_{pm}}{d_{act}\varepsilon} & -\frac{x_{pm}^c y_{pm}^c}{\varepsilon} \end{bmatrix}$$

and

$$F = \begin{bmatrix} \frac{-d_{pm}}{d_{act}}\cos\phi - \frac{x_{pm}^c d_{pm}}{d_{act}\varepsilon}\sin\phi & 0 & -\left(\frac{d_{pm}}{d_{act}}\sin\phi + \frac{x_{pm}^c d_{pm}}{d_{act}\varepsilon}\cos\phi\right) \\ \frac{y_{pm}^c d_{pm}}{d_{act}\varepsilon}\sin\phi & \frac{d_{pm}}{d_{act}} & \frac{y_{pm}^c d_{pm}}{d_{act}\varepsilon}\cos\phi \end{bmatrix}$$

5.4 Controller Design

This study applied two controllers, a Reference controller and PI-controller. The reference controller is derived and discussed thoroughly in chapter 3 and PI-controller applied is the same with PI-controller used in chapter 5.

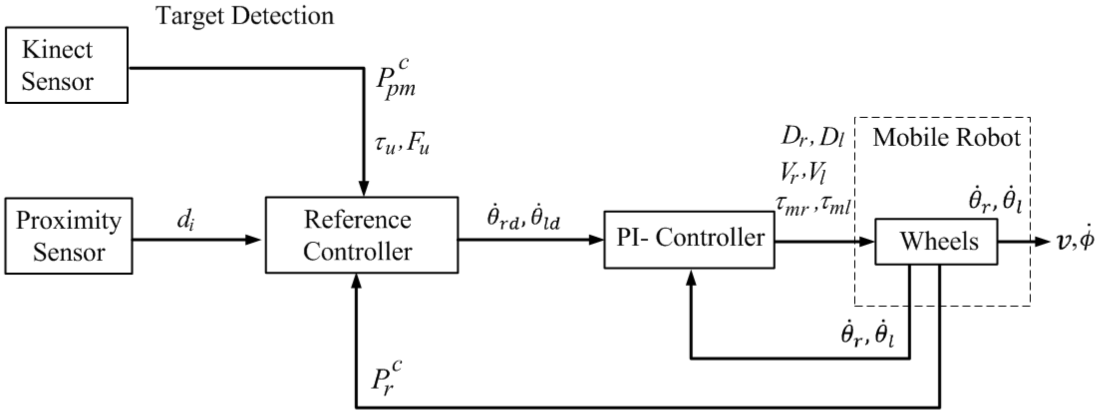


Figure 5.5: Block diagram of the proposed method

As in previous chapters, the potential field method is applied and the target creates the attractive force and the obstacle creates repulsive force. The reference controller generates the reference trajectory to reach the target and PI-controller moves the robot to follow the reference trajectory. Figure 5.5 shows the block diagram of the proposed method in this chapter.

5.5 Experiment

Experiment was conducted to validate the effectiveness of the proposed method. The robot is assigned to recognize and track the moving object. The moving object considered in this study is a blue circle mark of 25 cm diameter attached to human, wheelchair or leader robot. As the robot can recognize any blue circle, a large circle was used to ensure that the robot only follow the assigned one.

5.5.1 Experimental Environment Setting

The experiments related in this study are conducted in scenarios below

1. Obstacle free environment for moving target following system.

   The courses in this scenario is meant to test whether the robot can follow the moving object by not exposing the robot to the obstacle that we can focus only to the following algorithm.

- One human target is followed by the robot through three courses; a straight course, an L-shaped course and a crank course
- A wheel chair is followed by the robot through an L-shaped course. The wheelchair experiment is done only for one course scenario since other courses are already conducted in human-following and the effectiveness is confirmed.

2. Obstacle existence environment.

   This is the complete system of moving target following system by applying the obstacle avoidance algorithm at the same time with following algorithm.

   - Five human targets are followed through a crank course bordered by box functioning as obstacles. In this course, the obstacle avoidance is applied to ensure that the border and human target are not damaged.
   - Wheelchair is followed by the robot in L-shaped. The L-shaped course is first conducted for one human operating wheel chair, to see whether the system is also effective for the obstacle existence environment. The L-shaped course is relatively easy course for wheelchair operated by human to maneuver compared to crank course.
   - Wheel chair is followed by the robot in crank course. This scenario was done five times by different human operator.

3. Occlusion.

   In this study, occlusion is defined as the condition where the robot loses sight of the blue circle mark. This scenario is to prove that the robot gets the coordinate of the target position all the times (as if the robot sees the target all the times), therefore a brief occlusion will not affect the moving object following system.

   - One human target was followed in course where the human made a sudden turn causing the robot suffered a brief occlusion.
   - One human target was followed by the robot through a course while the robot is subjected to a brief occlusion by an obstacle.

4. Extended to Swarm Robot.

   This scenario is intended to test the possibility of creating swarm robot by using the proposed method. The robot is following the blue mark attached to another robot. In this scenario the movement of the leader robot creates brief occlusion to the robot making the moving object following system more challenging. Unfortunately, due to the limited availability of the robots, the system is only possible using one robot leader and one follower.

This system is also possible for a leader robot following a human target and a follower robot following the leader robot, however due to the limited test-bed space, this scenario cannot be applied. Instead, we applied the same courses applied to the previous scenario. The leader robot's trajectory is not applying the proposed method trajectory system.

- The robot follows the leader robot in L-shaped course.
- The robot follows the leader robot in crank course.

5. Extended to the hall way

   This scenario is conducted to prove that the moving target following system can be extended outside the laboratory test-bed area. The robot is set to follow a wheelchair in a hallway shown in figure 5.20.

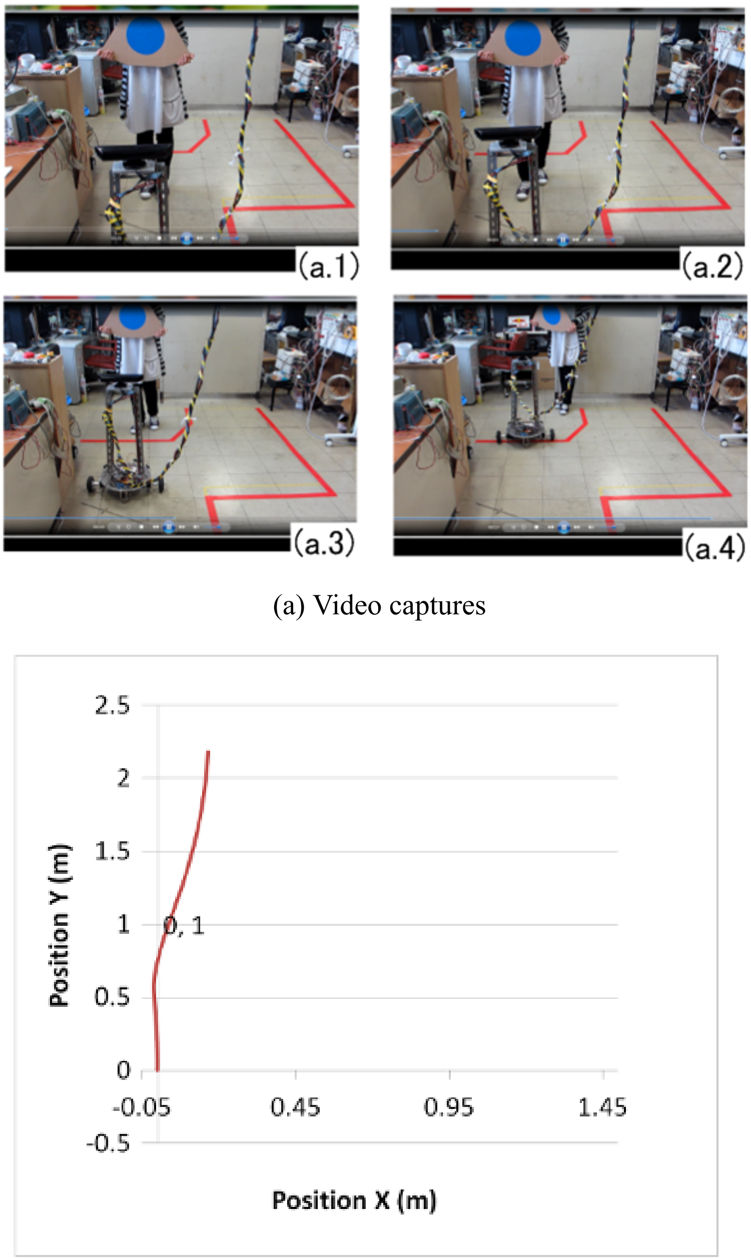
5.5.2 Experimental Results

The experiments were conducted according to the designed scenarios. The results of those experiments are elaborated below

1. Obstacle free environment for moving target following system.

   The obstacle free environment scenario is used to test whether the system is good enough for target following system. The courses are made similar to obstacle existence experiment. The robot intercepted and followed the moving object successfully in all the obstacle free environment courses as shown in figure 5.6a, 5.7a, 5.8a, and 5.9a for video captures, and figure 5.6b, 5.7b, 5.8b, and 5.9b for robot motion trajectories.

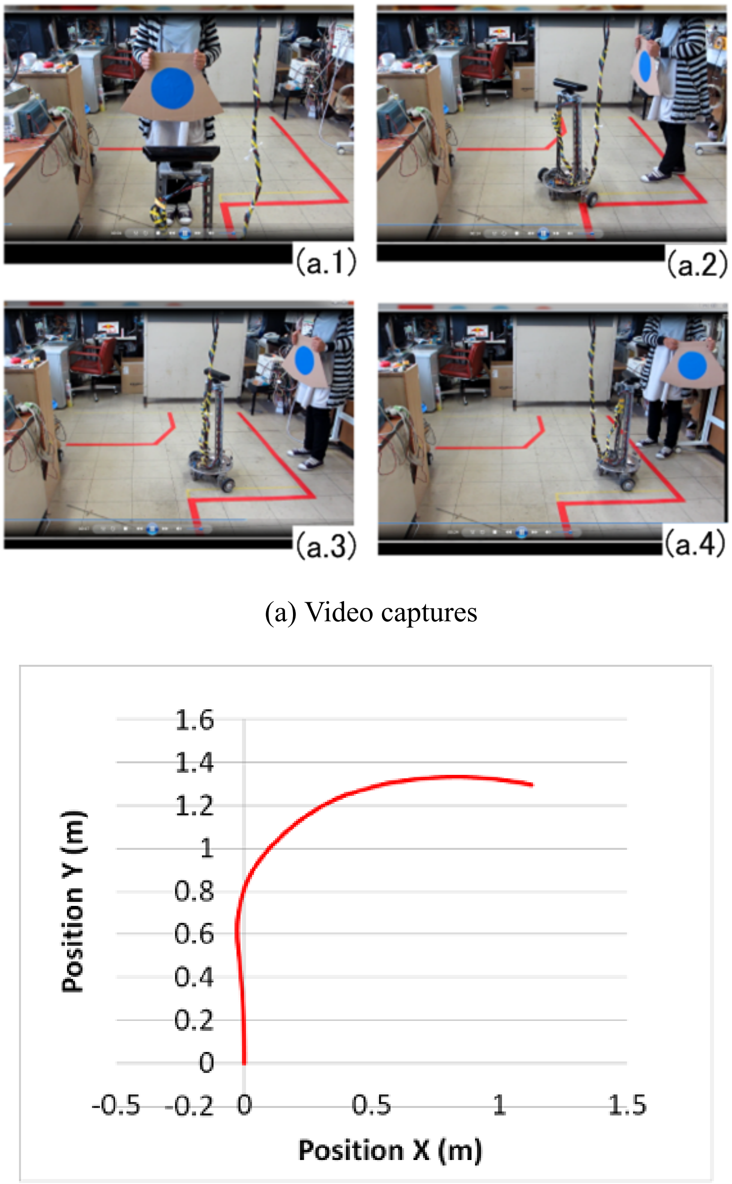
   The wheelchair following system, Figure 5.9a, was only conducted in one course since the effectiveness of the proposed method for obstacle free environment is already confirmed in human-following system for other courses in Figures 5.6a, 5.7a, and 5.8a.



(a) Video captures

(b) Robot motion trajectory

Figure 5.6: Obstacle free environment for straight course human following system



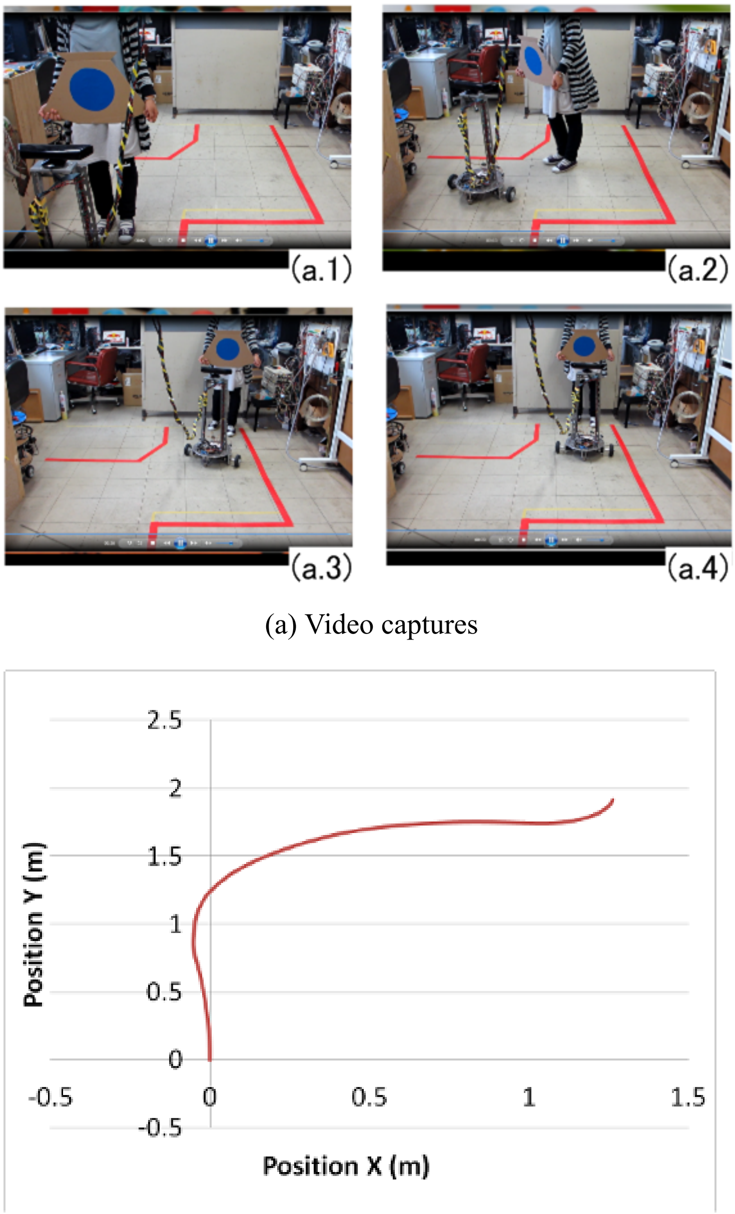
(a) Video captures

(b) Robot motion trajectory

Figure 5.7: Obstacle free environment for L-shaped course human following system

2. Obstacle existence environment.

   The next experiment is by applying the system in obstacle existence environment, where the robot is subjected to the courses bordered by boxes considered as the obstacles. The specification of the courses used are shown in figure 5.10. Figure 5.10b shows the track specification used by the robot to intercept and follow the human target in a crank course bordered by obstacles. Figure 5.11a shows the video captures of the robot following the human target, and avoiding the obstacles



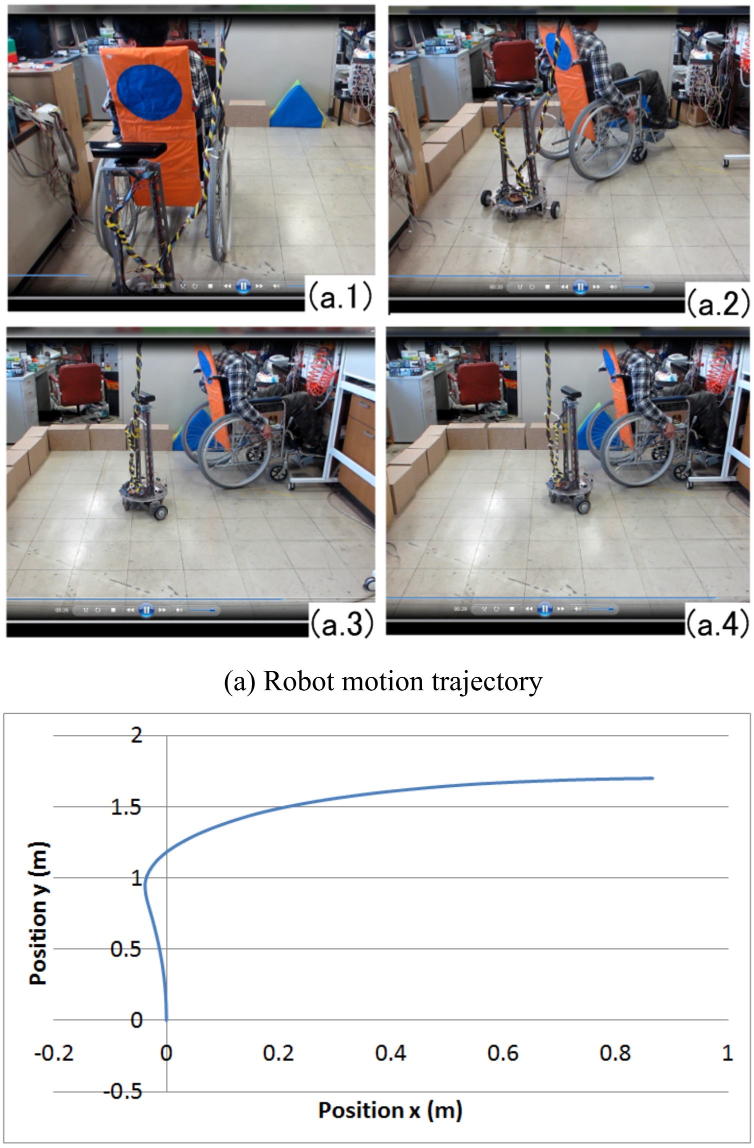
(a) Video captures

(b) Robot motion trajectory

Figure 5.8: Obstacle free environment for crank course human following system

(border boxes). The obstacle avoidance trajectory is generated by applying the repulsion virtual force in eq. (3.9).

To validate of the proposed method, we used a further four human targets to repeat the experiment as shown in Figure 5.11a. Figure 5.11b shows the results of robot motion trajectory for the five human targets, where H1-H5 denotes the human target number. The robot follows all the human targets continuously with the same motion trajectory. Figure 5.12a - 5.12d shows the distance data from the proximity sensors installed on the robot, where d0 - d3 are the distance data from proximity



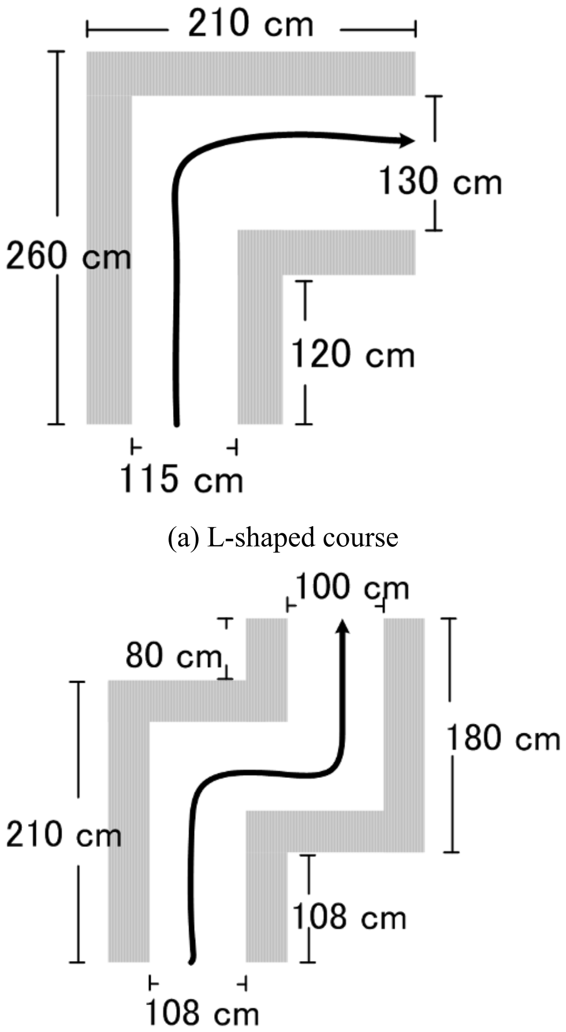
(a) Robot motion trajectory

(b) Robot motion trajectory for L-shaped course

Figure 5.9: Obstacle free environment for wheelchair following system

sensors 0 - 3, respectively. When the distance from the robot to an obstacle is less than the minimum, the robot avoids the obstacle while still following the human target. In the second scenario, occlusion does not occur in all cases.

In order to prove the effectiveness of the proposed method applied in service robot, we attached the blue circle mark to a wheelchair. Figure. 5.10a shows the L-shaped course and Figure 5.10b shows the crank course for the robot to track and follow a wheelchair and function as a service robot giving support to a human. Figure 5.13a is the video captures for wheelchair following, and figure 5.13b is the robot motion trajectory. After we confirmed that the robot can follow the



(a) L-shaped course

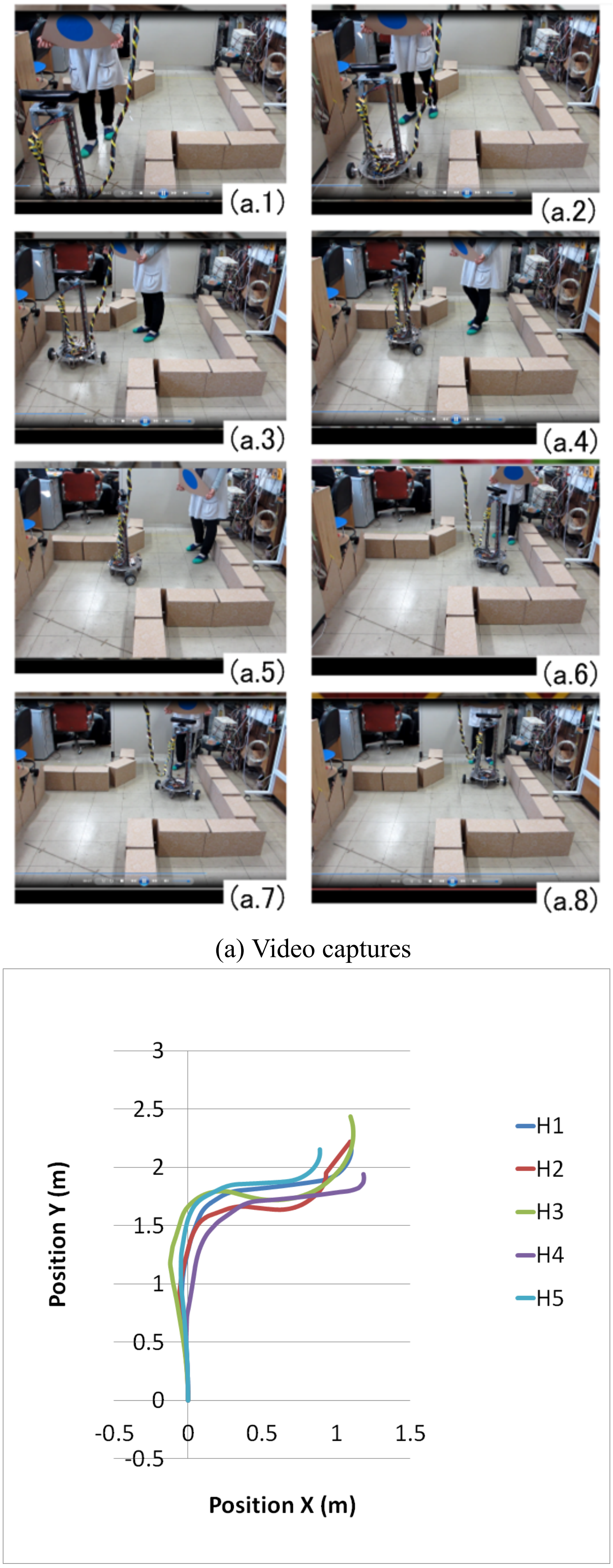
(b) Crank course

Figure 5.10: Obstacle existence course design

wheelchair as it followed a human, the robot was set to follow wheelchair in a crank course and the video captures is shown in figure 5.14a. To give repeatability proof of the proposed method, we asked other four humans to operate the wheelchair in the same course. Figure 5.14b shows the result of robot motion trajectory for five human-targets in wheelchair following system.

3. Occlusion.

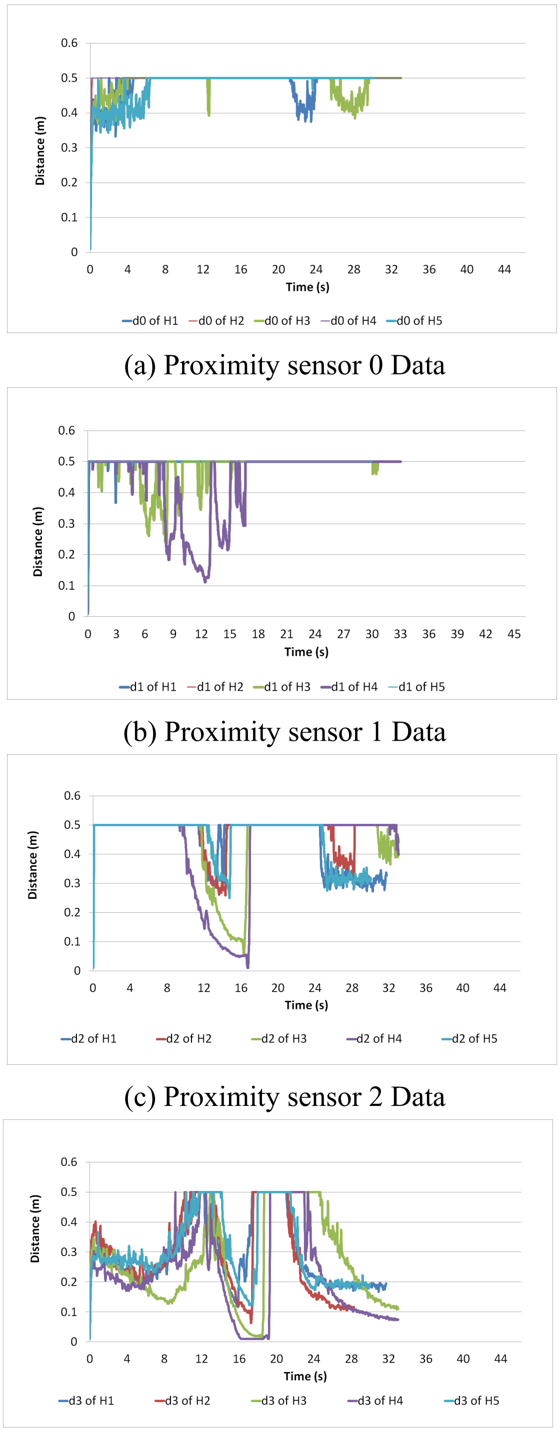
   In the third scenario, as the robot intercepts the mark attached to the human-target, it is subjected a brief occlusion (i.e., the robot loses sight of the blue circle mark). However, since the attractive virtual forces are applied to the blue circle mark, the robot turns back to the direction of the human target and continues to follow.



(a) Video captures

(b) Robot motion trajectory for 5 human targets

Figure 5.11: Human following in crank course bordered by boxes



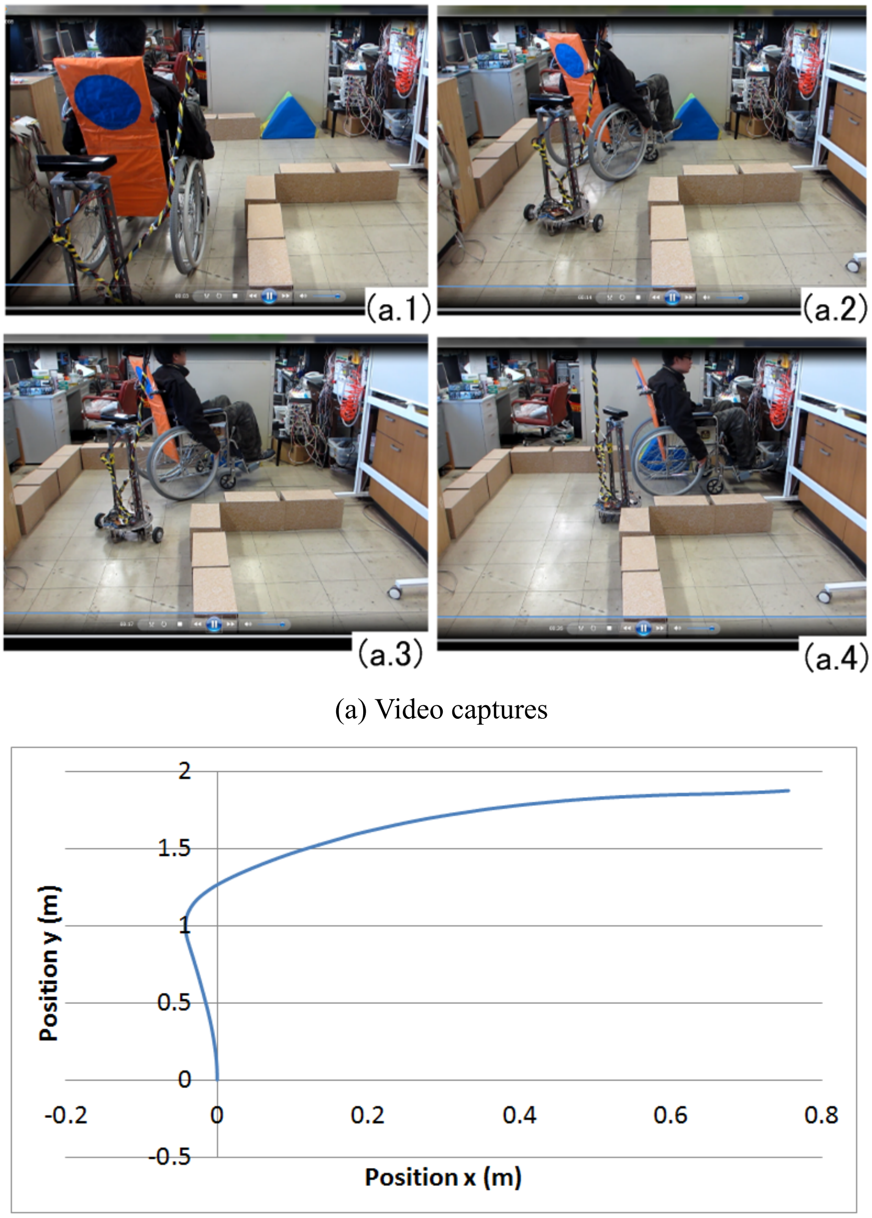
(a) Proximity sensor 0 Data

(b) Proximity sensor 1 Data

(c) Proximity sensor 2 Data

(d) Proximity sensor 3 Data

Figure 5.12: Proximity sensors data for human following system in crank course bordered by obstacles for five human targets



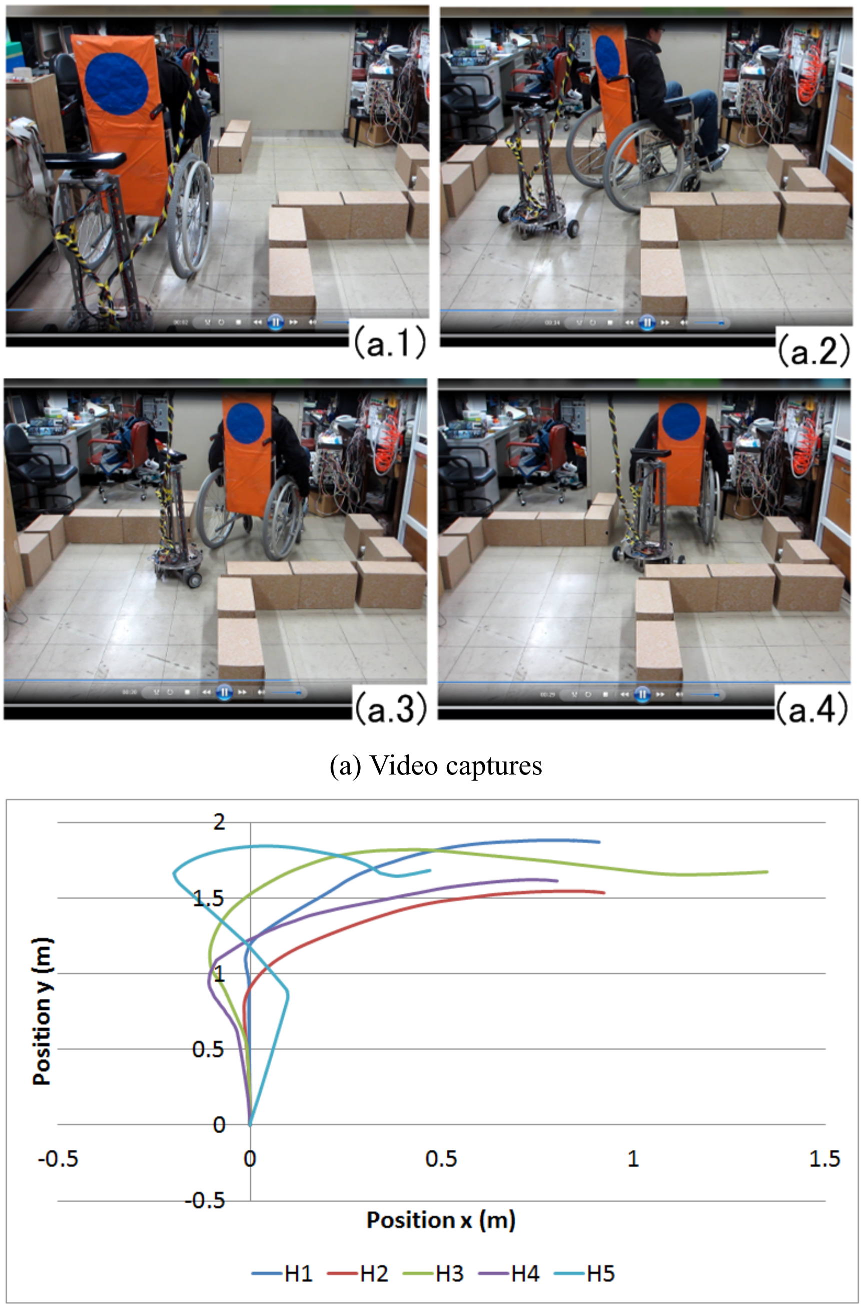
(a) Video captures

(b) Robot motion trajectory

Figure 5.13: Wheelchair following in L-shaped course bordered by boxes

The first occlusion experiment is given by video captures in figure 5.16a where the human target made a sudden turn, making the robot suffers a brief occlusion. Tracking means continuous following and the robot keeps on following and knowing the position of human-target all the times. Since the human-target direction is known to the robot, the robot can maintain the tracking and following although there is a brief occlusion between them.

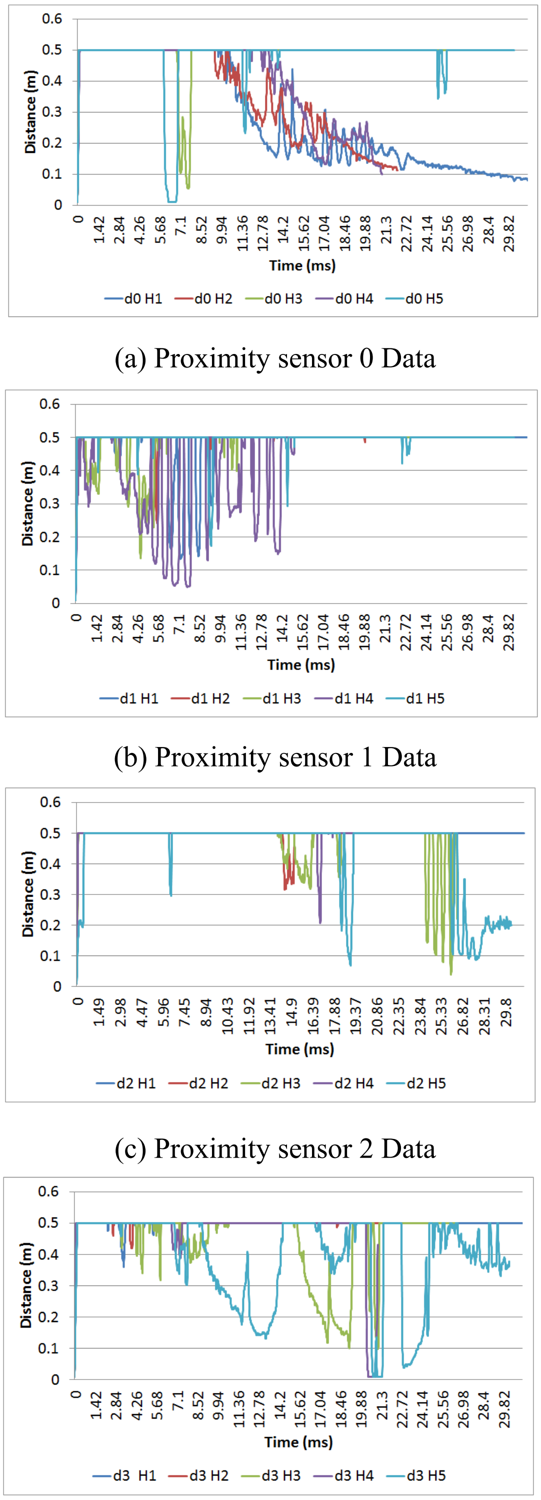
Since the pose of human-target was known by the robot beforehand, i.e. the robot



(a) Video captures

(b) Robot motion trajectory

Figure 5.14: Wheelchair following in crank course bordered by boxes



(a) Proximity sensor 0 Data

(b) Proximity sensor 1 Data

(c) Proximity sensor 2 Data

(d) Proximity sensor 3 Data

Figure 5.15: Proximity sensors data for wheelchair following system in crank course bordered by obstacles for five human targets

"sees" the next pose of the mark after reaching the current pose of the mark, therefore the robot can still maintain the human following system. Figure 5.16b shows the robot motion trajectory of continuous following despite the availability of brief occlusion.

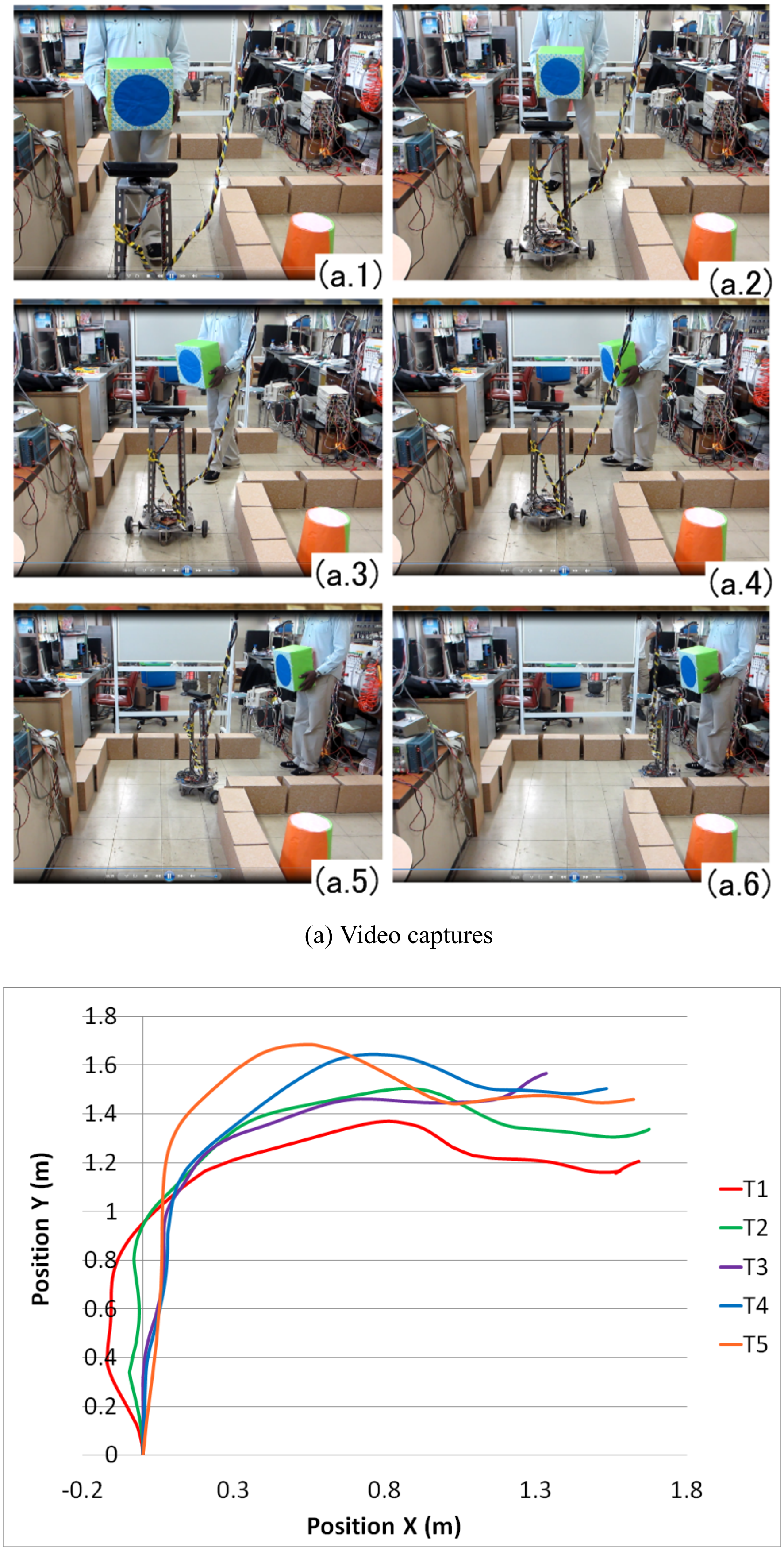
The brief occlusion can also occur due to obstacle avoidance. The obstacle avoidance in the third scenario is different from that in the second scenario. In the second scenario, the robot avoids obstacles without losing sight of the blue circle mark; therefore, it is still able to follow continuously. In the third scenario, the frontal obstacle avoidance causes the robot to change direction and human tracking is interrupted. The virtual attraction force of the target causes the robot to know the prior position of the human target, therefore if the occlusion happens briefly, the robot still can maintain the following system. The video captures of the third scenario in Figure 5.17a show that the robot turns back to the direction of the human target, and this is also shown in the robot motion trajectory in Figure 5.17b.

4. Extended to Swarm Robot.

   The fourth scenario is meant to test the possibility of creating swarm robot using the proposed method. The robot is tested to follow the leader robot in L-shaped course and crank course. Due to the limited availability of the robots, we can only test one robot leader and one robot follower. The robot leader trajectory generation was not applied using the proposed method of this study. Figure 5.18a shows the L-shaped course, and Figure 5.19a shows the crank course. The difficulty of leader-follower system is that the robot leader can make sudden turn causing the robot (follower robot) to suffer a brief occlusion. Since the vision-based control applied to the robot is robust to brief occlusion, the robot was able to make continuous following system for both the L-shaped and crank course, as shown in robot motion trajectory in 5.18b and 5.19b. The proposed method also makes it possible to create the swarm robot as discuss in figure 3.12 of Chapter 5. However, the limitation of space in our test-bed makes it very difficult.

5. Extended to Hallway

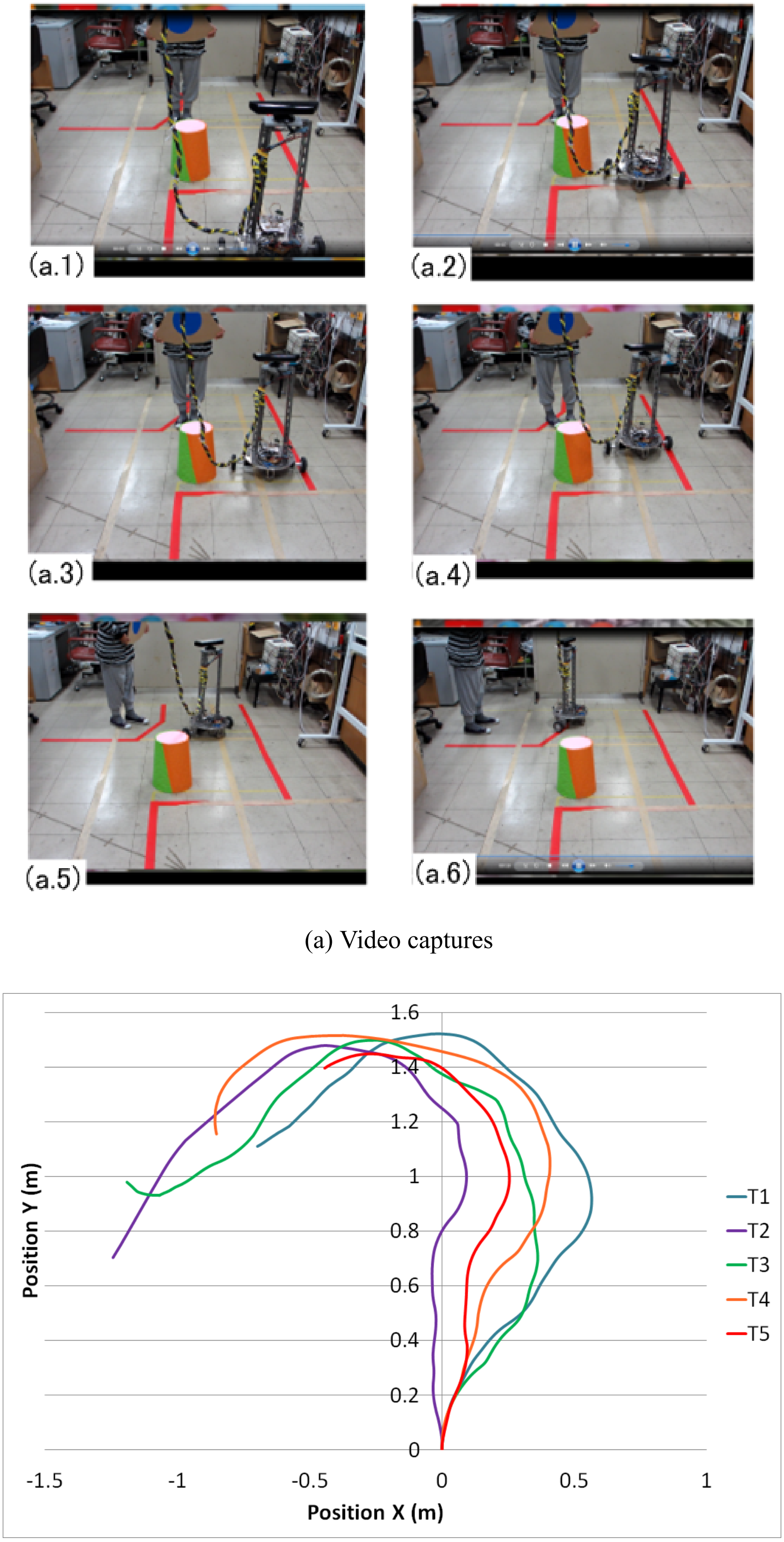
   In order to prove that the proposed system can be extended to outside laboratory testbed, we did experiment in hallway, shown in fig. 5.20. Fig. 5.21 is the video captures that show the robot successfully follows the wheelchairs. However due to the Kinect sensor limitation and sensitiveness to the light, the robot losses the sight of target many times (occlusion occurrences). We conducted the experiment



(a) Video captures

(b) Robot motion trajectory

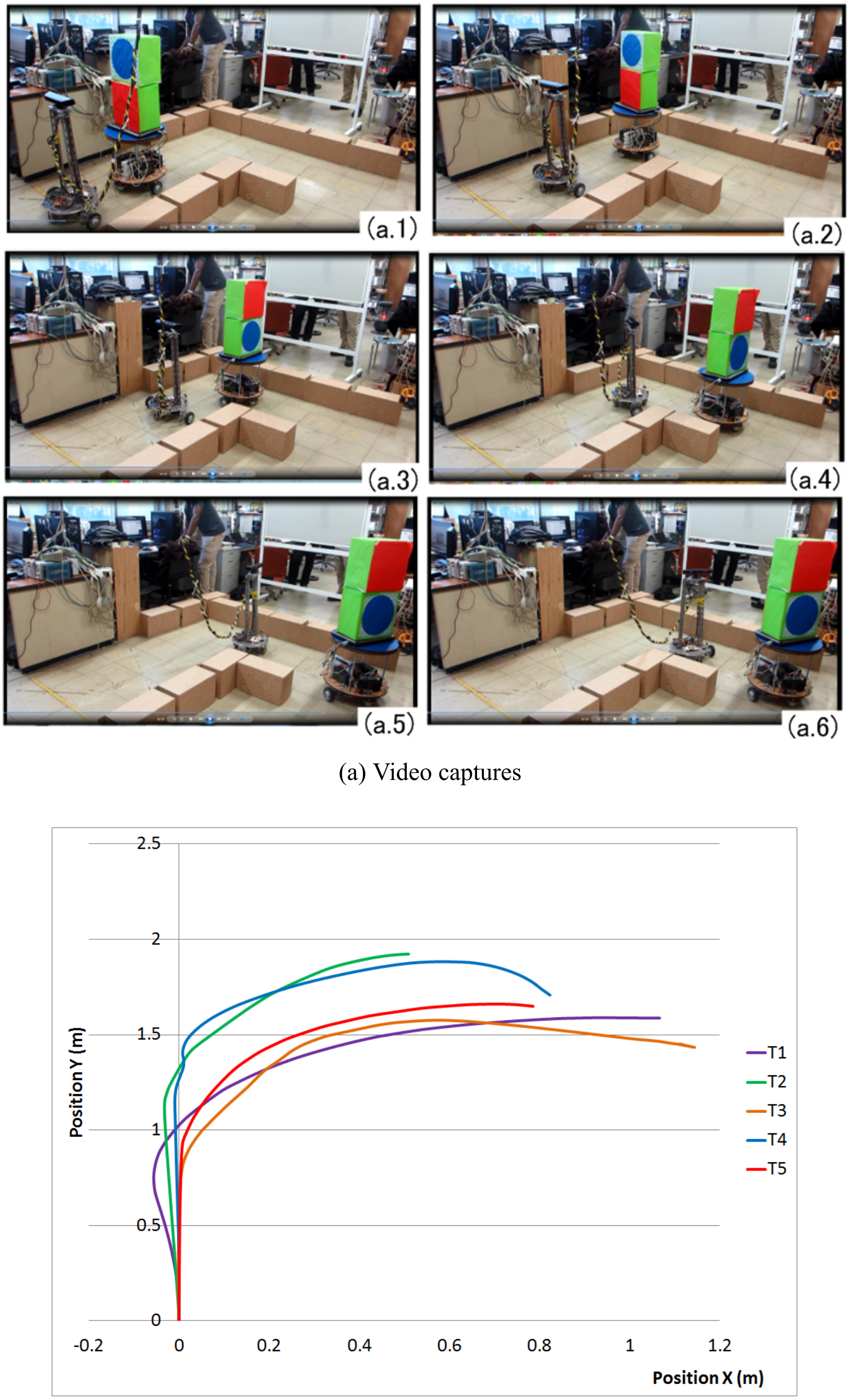
Figure 5.16: Brief occlusion due to human sudden turn



(a) Video captures

(b) Robot motion trajectory

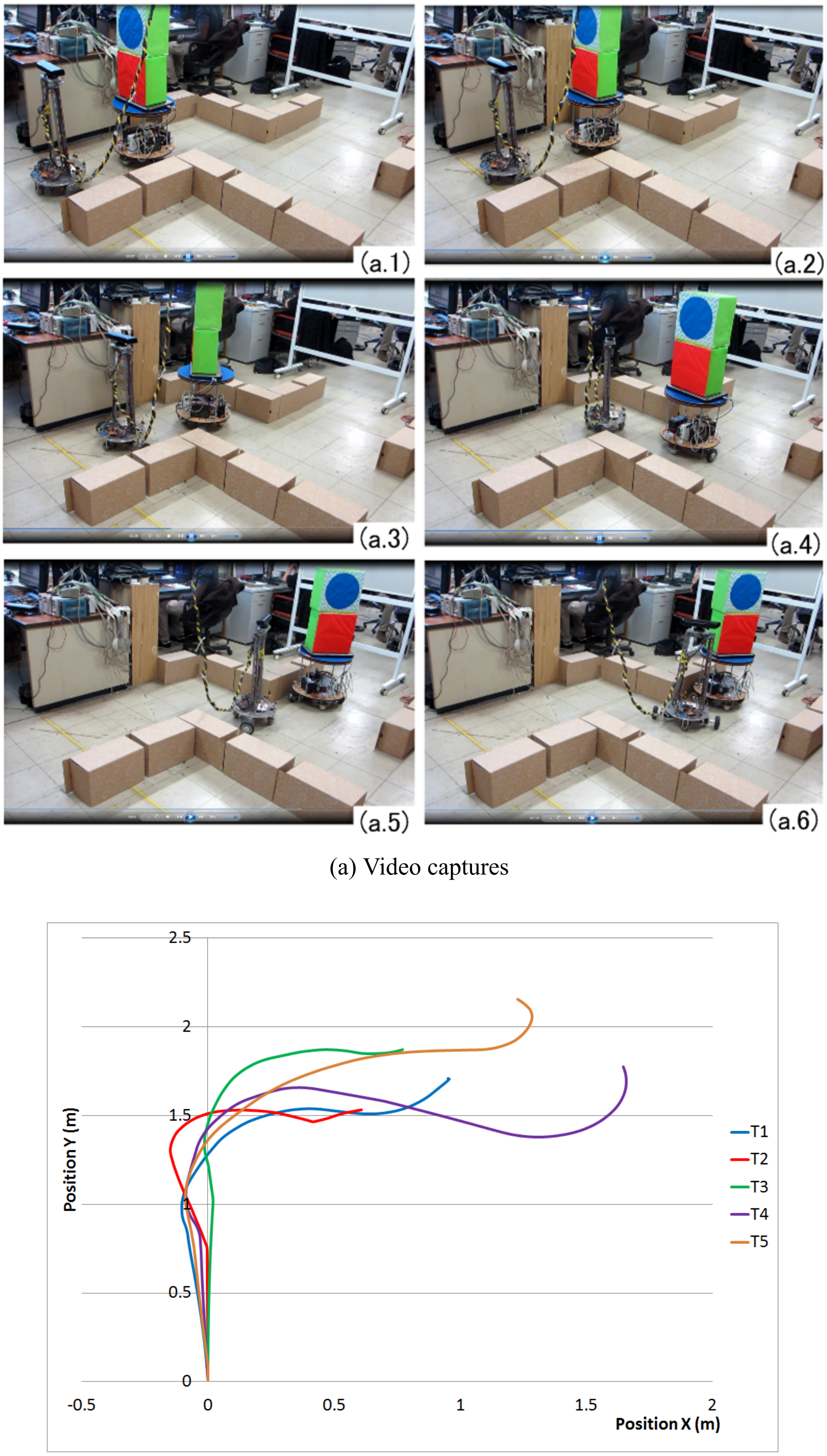
Figure 5.17: Brief occlusion course due to obstacle avoidance



(a) Video captures

(b) Robot motion trajectory

Figure 5.18: Leader robot following in L-shaped course



(a) Video captures

(b) Robot motion trajectory

Figure 5.19: Leader robot following in crank course

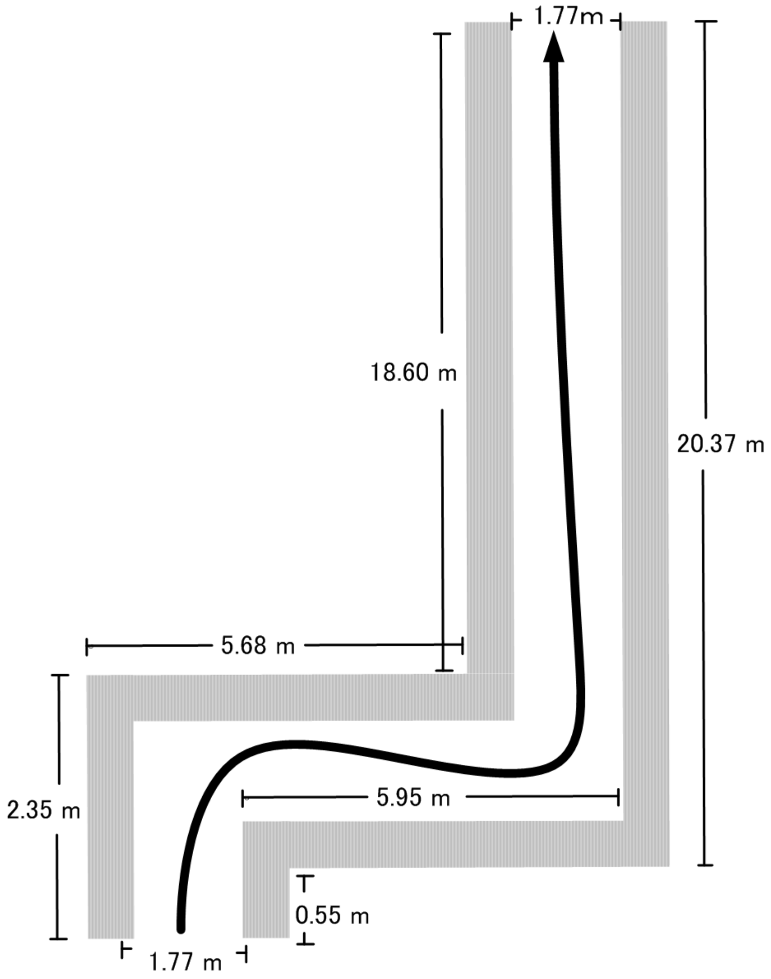


Figure 5.20: Hallway track course

in 3 steps, shown in figure 5.22b, from the shortest track distance to the longest track.

5.6 Conclusion

We applied the vision-based object recognition (image processing and depth mapping) and potential field methods in this study. This study is the continuation of previous chapters; however, here, the target is changed to a moving object (attached to a human target, wheelchair, or leader robot). Experiments were conducted by placing the robot in several environmental settings. All the experimental results show that the proposed method is effective for visual sensor based control of a mobile robot.

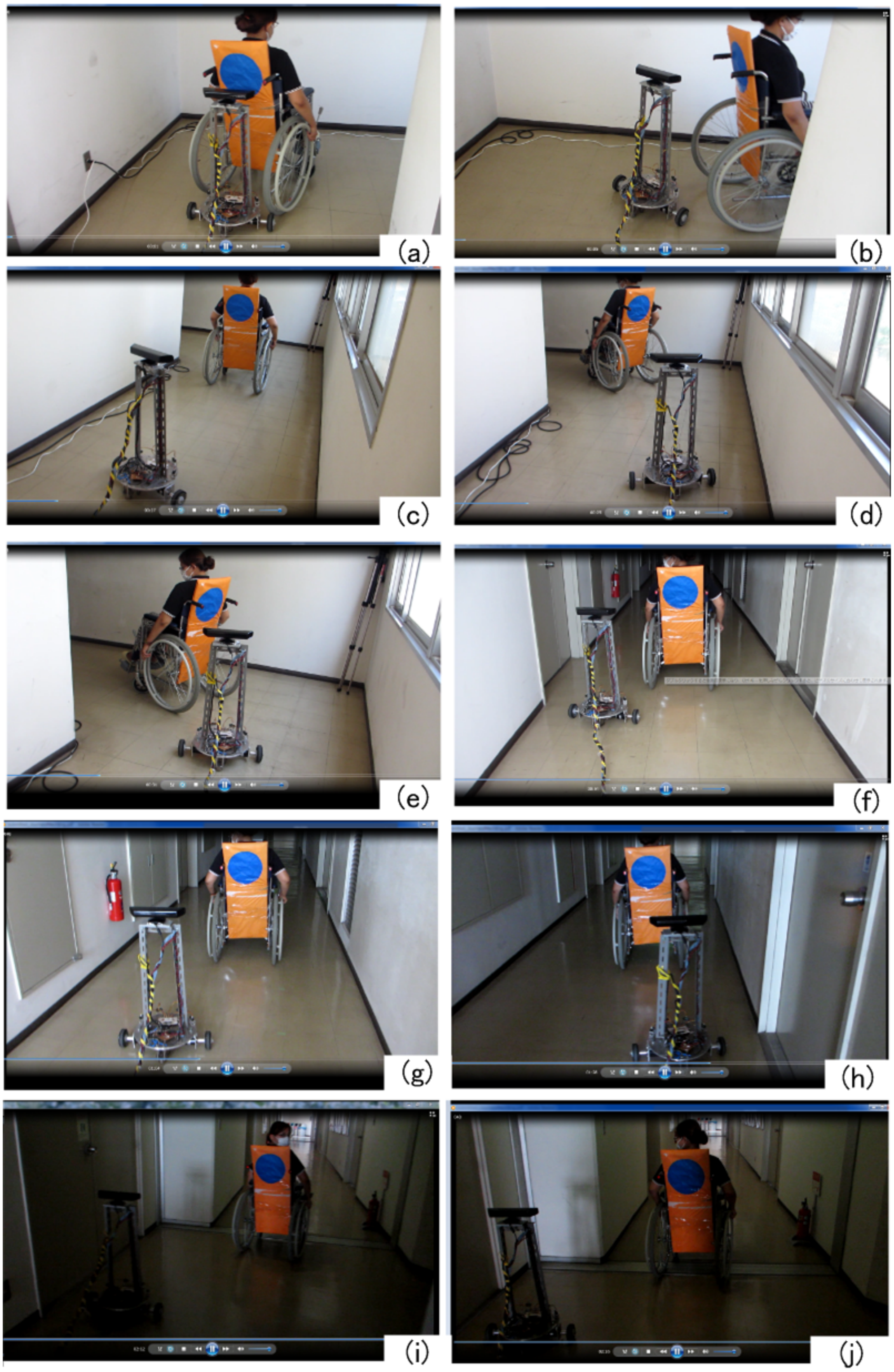
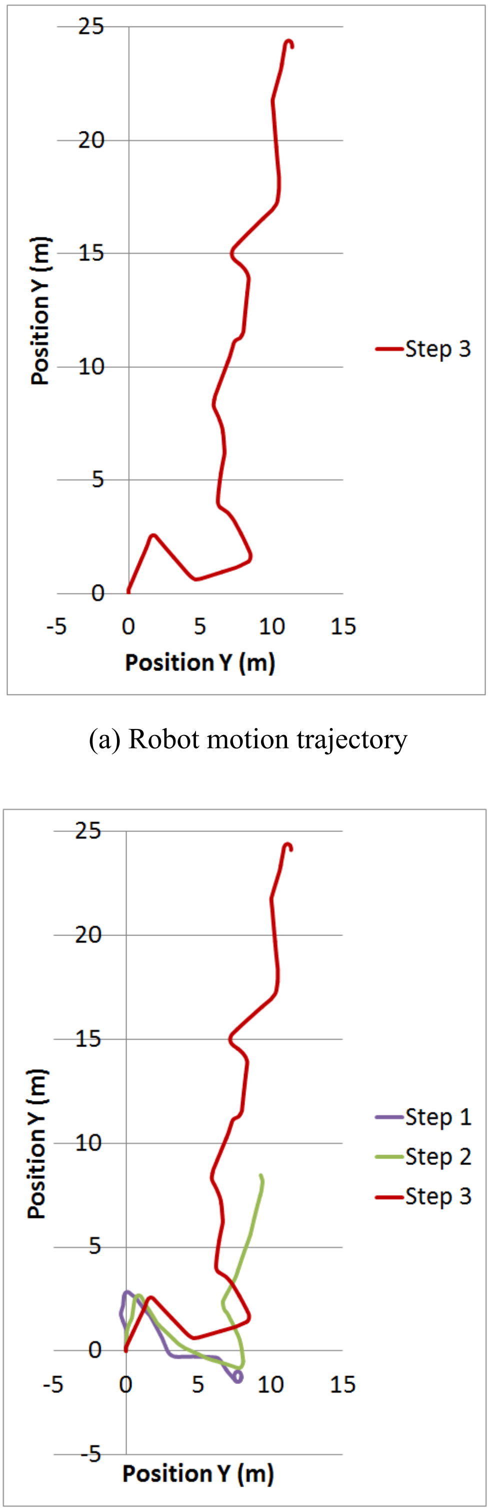


Figure 5.21: Video captures of wheelchair following in hallway



(a) Robot motion trajectory

(b) Robot motion trajectory for 3 attempts

Figure 5.22: Robot motion trajectory of wheelchair following in hallway

Chapter 6

Extended Work

6.1 Introduction

This chapter extends the work presented in Chapter 5. The robot specifications were those discussed in Chapter 2. This work reconstructs the research published in [13][62][117] and [118] by modifying the proposed method, thus demonstrating its adaptability to many applications.

Although color detection and human detection have been extensively reported, most of the studies have employed expensive sensors and complicated algorithms. This chapter discusses whether human and object following can be achieved by combining inexpensive sensors and a simple algorithm. The test robot is shown in figure 6.1.

This chapter is related with the work presented in [112] [113].

6.2 Blue Detection

6.2.1 Vision-based System Designed

In this study, object tracking and following were achieved by RGB color detection without specific shape detection, as discussed in Chapter 6. The proposed system design was compared with that used in [62]. The designed vision-based system can be summarized as follows:

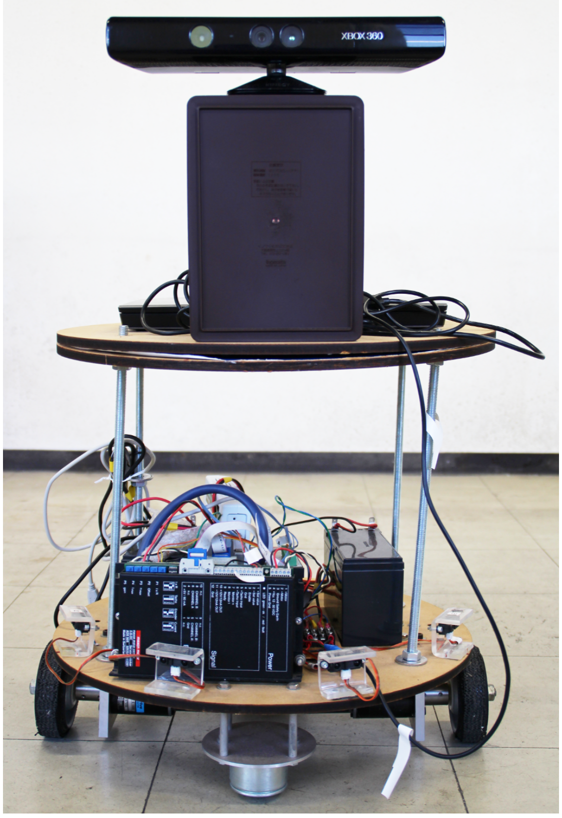


Figure 6.1: Two-wheeded mobile robot

- The target is a blue object attached to a human target or wheelchair. This target will be tracked and followed by the robot.

- The target attached to the human or wheelchair is identified by processing the RGB images. The distance from the robot to the blue object is calculated by depth mapping.

  – The blue color is filtered from other background colors by a Euclidean filter, which fills the entire background (other than the target color) with black. Figure 6.2 shows the result of Euclidean filtering.

  – The blue color is maintained at the center of the camera at all times.

  – The distance between the blue color and the camera is calculated by depth mapping.

  – The blue detection is analyzed by the algorithm discussed in Chapter 6.

- The desired (reference) trajectory is set by applying an attractive virtual force to the target.

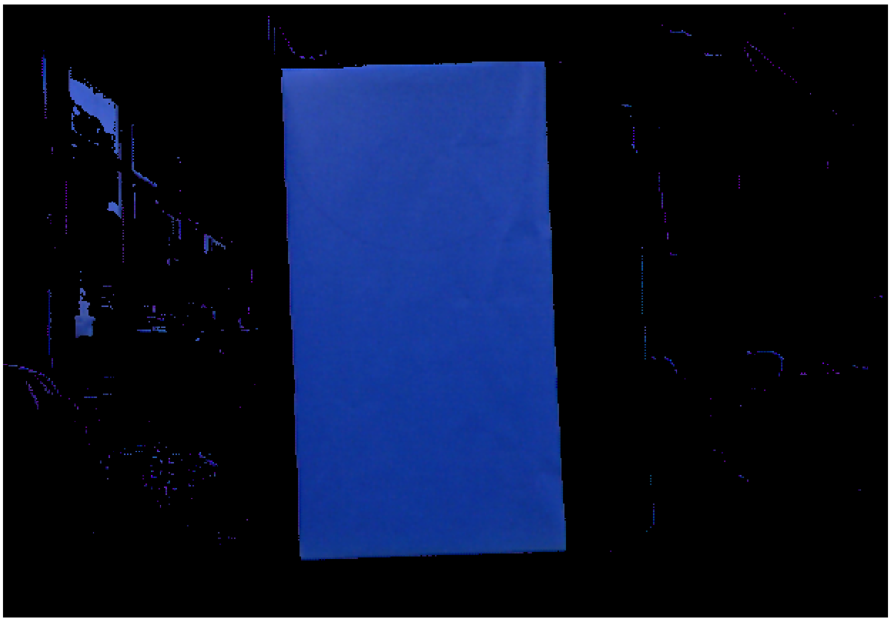


Figure 6.2: The blue mark detection using Euclidean filter

- To improve the effectiveness of the system and to differentiate the blue mark from other blue regions in the experimental environment, a second attractive force is computed from the IR proximity sensor data. This force maintains the robot close to the followed human or wheelchair.
- The desired trajectory of the robot is guided by a PI controller.

The following assumptions were made in this experiment:

- The camera was a Kinect sensor mounted on top of the robot, and its position was fixed.
- The poses of the camera and robot were identical.
- The pose of the image was modeled by the pinhole lens approximation.

6.2.2 Experiment

This experiment assessed the effectiveness of the proposed method. The robot was assigned to track and follow the moving blue object attached to the human or wheelchair. The blue mark was sufficiently large to be easily recognized by the robot. To prevent the robot from following other blue marks, we combined the attractive force toward the mark with the virtual attractive force toward the human/wheelchair attached with the blue mark. The virtual attractive force on the human/wheelchair was based on the data from the IR proximity sensors.

The blue-mark-following experiment was conducted under the following conditions:

1. Blue mark attached to a human
   - The robot followed a blue mark attached to a human inside the laboratory. The human moved freely and was followed by the robot.
   - The robot followed a blue mark attached to a human in a hallway.
2. Blue mark attached to a wheelchair
   - The robot followed a blue mark attached to a wheelchair inside the laboratory. The wheelchair was operated freely and was followed by the robot.
   - The robot followed a blue mark attached to a wheelchair in a hallway.

The experiments were conducted according to the designed scenarios. Figure 6.3 and 6.4 show the video captures of the robot following a blue mark attached to human who moved freely inside the laboratory and in the hallway respectively.

Figure 6.5 and 6.6 show the video captures of the robot following a blue mark attached to a wheelchair that operated by a human inside the laboratory and in the hallway, respectively. The experimental results show that the proposed method is effective for all of the scenarios. However, the designed system is not including obstacle avoidance and not robust to occlusion.

6.3 Human Detection

6.3.1 Vision-based System Designed

Human detection and following is typically achieved by a stereo-camera system and laser range finders. Satake et al. [117] [118] detected and tracked a walking person by a template matching method. Munoz-Salinas et al. [13] used a single camera for person-tracking and combined color detection with the position information.

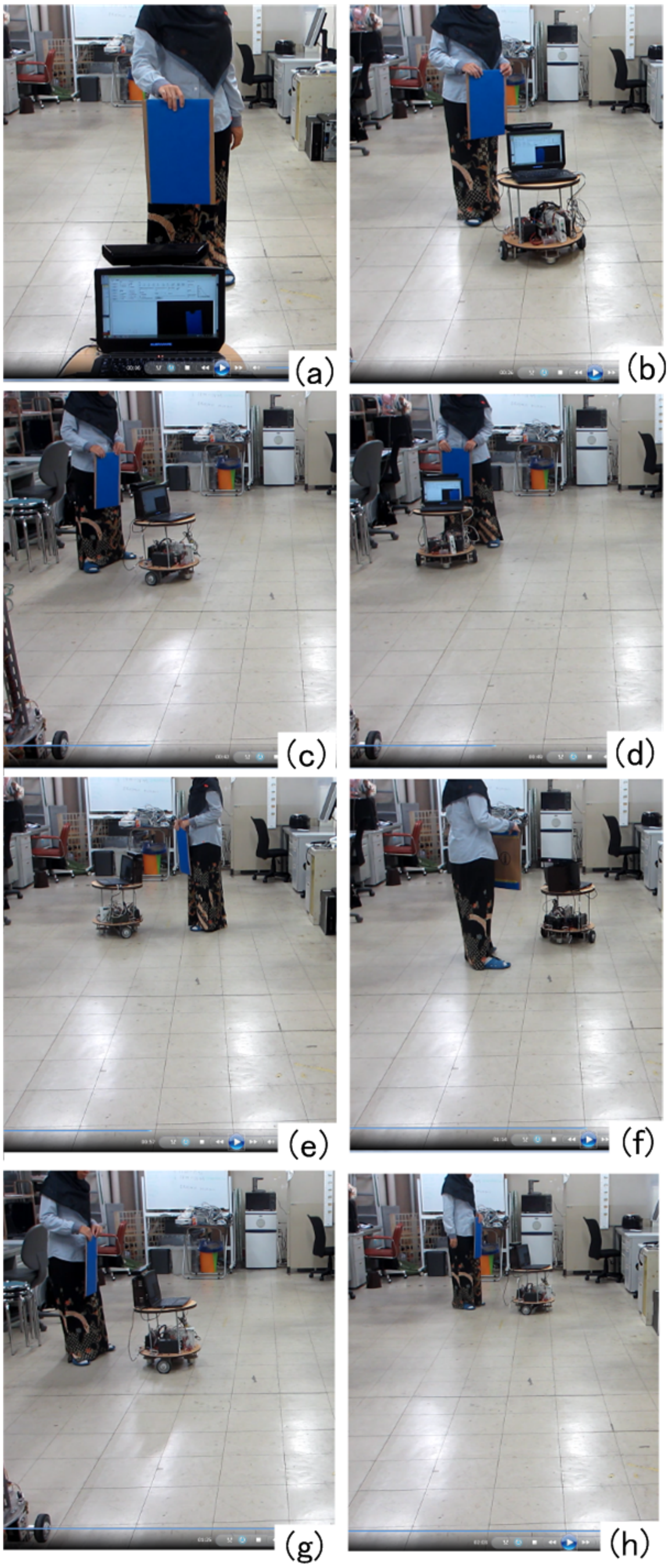


Figure 6.3: Video captures of a blue mark following attached to a human inside the laboratory

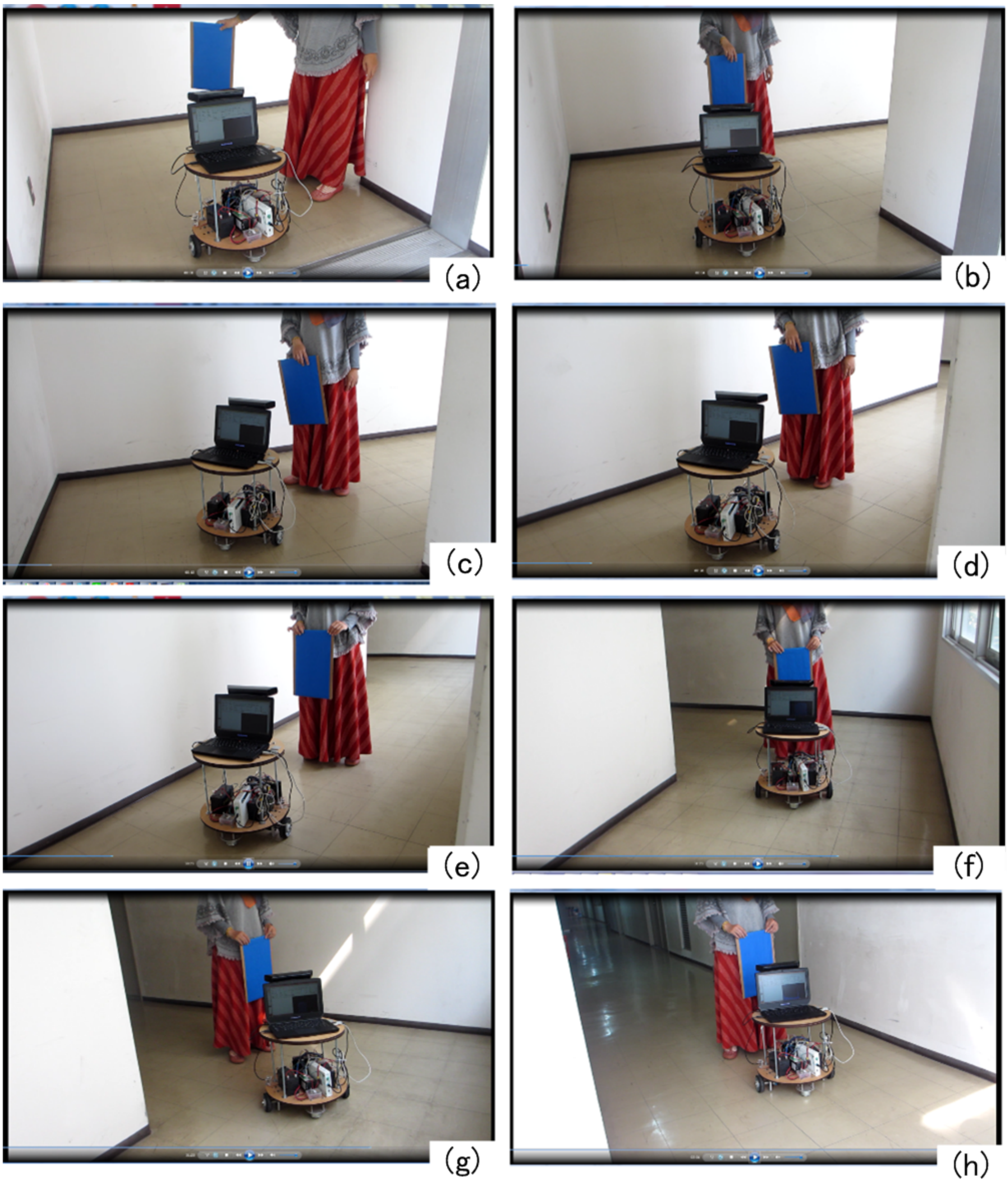


Figure 6.4: Video captures of a blue mark following in a hallway

The human skeleton was detected by the Kinect sensor, and the effectiveness of following the target was enhanced by applying a virtual attractive force to the human target. For this purpose, we used the data from the IR proximity sensor. Skeleton detection by the proposed system is shown in figure 6.7.

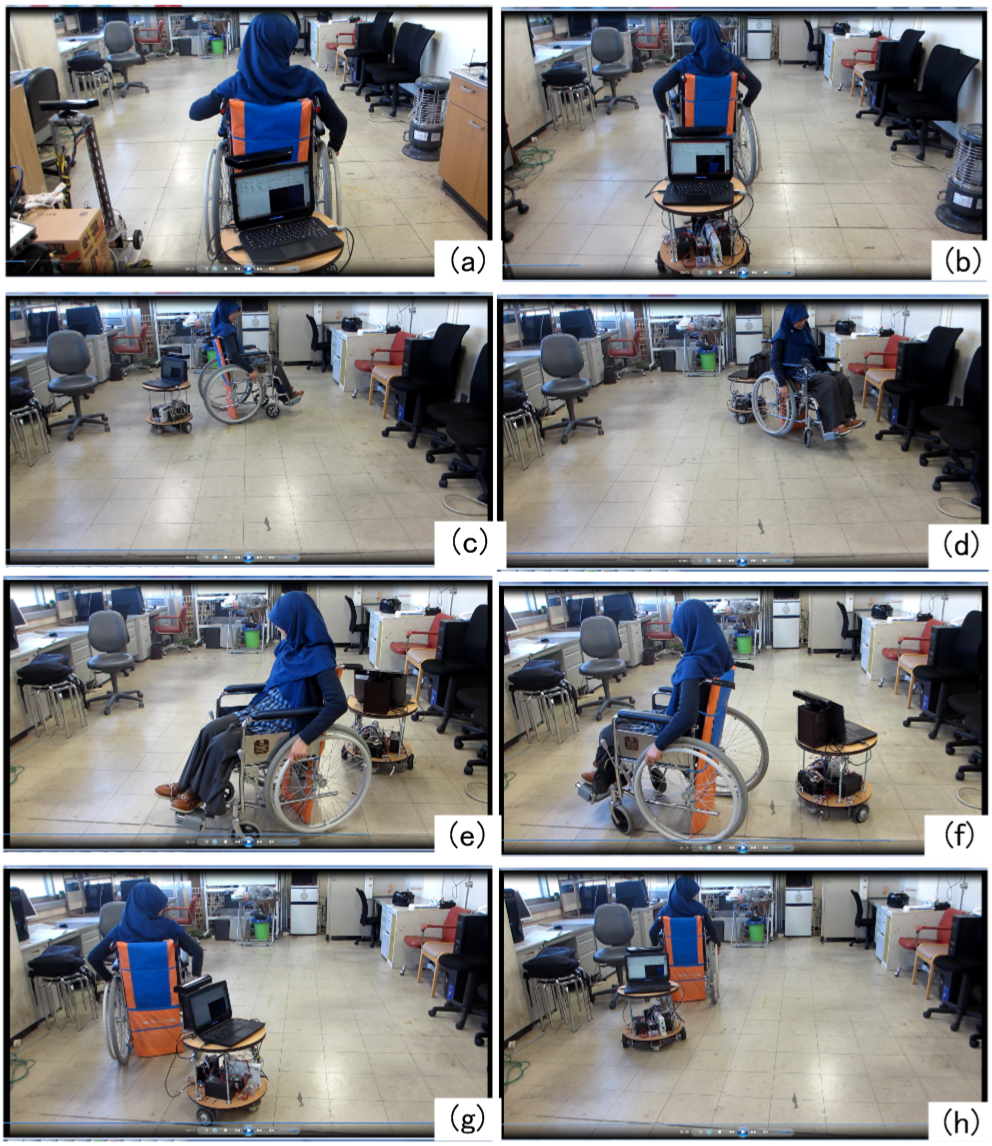


Figure 6.5: The Robot is following a blue mark attached to wheelchair that moves freely inside laboratory

6.3.2 Experiment

Human detection was conducted in two experimental settings: the laboratory and a hallway. In both settings, the human moved freely. Figure 6.8 and 6.9 show the video captures of human tracking and following in the laboratory and hallway, respectively. The proposed method proved effective in both settings. However, tests of obstacle avoidance and robustness to occlusion were excluded.

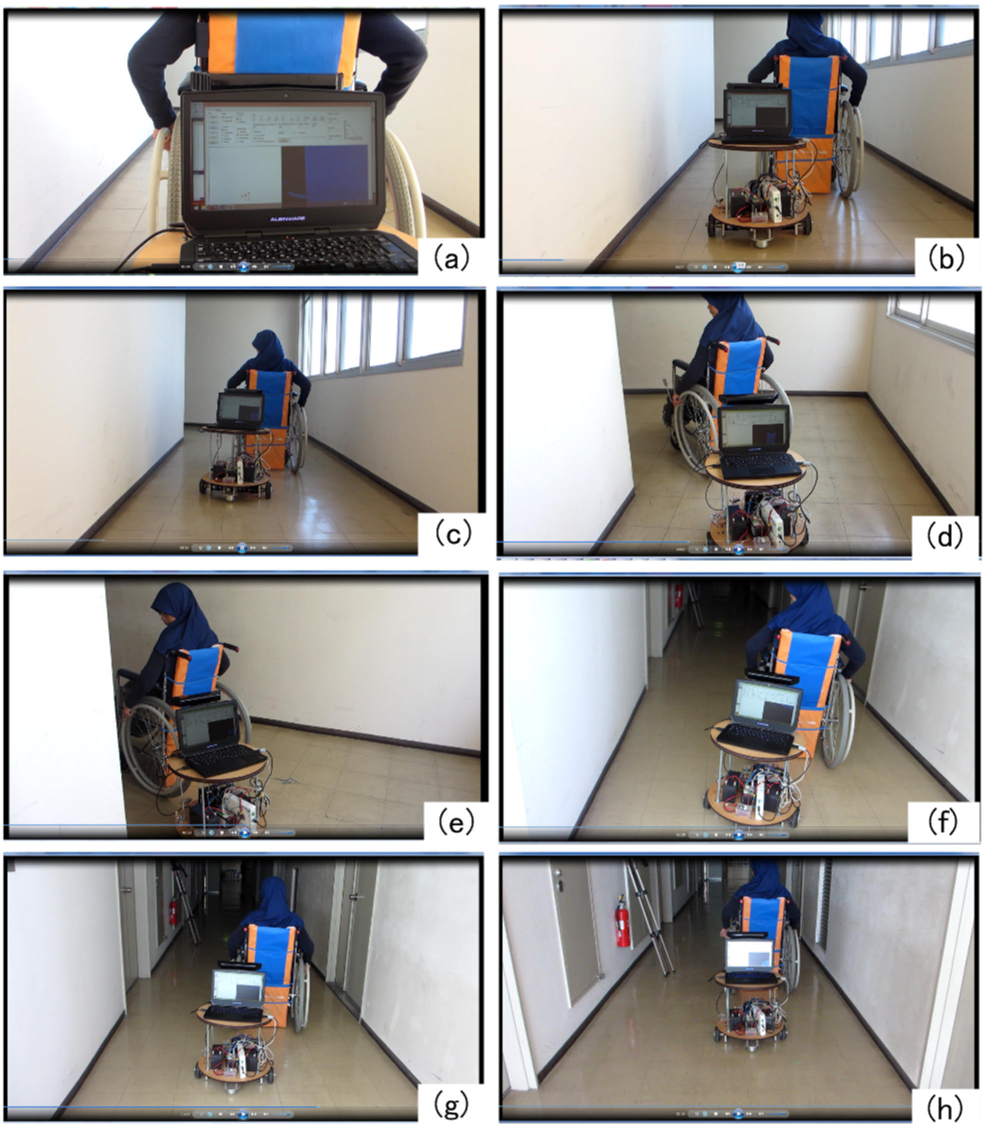


Figure 6.6: The robot is following a blue mark attached to wheelchair moving in a hallway

6.4 Conclusion

This chapter extended the present method to a wider range of applications. The controller was unchanged from previous chapters. The data from the IR proximity sensor provided the attractive force toward the blue mark and the human in the blue-mark-detection and human-detection tests, respectively. The experimental system was designed and the results show the effectiveness and flexibility of the proposed method in diverse applications of the service mobile robot. However, obstacle avoidance and robustness to occlusion were not considered in this chapter.

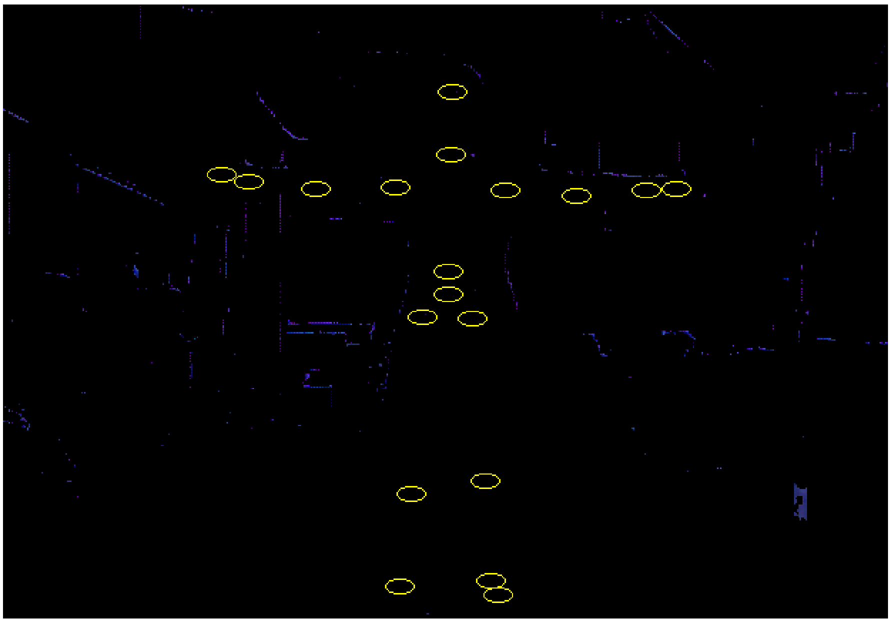


Figure 6.7: Skeleton detection by a Kinect sensor

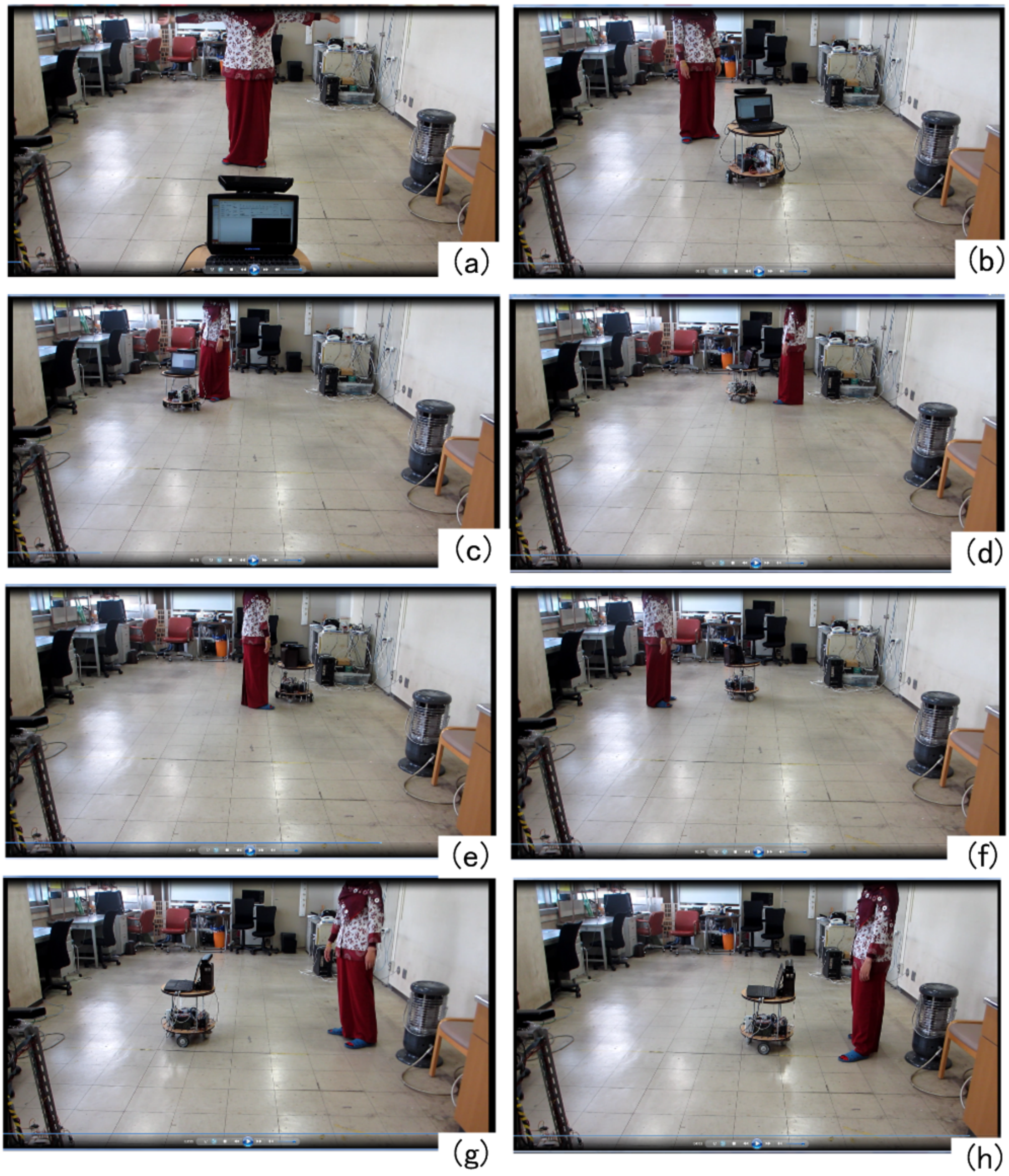


Figure 6.8: Video captures of a robot following a moving freely human

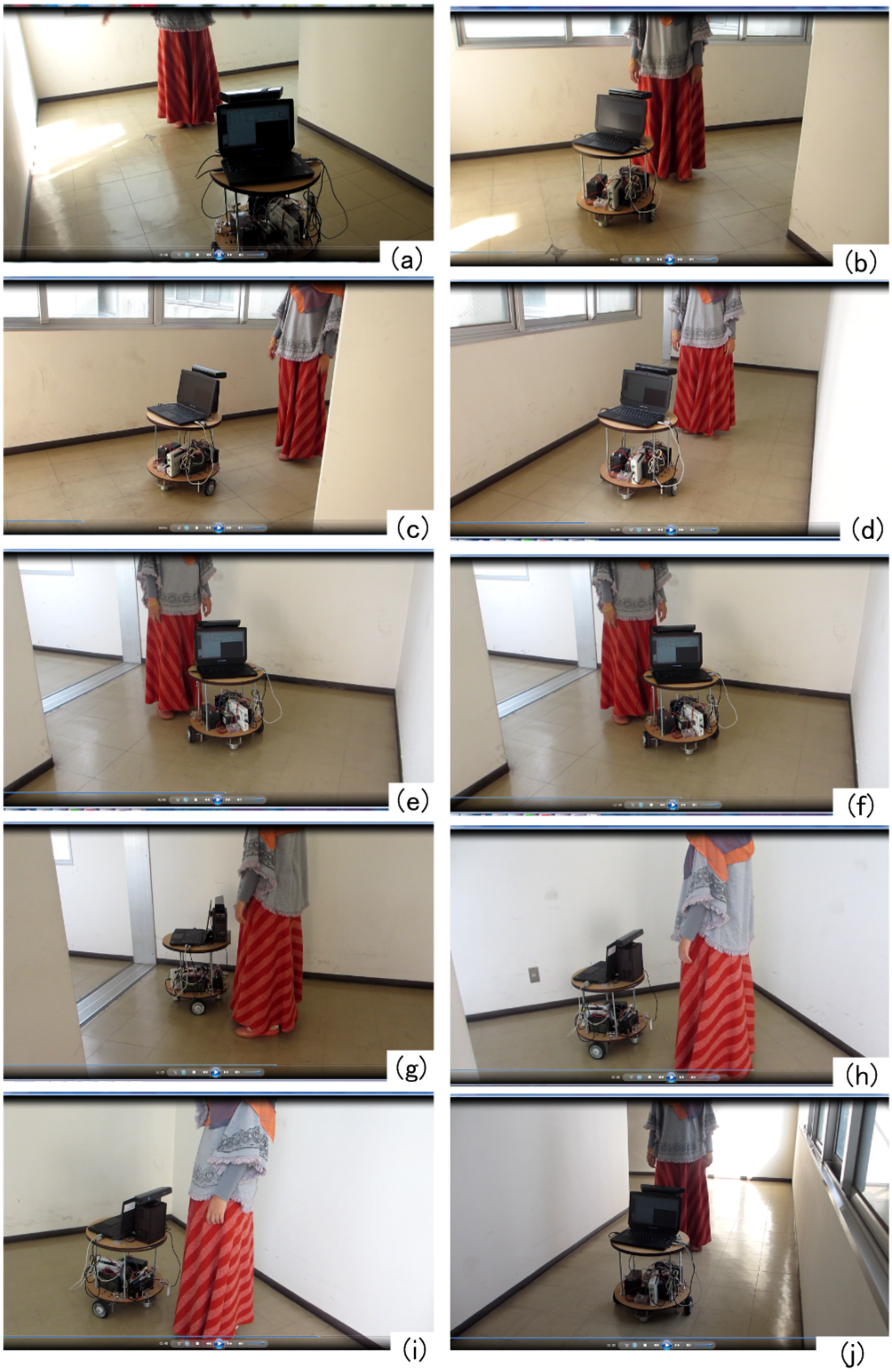


Figure 6.9: Video captures of a robot following human in a hallway

Chapter 7

Summary and Future Works

7.1 Summary

This thesis proposes a control design for collision avoidance and object-following control of mobile robots in human living environments. First we assembled a widely used mobile robot; namely, a two-wheeled differential-drive mobile robot balanced by two castor wheels. The vision sensor was a Kinect sensor mounted on top of the robot, and four IR-proximity sensors were installed for collision avoidance.

The designed reference controller was based on the potential field method, in which obstacles exert repulsive forces on the mobile robot. In Chapter 3, the goal setting was pre-defined by assuming that the targets were indicated by the position sensor. We also designed a well-known PI controller based on the kinematics and dynamics models derived in Chapter 2. An experimental system was constructed, and the effectiveness of the proposed system was confirmed.

Chapter 3 presents the collision avoidance scheme achieved by the model reference control based on the potential field path planner. The designed controllers installed in the mobile service robot are presented in the subsequent chapters. A system integration method, by which the robot moves in a dynamic environment, is also presented. The effectiveness of the proposed system is demonstrated by discussing the results of simulations and real experiments. This chapter also discussed the possibility of creating a swarm robot system by the proposed method by presenting the collision avoidance control for swarm robots moving in a rehabilitation environment of static and moving obstacles. Motion in the dynamic environment was enabled by a system integration of installing several sensors. Specifically, the proposed method combines the information obtained by several proximity sensors, an image sensor, and localization sensors (RFID system). The swarm robot team adopted a leader–follower formation, in which the leader robot followed the rehabilitee, while the other robots followed the leader robot. The robot controllers were the reference and PI controller. The reference controller generates the robot motion trajectory by referring to sensor information in real time, and the PI controller directs the robots along that trajectory. The effectiveness of the proposed design was verified in several simulation studies, with static and dynamic obstacles placed in a human living environment.

Chapter 4 extended the method in Chapter 3 to a human-operated four-wheeled mobile robot in a semi-autonomous setup. Using semi-autonomous robots increases the cost efficiency in some environments, where fully autonomous robots using the currently available technology are impractical. Human-operated robotic systems are widely studied for their suitability in many situations. Human-operated robots supplement human mechanical power, providing precise and smooth operations in difficult physical tasks, and when executing missions in remote or hazardous environments. Chapter 4 presented a novel collision avoidance for four-wheeled human-operated mobile robots using inexpensive infrared distance sensors. By imposing a non-holonomic constraint on the mobile robot, the proposed method provides practical collision avoidance control. The effectiveness of the proposed approach was experimentally verified by establishing that entirely unskilled operators could maneuver the robot to its destination without collisions.

In Chapter 5, we extended our study to vision-based control by maximally employing the Kinect sensor mounted on top of the robot. Semi- or fully autonomous service robots that assist humans in daily life must be able to continuously track humans. This tracking was enabled by installing a visual sensor. The potential field method was combined with visual sensing to create a human-following robot; that is, a robot that tracks and follows an object attached to the human. In this study, the attached object was a blue circular mark. The position of the blue mark and the depth mapping for distance calculation were detected by RGB image detection. As in previous studies, the algorithm implemented the potential field method for path planning. In this chapter, the method of setting the target position in Chapter 3 was modified for object detection, tracking, and following. The effectiveness of the proposed method was validated in human-following, wheelchair-following, and leader robot-following systems, and (as an extension) in a hallway outside of the laboratory testbed. During the experiments, the robot was placed in different environmental settings. The experimental results showed the effectiveness of the proposed method for visual-sensor- based control of a mobile robot.

In Chapter 6, we investigated whether a combination of two inexpensive sensors and a single algorithm could achieve human and object following. The object tracking and following were achieved by RGB color detection as presented in Chapter 6, without specific shape detection. We compared the proposed system design with the work undertaken in [60]. Many investigations on human detection and following have employed stereo-camera systems and laser range finders. To track a walking person, Satake et al. [117] [118] developed a template matching method, whereas Munoz-Salinas et al. [13] used a single camera and combined color detection with position information. In this chapter, we detected the human skeleton by a Kinect sensor. To ensure that the system followed the intended target, we applied a virtual attractive force toward the human target using the IR proximity sensor data. The effectiveness of the proposed method was confirmed in laboratory and hallway experiments. The experimental system was designed, and the proposed method proved effective and suitable for many applications of the service mobile robot. However, this chapter ignored obstacle avoidance and robustness to occlusion.

7.2 Future Works

The designed service mobile robot was based on the potential field method and visual servoing. The effectiveness of the proposed method was proven in several experimental scenarios, including swarm robots and environments outside the laboratory testbed.

The system proved applicable to swarm robots, but the experiment was limited to one leader and one follower robot. Therefore, the scenario investigated in Chapter 5 should be extended such that the leader robot follows a human and the follower robot follows the leader robot.

The robot performance was poorer in the hallway than in the laboratory, because the Kinect sensor was sensitive to the uneven lighting in the hallway. In the laboratory, the light was evenly distributed, so the performance remained strong. Therefore, this system could be improved by applying position prediction control [58].

Appendix A

Parameter Design for the Reference Controller

Referring to [116] the parameters for the reference controller were designed by adjusting the size of the virtual external force applied to the robot, which was determined based on the data from four IR proximity sensors installed on the robot. Figure A.1 shows the sensor configuration in the parameter design.

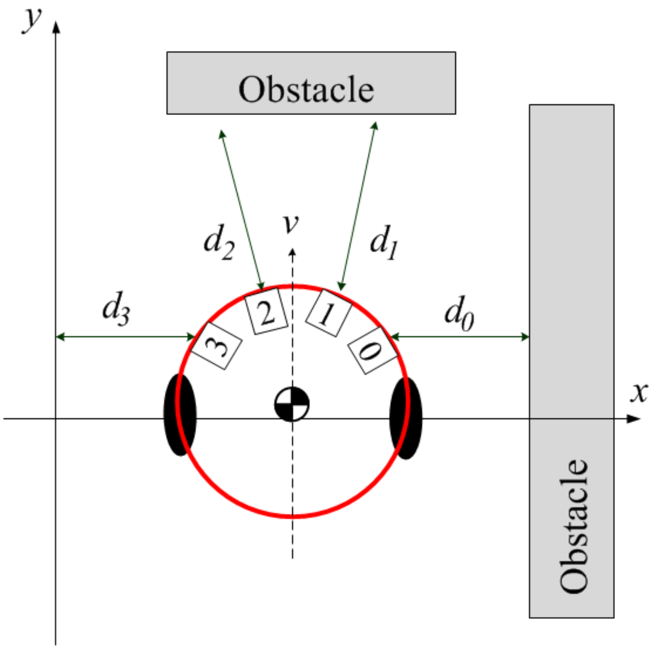


Figure A.1: Sensor configuration for parameter design

A.1 Acceleration Limitation

As a robot approaches an obstacle, its acceleration must be reduced to prevent sudden motions in avoiding the obstacle. In obstacle avoidance control of a wheelchair, smooth motion is required for the comfort of the user [116]. Acceleration limitation can be achieved by adjusting I and M as follows:

$$I = \frac{\left|\tau_u - C_\phi v\right|}{\ddot{\phi}\mathrm{max}} \tag{A.1}$$

$$M = \frac{\left|F_u - C_v v\right|}{\dot{v}\mathrm{max}} \tag{A.2}$$

From the dynamics modeling equation for the reference controller (see Chapter 3), eq. (3.9) can be rearranged as

$$\begin{aligned}
&\dot{v}_d + \frac{C_v}{M} v_d = \frac{F_u - F_1 - F_2}{M} \\
&\left(\dot{v}_d + \frac{C_v}{M} v_d\right) e^{\frac{C_v}{M}t} = \frac{F_u - F_1 - F_2}{M} \\
&\int_0^{\Delta t} \frac{d}{dt}\left(v_d e^{\frac{C_v}{M}t}\right) = \frac{F_u - F_1 - F_2}{M} \int_0^{\Delta t} e^{\frac{C_v}{M}t} d\tau \\
&v_d\left(\Delta t\right) e^{\frac{C_v}{M}t} - v_d\left(0\right) = \frac{F_u - F_1 - F_2}{C_v}\left(e^{\frac{C_v}{M}\Delta t} - 1\right)
\end{aligned}$$

$$v_d\left(\Delta t\right) = v_d\left(0\right) e^{-\frac{C_v}{M}\Delta t} + \frac{F_u - F_1 - F_2}{C_v}\left(1 - e^{\frac{C_v}{M}\Delta t}\right) \tag{A.3}$$

In the same manner, we have

$$\dot{\phi}_d\left(\Delta t\right) = \dot{\phi}_d\left(0\right) e^{-\frac{C_v}{M}\Delta t} + \frac{\tau_u - \tau_1 - \tau_2}{C_\phi}\left(1 - e^{\frac{C_\phi}{I}\Delta t}\right) \tag{A.4}$$

Therefore F_u and τ_u are given by

$$F_u = \frac{C_v\left(v_d e^{-\frac{C_v}{M}\Delta t} - v_d\left(0\right)\right)}{\left(e^{-\frac{C_v}{M}\Delta t} - 1\right)} \tag{A.5}$$

$$\tau_u = \frac{C_\phi \left(\dot{\phi}_d e^{-\frac{C_v}{M}\Delta t} - \dot{\phi}_d(0) \right)}{\left(e^{-\frac{C_v}{M}\Delta t} - 1 \right)} \tag{A.6}$$

A.2 Determining C_v and C_ϕ

The virtual damping coefficients (C_v and C_ϕ) in eq. (3.9) can be calculated as:

$$\begin{aligned} C_v &= 2(C + K_P) \\ C_\phi &= 2L^2(C + K_P) \end{aligned} \tag{A.7}$$

where $(C + K_P)$ comes from eq. (3.26), C is the actual damping coefficient and K_P is the gain of the proportional controller.

A.3 Determining α_1

The parameter α_1 figure A.1 is determined from the IR proximity sensor data. When the robot approaches an obstacle, IR proximity sensors 0 and 1 move closer to the obstacle than sensors 2 and 3, and we set $d_0 = 0$ and $d_i = d_{\max}$, $(i = 1, 2, 3)$. From the condition of the shape function given in eq. (3.14), $d_0 < \bar{d}$ then $s(0) = 1$, and we have

$$\begin{aligned} \tau_1 &= \operatorname{sgn}(v)\,\alpha_1 \\ \tau_2 &= \alpha_2 \end{aligned} \tag{A.8}$$

If the initial condition is determined as $\ddot{\phi} = \dot{\phi} = 0$, then eq. (3.9), for the calculation of τ_u, can be written as

$$\operatorname{sgn}(v)\,\alpha_1 + \alpha_2 = -\tau_u \tag{A.9}$$

By substituting eq. (A.6) into eq. (A.9), we have

$$\operatorname{sgn}(v)\,\alpha_1 + \alpha_2 = \frac{C_\phi \left(\dot{\phi}_d e^{-\frac{C_v}{M}\Delta t} - \dot{\phi}_d(0) \right)}{\left(e^{-\frac{C_v}{M}\Delta t} - 1 \right)} \tag{A.10}$$

Therefore, setting $\alpha_2 = 0$, we get

$$\bar{\alpha_1} = \frac{C_\phi \left(\dot{\phi}_d e^{-\frac{C_v}{M}\Delta t} - \dot{\phi}_d(0) \right)}{\left(e^{-\frac{C_v}{M}\Delta t} - 1 \right)} \tag{A.11}$$

A.4 Determining α_2

From eq. (3.38) in the stability analysis, we have

$$\operatorname{sgn}(v_0)\,\alpha_1 \{Lp_W - Lp_D\} + \alpha_2 \{Lp_W - Lp_D\} > 0 \tag{A.12}$$

$$\alpha_2 > -\frac{\operatorname{sgn}(v_0)\,\alpha_1 (Lp_W - Lp_D)}{(Lp_W - Lp_D) - Lp_D} \tag{A.13}$$

By substituting eq. (A.13) into eq. (3.44), we have

$$\frac{C_\phi}{I}\left[\operatorname{sgn}(v_0)\,\alpha_1 \{Lp_W - Lp_D\} + \alpha_2 \{Lp_W - Lp_D\}\right] - \alpha_1 p_W |v_0| > 0 \tag{A.14}$$

Therefore, in order to satisfy the stability condition, the choice for α_2 is given by

$$\alpha_2 > \frac{\frac{I}{C_\phi}\alpha_1 |v_0| p_W - \alpha_1 \operatorname{sgn}(v_0)(Lp_W - Lp_D)}{Lp_W - Lp_D} \tag{A.15}$$

A.5 Determining β_1

The value of β_1 is referring to the virtual force in eq. (3.9), responsible in acceleration limitation. Based on sensor configuration in figure A.1. if $d_1 = d_2 = 0$, and d_0 and d_3 equal to d_{max} , then $F_1 = \beta_1$. Since the robot is considered stopped, the virtual force F_2 is equal to zero.

Given that $\dot{v}_d = v_d = 0$, from eq. (A.5) we have

$$\begin{aligned} \beta_1 &= F_u \\ \bar{\beta_1} &= \frac{C_v \left(v_{ref} e^{-\frac{C_v}{M}\Delta t} \right)}{\left(e^{-\frac{C_v}{M}\Delta t} - 1 \right)} \end{aligned} \tag{A.16}$$

A.6 Determining β_2

The robot cannot detect distances beyond d_{max}, the furthest distance that can be measured by the IR-proximity sensors. Therefore, depending on the robot's speed, the deceleration scheme might start dangerously close to the obstacle. If the deceleration algorithm relies only on F_1, the obstacle may not be avoided in time. Therefore, to ensure sufficient deceleration time during obstacle approach, we add another virtual force, F_2. The reference value β_2 is determined by adding the virtual sensor velocities obtained by differentiating the sensor data from d_1 and d_2. If $\dot{d}_1 = \dot{d}_2 = v$, then $\dot{d}_0 = \dot{d}_3 = 0$.

As shown in figure A.1, when d_1 and d_2 approach an obstacle at constant speed (up to v_{max}) and $d_1 = d2 = \bar{d}$, $\beta_1 = 0$ during the approach time. Therefore, the reference value of β_2 that will avoid collision with the obstacle is given by

$$F_2 = \beta_2 v \tag{A.17}$$

In the above, we note that if $\beta_1 = 0$, then $F_1 = 0$, and if $v_d = 0$, we have

$$F_u = \frac{C_v}{\left(e^{-\frac{C_v}{M}\Delta t} - 1\right)} v \tag{A.18}$$

Bibliography

[1] C. Daimin, Y. Miao, Z. Daizhi, S. Binghan, and C. Xiaolei. Planning of motion trajectory and analysis of position and pose of robot based on rehabilitation training in early stage of stroke. In *International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE)*, pages 318–321, 2010

[2] Y. Qing-xiao, Y. Can, F. Zhuang, and Z. Yan-Zhengo. Research of the localization of restaurant service robot. *International Journal of Advanced Robotics Systems (ARS)* , 7(3):227–238, 2010.

[3] D. Jung, J. Heinzmann, and A. Zelinsky. Range and pose estimation for visual servoing of a mobile robot. In *International Conference of Robotics and Automation (ICRA)*, volume 2, pages 1226–1231, 1998.

[4] S. Thrun, M. Bennewitz, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte and D. Schulz. Minerva: A second-generation museum tour-guide robot. In *Proceeding of the 1999 IEEE International Conference on Robotics and Automation*, 1999.

[5] S. G. Tzafestas. *Introduction to Mobile Robot Control*. Elsevier, 2014. ISBN 978-0-12- 417049-0.

[6] P. Corke. *Robotics Vision and Control: Fundamental Algorithm in MATLAB*. Springer, 2011. ISBN 978-3-642-20144-8.

[7] R. Siegwart, I. R. Nourbakhsh, and Scaramuzza. *Introduction to Autonomous Mobile Robot*. The MIT Press, 2011.

[8] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley and Sons, Inc., 2006.

[9] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki and S. Thrun. *Principles of Robot Motion; Theory, Algorithms and Implementation*. John Wiley and Sons, Inc., 2005.

[10] W. Zhang, Y.Wang, Y. Li, and M. Li. Autonomous target interception using hybrid sensor networks. *Intelligent Control and Automation*, 2:196–202, 2011.

[11] C. Tsai and K. Song. Dynamic visual tracking control of a mobile robot with image noise and occlusion robustness. *Image and Vision Computing*, 27:1007–1022, 2009.

[12] C. Hu, X. Ma, X. Dai, and K. Qian. Reliable people tracking approach for mobile robot in indoor environment. *Robotics and Computer-Integrated Manufacturing*, 26:174–179, 2010.

[13] R. Munoz-Salinaz, E. Aguirre, and M. Garcia-Silvente. People detection and tracking using stereo vision and color.*Image and Vision Computing*, 25:995–1007, 2007.

[14] A. Ohya, A. Kosaka, and A. Kak. Vision-based navigation by a mobile robot with obstacle avoiance using single-camera vision and ultrasonic sensing. *IEEE Transactions on Robotics and Automation*, 14(6):969–978, 1998.

[15] A. Tsalatsanis, K. Valavanis, and A. Yalcin. Vision based tracking and collision avoidance for mobile robot. *J. Intell. Robot Syst.*, 48:285–304, 2007.

[16] M. Kam, X. Zhu, and P. Kalata. Sensor fusion for mobile robot navigation. *In Proceeding of IEEE*, volume 85, pages 108–119, 1997. doi: 10.1109/5.554205.

[17] L. Jetto, S. Longhi, and D. Vitali. Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted kalman filter. *Control Engineering Practice*, 7: 763–771, 1999.

[18] D.L. Hall and J. Llinas. An introduction to multi-sensor data fusion. In Proceeding of IEEE, volume 85, pages 6–23, 1997. doi: 10.1109/5.554205.

[19] R. C. Luo, NW. Chang, SC. Lin, and SC. Wu. Human tracking and following using sensor fusion approach for mobile assistive companion robot. In *35th Annual Conference of IEEE Industrial Electronic, 2009 IECON'09*, pages 2235–2240, 2009.

[20] YT.Wang, YC. Chen, and MC. Lin. Dynamic object tracking control for a nonholonomic wheeled autonomous robot. *Tamkang Journal of Science and Engineering*, 12(3):339–350, 2009.

[21] E. Machida, M. Cao, T. Murao, and H. Hashimoto. Human motion tracking of mobile robot with kinect 3d sensor. In *SICE Annual Conference 2012*, pages 2207–2211, 2012.

[22] M.N. Abu Bakar and A. Saad. A monocular vision-based specific person detection system for mobile robot application. In *International Symposium on Robotics and Intelligent Sensors, IRIS 2012, Procedia Engineering*, volume 41, 2012.

[23] L. Xia, CC. Chen, and J.K. Aggarwal. Human detection using depth information by kinect. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 15–22, 2011.

[24] G. Medioni, A. Francois, M. Siddiqui, K. Kim, and H. Yoon. Robust real-time vision modules for a personal service robot. *Computer Vision and Image Understanding*, 108:196 – 203, 2007.

[25] N. R. Gans and S. A. Hutchinson. A stable vision-based control scheme for nonholonomic vehicles to keep a landmark in the field of view. In I*EEE International Conference on Robotics and Automation*, 2007.

[26] H. Hugli, F. Tieche, and C. Facchinetti. Integration of vision in autonomous mobile robotics. In *IEEE International Conference on Systems, Man, and Cybernetics, 1994. Human Information and Techonology*, volume 1, pages 1047–1052, 1994.

[27] J. Hernandes-Lopez, A. Quintanilla-Olvera, J. Lopez-Ramirez, F. Rangel-Butanda, M. Ibarra-Manzano, and D. Almanza-Ojeda. Detecting object using color and depth segmentation with kinect sensor. In *The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science (Procedia Technology)*, volume 3, 2012.

[28] K Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping application. *Sensors*, pages 1437–1454, 2012.

[29] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using kinect-style depth camera for dense 3d modeling of indoor environment. *The International Journal of Robotics Research*, 31(5):647–663, 2012.

[30] F. Jurado, G. Palacios, and F. Flores. 3d color-based tracking system for real-time using kinect sensor. In *SOMIXXVII Congreso de Instrumentaci*, 2012.

[31] M. Ju, C. K. Lin, D. Lin, I. Hwang, and S. Chen. A rehabilitation robot with force-position hybrid fuzzy controller: Hybrid fuzzy control of rehabilitation robot. *IEEE Transaction on Neural Systems and Rehabilitation Engineering*, 13(3):349–358, 2005.

[32] E. Akdog and M.A.Adli. he design and control of a therapeutic exercise robot for lower limb rehabilitation: Physiotherabot. *Mechatronics Journal*, 213:509–522, 2011.

[33] W. S. Lob. Robotics transportation. *Clinical Chemistry*, 36:1544–1550, 1990.

[34] J. Evans, B. Krisnamurthy, B. Barrows, T. Skewis, and V. Lumelsky. Handling real-world motion planning: A hospital transport robot. *IEEE Control System*, 12:15–19, 1992.

[35] A. G. Ozkil, Z. Fan, S. Dawids, H. Aanaes, J. K. Kristensen, and K. H. Christensen. Service robots for hospitals: A case study of transportation task in a hospital. In *Proceedings of the IEEE International Conference on Automation and Logistic*, pages 61–68, 2009.

[36] M. Takahashi, T. Suzuki, H. Shitamoto, T. Moriguchi, and K. Yoshida. Developing a mobile robot for transport applications in the hospital domain. *Robotics and Autonomous System*, 58:889–899, 2010.

[37] F. Capezio, F. Mastrogiovanni, A. Scalmato, A. Sgorbissa, P. Vernazza, T. Vernazza, and R. Zaccaria. Mobile robots in hospital environments: an installation case study. In *Proceedings of the 5th European Conference on Mobile Robots (ECMR 2011)*, pages 61–68, 2011.

[38] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robot. The *International Journal of Robotics Research*, 5(1):345–356, 1986. doi: 10.1017/S0263574707004092.

[39] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In*Proceedings of IEEE Conference on Robotics and Automation*, pages 1398–1404, 1991.

[40] N. Ohnishi and A. Imiya. *Navigation of Nonholonomic Mobile Robot Using Optical Flow and Visual Potential Field*, volume 4931, pages 412–426. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78156-1. doi: 10.1007/978-3-540-78157-8 32.

[41] C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[42] J. Borenstein and Y. Koren.Virtual Obstacle Concept for Local-minimum-recovery in Potential-field Based Navigation. In*Proceedings of 2000 IEEE International Conference on Robotics and Automation*, pages 983-988, 2000.

[43] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.

[44] G.C. Pettinaro, I.W. Kwee, L.M. Gambardella, F. Mondada, S. Nolfi D. Floreano, J. Daneubourg, and M. Dorigo. Swarm robotics: A different approach to service robot. In *Proceeding of the 33rd International Symposium on Robotics*, 2002.

[45] A. Turgut, H. Celikkanat, F. Gokce, and E. Sahin. Self-organized flocking in mobile robot swarms. *Swarm Intell*, 2:97–120, 2008.

[46] N. Xiong, J. He, Y. Yang, Y. He, T. Kim, and C.Lin. A survey on decentralized flocking schemes for a set of autonomous mobile robot. *Journal of Communication*, 5(1):31–38, 2010.

[47] R. Havangi, M.A. Nekoui, and M. Teshnehlab. A multi swarm particle filter for mobile robot localization. *IJCSI International Journal of Computer Science Issues*, 7(2):15–22, 2010.

[48] B. Eikenberry, O. A. Yakimenko, and M. Romano. A vision based navigation among multiple flocking robot: Modeling and simulation. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2006.

[49] R.Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[50] K. Ishii and T. Miki. Mobile robot platforms for artificial and swarm intelligence researches. In *International Congress Series 1301*, 2007.

[51] H. Lee, E-J. Jung, B-J. Yi, and Y. Choi. Navigation strategy of multiple mobile robot systems based on the null-space projection method. *International Journal of Control, Automation, and Systems*, 9(2):384–390, 2011.

[52] T. Germa, F. Lerasle, N. Oudah, and V.Cadenat. Vision and rfid data fusion for tracking people in crowds by a mobile robot. *Computer Vision and Image Understanding*, 114: 641–651, 2010.

[53] K. Prathyusha, V. Harini, and S. Balaji. Design and development of a RFID based mobile robot. *International Journal of Engineering Science Advanced Technology*, 1(30-35), 2011.

[54] T. Tammet and A. Kuusik J. Vain. Distributed coordination of mobile robots using RFID technology. In *Proceeding of the 8th WSEAS Int. Conference on Automatic Control, Modeling and Simulation*, 2006.

[55] S. Park and S. Hashimoto. Indoor localization for autonomous mobile robot based on passive rfid. In *Proceeding of the 2008 IEEE, International Conference on Robotics and Biomimetics*, 2009.

[56] G. L. Mariottini, G. Oriolo, and D. Prattichizzo. Image-based visual servoing for nonholonomic mobile robots using epipolar geometry. *IEEE Transactions on Robotics*, 23 (1):87–100, 2007.

[57] JX. Wang and XF. Jin. *Moving Object Detection and Tracking in Mobile Robot System Based on Omni-Vision and Laser Range-finder*, volume 2, pages 133–139. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-25788-9. doi: 10.1007/978-3-642-25789-6 20.

[58] L. Huang. A potential field approach for controlling a mobile robot to track a moving object. In *IEEE International Symposium on Intelligent Control Part of IEEE Mulitconference on Systems and Control*, pages 65–70, 2008.

[59] L. Huang. Velocity planning for a mobile robot to track a moving target - a potential field approach. *Robotics and Autonomous Systems*, 57:55–63, 2009.

[60] G. L. Mariottini, S. Scheggi, F. Morbidi, and D. Prattichizzo. An accurate and robust visual-compass algorithm for robot mounted omnidirectional cameras. *Robotics and Autonomous Systems*, 60:1179–1190, 2012.

[61] R. Carelli, C.M. Soria, and B. Morales. Vision-based tracking control for mobile robots. In Advanced Robotics, 2005. *ICAR'05. Proceedings, 12th International Conference on*, pages 148–152, 2005.

[62] L. Freda and G. Oriolo. Vision-based interception of a moving target with a nonholonomic mobile robot.*Robotics and Autonomous System*, 55:419–432, 2007.

[63] A. De Luca, G. Oriolo, and P. R. Giordano. Image-based visual servoing schemes for nonholonomic mobile manipulator.*Robotica*, 25:131-145, 2007.

[64] F. Belkhouche and B. Belkhouche. On the tracking and interception of a moving object by a wheeled mobile robot. In *IEEE Conference on Robotics, Automation and Mechatronics 2004*, pages 130–135, 2004.

[65] N.S. Nise.*Control Systems Engineering-6th edition*. John Wiley and Sons Inc., 2011.

[66] D. Sun, J. Zhu, C. Lai, and S.K. Tso. A visual sensing application to a climbing cleaning robot on the glass surface. *Mechatronics Journal*, 14:1089–1104, 2004.

[67] V. Sezer and M. Gokasan. A novel obstacle avoidance algorithm: Follow the gap method. *Robotics and Autonomous System*, 60:1123–1134, 2012.

[68] G. Li, A. Yamashita, H. Asama, and Y. Tamura. An efficient improved artificial potential field based regression search method for robot path planning. In *International Conference on Mechatronics and Automation (ICMA)*, pages 1227–1232, 2012.

[69] F. Kunwar and B. Benhabib. Advanced predictive guidance navigation for mobile robots: A novel strategy for rendezvous in dynamic setting. In *International Journal on Smart Sensing and Intelligent System*, volume 1, pages 858–890, 2008.

[70] J. Kim and Y. Do. Moving obstacle avoidance of a mobile robot using single camera. *In Procedia Engineering, International Symposium on Robotics and Intelligent Sensor (IRIS)*, volume 41, pages 911–916, 2012.

[71] J. Llinas and D.L. Hall. Moving obstacle avoidance of a mobile robot using single camera. *In Proceeding of the 1998 IEEE International Symposium on Circuits and System*, volume 6, pages 537–540, 1998.

[72] T. Haidegger, M. Barreto, P. Goncalves, M.K. Habib, V. Ragavan, H.Li, A. Vaccarella, R. Perrone, and E. Prestes. Applied ontologies and standards for service robots. *Robotics and Autonomous Systems*, 61:1215–1223, 2013.

[73] Tschichold-Gurman, S.J. Vestli, and G. Schweitzer. The service robot mops: First operating experiences. *Robotics and Autonomous Systems*, 61:165–173, 2001.

[74] X. Wei, M. Jiachen, Y. Mingli, and L. Kui. Agent-oriented architecture for intelligent service robot. In *IEEE/ASMEE Transac*, pages 5964–5967, 2010.

[75] Y. Wu, C. Fassert, and A. Rigaud. Designing robot for elderly: Appearance issue and beyond. *Archives of Gerontology and Geriatrics*, 54:121–126, 2012.

[76] JH. Tzou and K.L. Su. The development of the restaurant service mobile robot with a laser positioning system. *In Proceeding of the 27th Chinese Control Conference*, pages 662–666, 2008.

[77] EJ. Jung, J.H. Lee, BJ. Yi, S. Yuta, and ST. Noh. Development of a laser-range-finder based human tracking and control algorithm for a marathoner service robot. *IEEE/ASME Transactions on Mechatronics*, 19(6):1963–1975, 2014.

[78] V. Alvarez-Santoz, R. Iglesias, X.M. Pardo, C.V. Regueiro, and A. Candedo-Rodrigues. Gesture-based interaction with voice feedback for a tour-guide robot. *J. Vis. Commun. Image R.*, 25:499–509, 2014.

[79] N. Uchiyama, T. Hashimoto, S. Sano, and S. Takagi. Model-reference control approach to obstacle avoidance for a human-operated mobile robot. *IEEE Trans. Industrial Electronics*, 56(10):3892–3896, 2009.

[80] N. Uchiyama and T. Hashimoto. Model reference control of human-operated mobile robot for object transportation. *Factory Automation Book*, pages 557–568, 2010.

[81] A. Sgorbissa and R. Zaccaria. Planning and obstacle avoidance in mobile robotics. *Robotics and Autonomous System*, 60:628–6238, 2012.

[82] T. Dewi, N. Uchiyama, and S. Sano. Simple obstacle avoidance for a mobile robot moving through via points. In *IEEE/SICE International Symposium on System Integration (SII)*, pages 111–114, 2013.

[83] J. Vallve and J. Andrade-Cetto. Potential information field for mobile robot exploration. *Journal Robotics and Autonomous System*, 69:68–79, 2014.

[84] N. Uchiyama, T. Dewi, and S. Sano. Collision avoidance control for a human-operated four-wheeled mobile robot. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 228:2278–2284, 2014.

[85] H. Kazerooni. Human-robot interaction via the transfer of power and information signals. *IEEE Trans. Systems, Man, and Cybernetics*, 20(2):450–463, 1990.

[86] H. Kazerooni and R. Steger. The berkeley lower extremity exoskeleton. *ASME J. Dyn. Syst. Meas., Control*, 128(1):14–25, 2006.

[87] M. A. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. A. Moore, B. Gillespie, and P. Akella. Cobot architecture. *IEEE Trans. Robotics and Automation*, 17(4):377–390, 2001.

[88] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hagera. Vision-assisted control for manipulation using virtual fixtures. *IEEE Trans. Robotics and Automation*, 20 (6):953–966, 2001.

[89] R. J. Anderson and M.W. Spong. Bilateral control of teleoperators with time delay. *IEEE Trans. Automatic Control*, 34(5):494–501, 1989.

[90] D. A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Trans. Robotics and Automation*, 9(5):624–637, 1993.

[91] A. Chakravarthy and D. Ghose. Obstacle avoidance in a dynamic environment: A collision cone approach. *IEEE Trans. Systems, Man and Cybernetics-Part A: System and Humans*, 28(5):562–574, 1998.

[92] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Trans. Systems, Man, and Cybernetics*, 19(5):1179–1187, 1989.

[93] F. Lamiraux, D. Bonnafous, and O. Lefebvre. Reactive path deformation for non-holonomic mobile robots.*IEEE Trans. Robotics and Automation*, 20(6):967–977, 2004.

[94] Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4:22–23, 1997.

[95] Ogren and A N. E. Leonard. A convergent dynamic window approach to obstacle avoidance. *IEEE Trans. Robotics and Automation*, 21:188–195, 2005.

[96] D. Reichardt and J. Schick. Collision avoidance in dynamic environments applied to autonomous vehicle guidance on the motorway. In *Proceeding of the Intelligent Vehicle Symposium*, pages 74–78, 1994.

[97] J. C. Gerdes and E. J. Rossetter. A unified approach to driver assistance systems based on artificial potential fields. *Trans. ASME J. Dynamic Systems, Measurement, and Control*, 23:431–438, 2001.

[98] E. J. Rossetter, J. P. Switkes, and J. C. Gerdes. Experimental validation of the potential field driver assistance system. *Int. J. of Automotive Technology*, 5(2):95–108, 2004.

[99] M.T. Wolf and J. W. Burdick. Artificial potential functions for highway driving with collision avoidance. In *Proceeding of IEEE International Conference on Robotics and Automation*, pages 3731–3736, 2008.

[100] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review*, E51(5):4282–4286, 1995.

[101] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.

[102] T. I. Lakoba and D. J. Kaup. Modification of the helbing-molnar-farkas-vicsek social force model for pedestrian evolutionn. *The Society for Modeling and Simulation International*, 81(5):339–352, 2005.

[103] T. Dewi, N. Uchiyama, S. Sano, and H. Takahashi. Swarm robot control for human services and moving rehabilitation by sensor fusion.*Journal of Robotics*, Article ID.278659, 2014.

[104] Petar Curkovic, Bojan Jerbic, and Tomislav Stipancic, *Comparison of Swarm Optimization and Genetic Algorithm for Mobile Robot Navigation*, chapter 3, pages 47–58. InTech, 2010. ISBN 978-953-307-075-9.

[105] R. Haghighi and C.C. Cheah. Multi-group coordination control for robot swarms. *Automatica*, 48:2526–253, 2012.

[106] S.P. Hou, C.C. Cheah, and J.J.E. Slotine. Dynamic region following formation control for a swarm of robots. In *IEEE International Conference on and Automation (ICRA '09)*, 2009.

[107] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques. Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, 44:1343–1349, 2008.

[108] H.A. Poonawala, A. C. Satici, and M.W. Spong. Leader-follower formation control of nonholonomic wheeled mobile robot using only position measurements. In *Control Conference (ASCC), 2013 9th Asian*, 2013. ISBN 978-1-4673-5767-8. doi: 10.1109/ASCC.2013.6606313.

[109] R. Sigal. Algorithm for the Routh-Hurwitz stability test. *Mathl Comput. Modeling*, 13(8):69-77, 1990.

[110] M. Lieberei, C. Ruwwe, B. Keck, O. Rusch, and U. Zolzer. Computing camera orientation relative to a world coordinate frame by detecting its projected axes. In *Proceeding SIGGRAPH '08*, 2008.

[111] S. Gasparrini, E. Cippitelli, S. Spinsante, and E. Gambi. A depth-based fall detection system using a kinect sensor. *Sensors*, 14:2756–2775, 2014. doi: 10.3390/s140202756.

[112] T. Dewi, N. Uchiyama, and S. Sano. Service mobile robot control for tracking a moving object with collision avoidance. In *IEEE Workshop on Advanced Robotics and its Social Impact (ARSO 2015)*, pages 148–152, 2015.

[113] T. Dewi, N. Uchiyama, and S. Sano. Vision control of a mobile robot tracking a moving object under occlusion and obstacles. In *International Conference on Advanced Technology in Experimental Mechanic (ATEM'15)*, pages 858–890, 2015.

[114] T. Dewi, N. Uchiyama, and S. Sano. Vision-based object recognition and tracking control of a service mobile robot. 2015 *Unpublished.*

[115] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. In *IEEE International Conference on Robotics and Automation*, 1990. Proceedings, pages 384– 389, 1990.

[116] H. Takahashi. (2012). Control system design for improved operability of the electric wheelchair (Master's Thesis). Retrieved from System Engineering Laboratory, Toyohashi University of Technology.

[117] J. Satake, and J. Miura. Robust Stereo-Based Person Detection and Tracking for a Person Following Robot. In *IEEE ICRA 2009 Workshop on People and Detection*, Proceeding, 2009.

[118] J. Satake, and J. Miura. Experimental Evaluation of Stereo-Based Person Tracking in Real Environment. In *MVA2013 IAPR International Conference on Machine Vision Application*, Proceeding, 2013.

[119] Y. Tan and Z. Zheng. Research advance in swarm robotics. In *Defence Technology*, 9:18-39, doi: 10.1016/j.dt.2013.03.001, 2013.

Publications

Journal papers:

J. 1 Tresna Dewi, Naoki Uchiyama, Shigenori Sano, and Hiroki Takahashi, "Swarm Robot Control for Human Services and Moving Rehabilitation by Sensor Fusion", Journal of Robotics, vol. 2014, Article ID. 278659, 2014.

J. 2 Naoki Uchiyama, Tresna Dewi, and Shigenori Sano, "Collision Avoidance Control for a Human-Operated Four-Wheeled Mobile Robot", Proceeding of the Institute of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, vol. 228. No. 13, pp. 2278-2284, 2014.

Conference papers:

C. 1 Tresna Dewi, Naoki Uchiyama, and Shigenori Sano, "Simple Obstacle Avoidance for a Mobile Robot Moving through Via Points", IEEE/SICE International Symposium on System Integration (SII 2013), pp. 2207-2211, December 15-17, 2013, Kobe, Japan.

C. 2 Tresna Dewi, Naoki Uchiyama, and Shigenori Sano, "Service Mobile Robot Control for Tracking a Moving Object with Collision Avoidance", IEEE International Workshop on Advance Robotics and its Social Impacts (ARSO 2015), Presentation No. frPM1.5, July 1–3, 2015, Lyon, France.

C. 3 Tresna Dewi, Naoki Uchiyama, and Shigenori Sano, "Vision Control of a Mobile Robot Tracking a Moving Object under Occlusion and Obstacles", International Conference on Advanced Technology in Experimental Mechanics 2015 (ATEM 15), October 4-8, 2015, Toyohashi, Japan.