TOWARDS A ROUGH-FUZZY PERCEPTION-BASED COMPUTING FOR VISION-BASED INDOOR NAVIGATION

by

Tong Duan

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements of the degree of

Master of Science

Department of Electrical and Computer Engineering

University of Manitoba

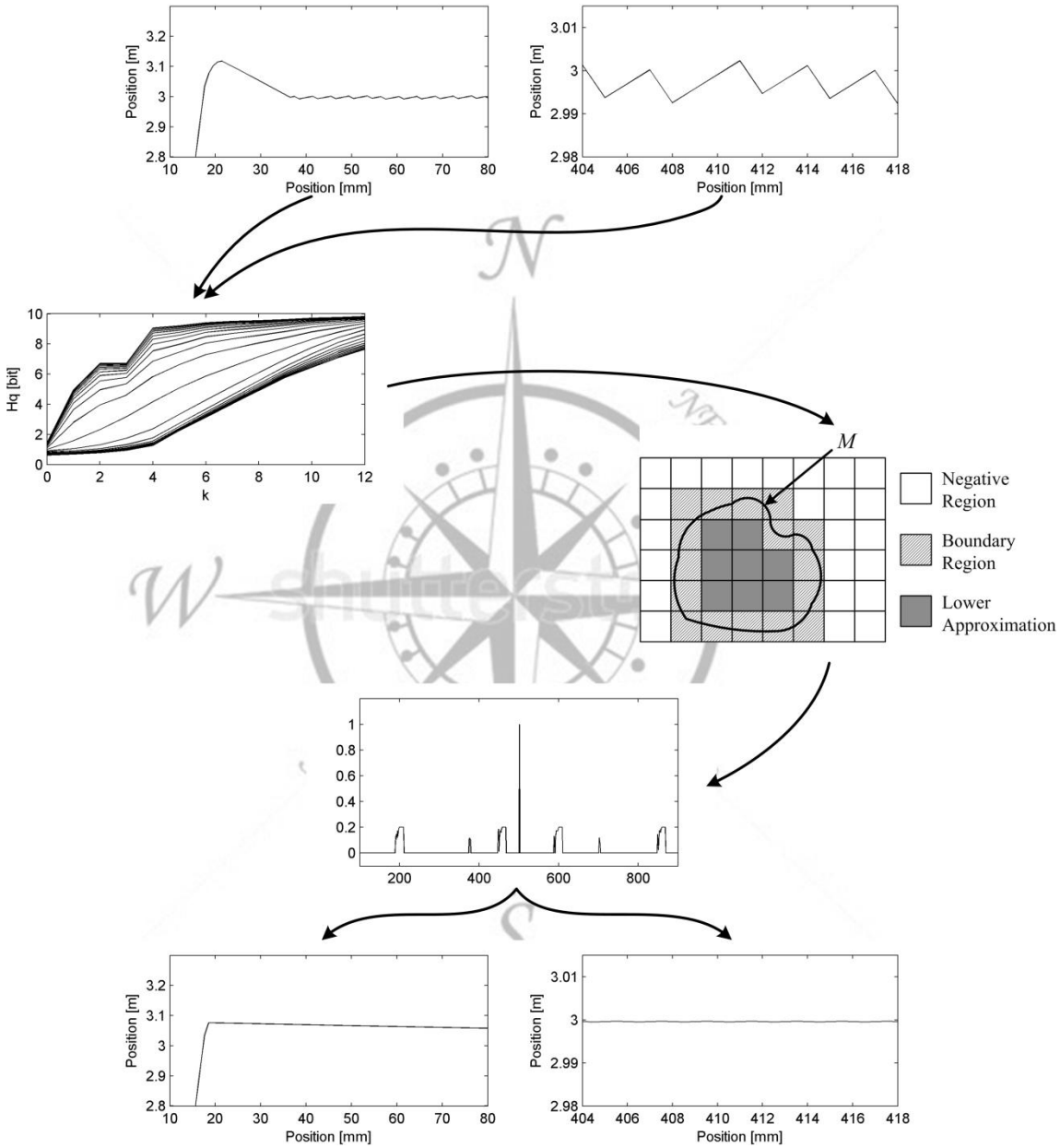
Winnipeg, Canada

ABSTRACT

An indoor environment could be defined by a complex layout in a compact space. Since mobile robots can be used as substitute for human beings to access harmful and inaccessible locations, the research of autonomous indoor navigation has attracted much interest. In general, a mobile robot navigates in an indoor environment where acquired data are limited. Furthermore, sensor measurements may contain errors in a number of situations. Therefore, the complexity of indoor environment and ability of sensors have determined that it is an insufficient to merely compute with data.

This thesis presents a new rough-fuzzy approach to perception-based computing for an indoor navigation algorithm. This approach to perceptual computing is being developed to store, analyze and summarize existing experience in given environment so that the machine is able to detect current situation and respond optimally. To improve uncertainty reasoning of fuzzy logic control, a rough set theory is integrated to regulate inputs before applying fuzzy inference rules. The behaviour extraction is evaluated and adjusted through entropy-based measures and multi-scale analysis. The rough-fuzzy based control algorithm aims to minimize overshoot and optimize transient-state period during navigation. The proposed algorithm is tested through simulations and experiments using practical common situations. The performance is evaluated with respect to desired path keeping and transient-state adaptability.

VISUAL ABSTRACT



ACKNOWLEDGEMENTS

There were many people who supported and encouraged me throughout my research life and I would not have achieved my goal without them.

I would like to thank my advisor Professor Witold Kinsner for his endless contribution throughout this degree. He encouraged me to discover many new dimensions that I have never met in my life. My research attitude, study procedures and protocols, and problem-solving approaches are strongly formed under his guidance. It is my honor to be one of his students and to have a chance to achieve many goals through his tremendous help and supervision.

It is my honour to have Professor James Peters and Professor Carson Leung to be my examination committee. Their advices for the development of high quality thesis are extremely of importance. I would like to express my gratitude to them for their tremendous effort.

I am so grateful to current and previous members of the Delta Research Group for their help and encouragement. In particular, I would like to thank Mohamed Nasri and Hieu for their kindness help with my research and life abroad. The diligent and rigorous group spirit always inspires me to explore the next difficult peak. Furthermore, I would like to express my appreciation to my friends in the Faulty of Engineering and

Department of Computer Science. In particular, special thanks to Fan Jiang, Aditya Kedia, Tao Chen, Foster Sun, Yu Zhou, and Meiting Li for their valuable advice and help throughout my master's study.

I bring my deepest respect to my father Jianhong and my mother Huilian without whom I would ever have been the man I am today. I cannot have this achievement without their selfless support at each stage of my life. I would like to thank my wife Feifei for her precious accompany. She makes my life in Winnipeg colourful and unforgettable. Special thanks to my brother Lei for helping me with my adventure in Canada.

Finally, but certainly not last, I am thankful to all my previous teachers and professors from the primary to the graduate level. My success with this current endeavour is a result of their teaching and effort.

My supportive friends that were there when I needed a hand and encouraging words also desire my gratitude.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI	**A**rtificial **I**ntelligence
API	**A**pplication **P**rogramming **I**nterface
BBC	**B**oundary **B**ased **C**ontrol
CFBC	**C**onventional **F**uzzy **B**ased **C**ontrol
FPS	**F**rame **P**er Second
GUI	**G**raphical **U**ser **I**nterface
IBVS	**I**mage **B**ased **V**isual **S**ervoing
IR	**I**nfra**r**ed
LE	Learning Entropy
PBC	**P**erception-**B**ased **C**omputing
PBVS	**P**osition **B**ased **V**isual **S**ervoing
PID	**P**roportional-**I**ntegral-**D**erivative
pmf	**P**robability **M**ass **F**unction
PWM	**P**ulse **W**idth **M**odulation

RFBC **R**ough-**F**uzzy **B**ased **C**ontrol

SDK **S**oftware **D**evelopment **K**it

SISO **S**ingle **I**nput **S**ingle **O**utput

SNN **S**piking **N**eural **N**etwork

SNR **S**ignal to **N**oise **R**atio

UPS **U**ninterruptible **P**ower **S**upply

LIST OF SYMBOLS

Scalars:

β	Parameter of Gaussian fuzzy membership function
γ	Detection sensitivity parameter of a Learning Entropy
θ	Orientation difference
σ	Parameter of Gaussian fuzzy membership function
τ	Learning rate of a Learning Entropy
ω	A mobile robot's heading angle
a	An arbitrary variable
d	Measured distance between a mobile robot and a wall
d_{min}	Minimum measured distance of a spiking neural network
d_{m3}	Maximum distance threshold of a spiking neural network
d_{m1}	Minimum distance threshold of a spiking neural network
d_{m2}	Moderate distance threshold of a spiking neural network
D_q	Rényi's dimension at the moment order of q
$f(\bullet)$	An output of fuzzy logic control
h_1, h_2, h_3	Neurons in hidden layer of a spiking neural network

$H(\bullet)$	Shannon's entropy of a variable or an event
$H_q(\bullet)$	Rényi's entropy of a variable or an event
$L(\mathbf{x}_{t+n})$	Learning Entropy of measured data $\mathbf{x}$ at time $t+n$
m	Boundary parameters of membership functions
MP(•)	Membrane potential of a neuron in hidden layer or output layer
MP_{m2}	Maximum membrane potential threshold of neuron o_1
MP_{m1}	Minimum membrane potential threshold of neuron o_1
N	Number of potential values of measured distance contained in a data vector $\mathbf{x}$
M	Number of measures in a trajectory $\mathbf{J}$
o_1	Neuron in output layer of a spiking neural network
$p(\bullet)$	Probability of a variable or an event
q	Moment order of Rényi's entropy
$S_{ns}(\bullet)$	Number of spikes generated by a neuron in hidden layer
$I(\bullet)$	Self-information of a potential value of a variable
V_{rest}	Rest state of an output neuron in a spiking neural network
w_{h1}, w_{h2}, w_{h3}	Connection weights of neuron h_1, h_2, and h_3,

Vectors:

$\mathbf{B}_t$	Kalman filter control input model at time t
$\mathbf{C}_{t,t+n}$	Quadratic multiplication of a collection of sensory input data vector $\mathbf{x}$ in the time interval of $[t, t+n]$

$\mathbf{e}_{t+n}$	Difference of a measured data and predicted data at time $t+n$
$\mathbf{F}_t$	Kalman filter state transition matrix at time t
$\mathbf{J}$	Trajectory formed by measured distance contained in $\mathbf{x}$
$\mathbf{u}_t$	Kalman filter control vector at time t
$\mathbf{K}_t$	Kalman gain at time t
$\mathbf{P}_{t\|t-1}$	Kalman filter *a prior* estimate error covariance at time t given time t-1
$\mathbf{q}_t$	Kalman filter process noise at time t
$\mathbf{Q}$	Kalman filter process noise covariance (assume to be a constant)
$\mathbf{x}$	A sensory input data vector of a mobile robot in real-time navigation
$\mathbf{w}_t$	Kalman filter measurement noise at time t
$\mathbf{W}$	Kalman filter measurement noise covariance, assume to be a constant
$\Delta\mathbf{w}_{t+n}$	Weights of a measured data vector $\mathbf{x}$ at time $t+n$
$\mathbf{z}_t$	Kalman filter measurement at time t
$\mathbf{Z}_t$	Kalman filter measurement model at time t

Functions and Other Symbols:

$\alpha_R(M)$	Vagueness degree of a rough set M given equivalence relation rules R
$\lambda_M^R(\bullet)$	Rough membership function of a rough set M given equivalence relation rules R

$\mu_F(\bullet)$	Fuzzy membership function of a fuzzy set F
ρ	Information function of an interaction model
G_{m_i}	Gaussian fuzzy membership function with center at m_i
$[a]_R$	Equivalence class of a given equivalence relation rules R
$N(\bullet,\bullet)$	Gaussian distribution function
$P(\bullet)$	Probability mass function of a discrete variable
$r_F^R(\bullet)$	Rough-fuzzy membership function of a fuzzy set F given equivalence relation rules R
A	A set of features associated with X
O	Approximation space
$t(a)$	features of a variable a
$BN_R(\bullet)$	Boundary region of a rough set given equivalence relation rules R
F	An arbitrary fuzzy set
M	An arbitrary rough set
$l(a)$	Fuzzy constrains associated with a
R	Equivalence relation rules
$\underline{R}(M)$	Lower approximation space of a set M
$\overline{R}(M)$	Upper approximation space of a set M
S	Interaction model
V	A union of sets of feature values of the given A
V_{atr}	A set of feature values of a feature in A

X	A set of data consists of historical sensory data vector $\mathbf{x}$
$[\bullet]$	A collection of elements (allows equity of elements)
$\{\bullet\}$	A collection of elements (precludes equity of elements)
$\langle\bullet\rangle$	A collection of sets
$(\bullet)_{t\|t-1}$	*A prior* estimate of the variable $(\bullet)$ at time t given t-1 measurements
$(\bullet)_{t\|t}$	*A posterior* estimate of the variable $(\bullet)$ at time t given t measurements

Chapter 1

INTRODUCTION

1.1 Motivation

Design of indoor autonomous mobile robots and mobility systems has experienced an increasing interest from the scientific and engineering communities due to their good application prospects [RaTS09]. A variety of applications have been implemented for civilian and military purposes (*e.g.*, mobile robots for warehousing, indoor service robots, and weapon transportation robots) [RaMa11] [LoHa07]. Generally, indoor autonomous mobile robots are required to find their way out, as well as respond to a number of complex situations such as the ability of self-localization, path selection, and obstacles avoidance. Consequently, efficient indoor navigation algorithms are essential to an autonomous mobile robot.

In order to implement an indoor navigation algorithm for a mobile robot, it is necessary to develop schemes that perceive current state of a mobile robot, the indoor environment, and proper results should be reasoned to react to various situations.

[Kawa91]. In other words, a mobile robot should be trained to recognize the environment in order to determine its location and condition, and then select proper actions.

Navigation involves path planning, collision avoidance and motion planning. Primary academic research interests in the navigation field are task performance and completion time. In practice, these goals are achieved through optimization of vehicles' path and velocity [LiWK06]. Navigation behaviour can be decomposed and analyzed from three levels [ThSu00]. The highest level behaviour is the task-oriented behaviour which consists of two subtasks: wall-following and goal seeking. The middle level behaviour is an obstacle avoiding behaviour. The lowest level is an emergency behaviour.

With the development of electronics techniques, a variety of reliable and inexpensive sensors provide more possible solutions for navigation related issues. On the one hand, new type and high efficient sensory data lead to precise control. Nevertheless, sensory measurements could contain errors in a number of situations, which should to be minimized. On the other hand, the data processing efficiency of current navigation algorithms gradually falls behind the evolution of hardware. For the particular case of an indoor environment, one considerable limit of a mobile robot is the short reaction time to situations (*e.g.,* obstacles and corners) due to the compact space. In order to overcome such problem, the control algorithm should be not only capable of dealing with a large amount of data more efficiently, but also be able to perceive information about complex situations.

Solutions of this problem can be improved by understanding the way that human beings collect information, analyze and make decisions, which is normally categorized as

the intelligent behaviour [Arau06]. In general, the intelligent behaviour is interpreted as the ability of perceiving essential information of self-state and the environment, thereby decide a proper action to achieve the objective. When compared with current techniques, the intelligent behaviour demonstrates advantages through information extraction and summarization. In other words, an intelligent system deals not only with collected data, but also senses induced from sensory data, in order to respond to the environment optimally. Therefore, this thesis proposes an intelligent indoor navigation algorithm that aims to enhance uncertainty reasoning performance against various situations.

1.2 Problem Definition

This thesis addresses a vision-based indoor wall-following algorithm for a small autonomous robot. Wall-following behaviour is considered as an important feature of mobile robots since it can be derived into various practical scenes in navigation tasks. For instance, a path can be formed by two parallel walls. A wall with small length can be regarded as an obstacle. Wall-following task is used extensively to evaluate the robustness, stability, and efficiency of indoor navigation algorithms [LiCF08] [RaMa11] [RaTS09] [LeBZ06] [KiLy09] [SuHP95].

An indoor environment is characterized by the complex layout in the constrained space. In order to perform successful autonomous navigation in the described environment, highly accurate control and proper reactions are significant for a mobile robot. Furthermore, the irregularity of an indoor space arrangement also requires a certain understanding level of the environment. In other words, a mobile robot should be able to

grasp information about encountered situation from sensory data. Fig. 1.1 illustrates a typical office layout with a mobile robot placed at one door in the top-left corner, and it is designed to navigate autonomously through complex situations until reach the destination. The knowledge about navigation environment is limited and walls are the major reference coordinate data that a mobile robot can collect via a visual sensor. A mobile robot distorts desired path or even may be trapped in shadow regions. As a result, besides contours of objects such as walls and tables, dashed lines represent virtual walls that should be created by a mobile robot.

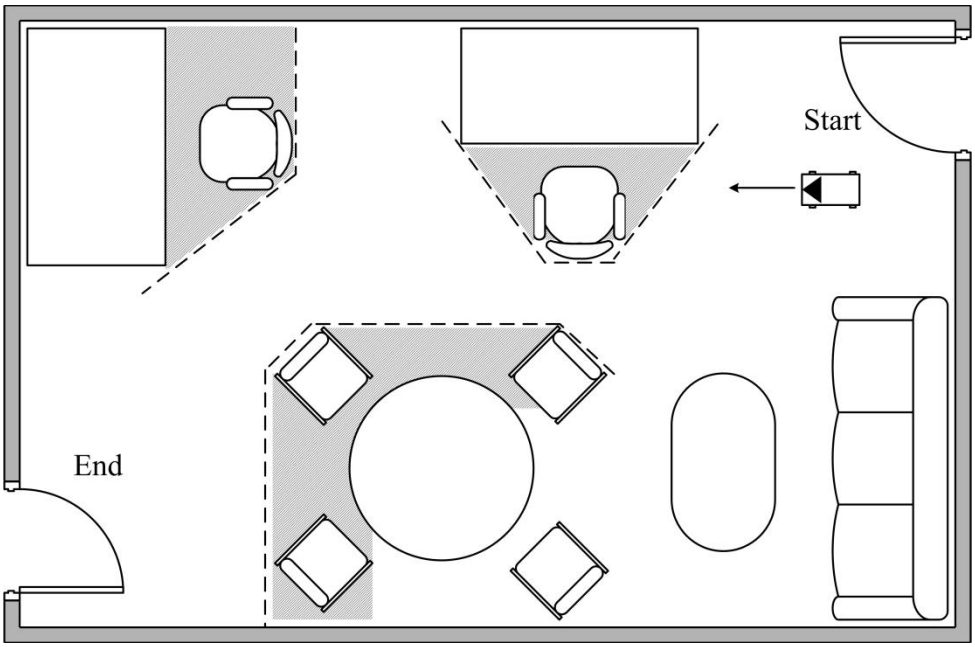


Fig. 1.1: Top view of a typical office layout.

As shown in Fig. 1.1, a common "U" path is formed by a coach, a table and two chairs. In addition, sides of a table or chair may not be as smooth as a coach, which increases the navigation difficulty. The edges of objects (*e.g.,* desk and coach) create convex and concave corners, which require accurate distance control. Furthermore, it can

be inferred that the navigation path structure could be complicated and unpredictable. In other words, a mobile robot spends the most of time to approach and lock on the desired path during the navigation period. In this case, to perform a successful wall-following task in such indoor environment, a mobile robot should (i) avoid obstacles via efficient heading angle control, (ii) try to stay on a desired path and maintain steady while following walls. It means conditions such as transient-state and overshoot should be optimized. (iii) Under the promise of former requirements, a mobile robot should also optimize navigation velocity in order to reduce task completion time.

1.3 Proposed Solution

In recent years, a number of important applications have shown that the concern of information is transferring from measure to meaning [Zade99a]. It leads to a trend of human-centric design rather than manipulation of numbers and symbols since humans have amazing capability of performing physical and mental tasks without precise measurements and computations, which is called perception [Zade01]. Perception is a biological form of human sense to acquired environmental information.

Since *perception-based computing* (PBC) has not developed a coherent set of methods or principles, this thesis attempts to propose a new implementation approach for such a perception-based computing. The proposed navigation method is on the basis of a *conventional fuzzy based control* (CFBC) mechanism. A rough set theory is integrated to filter data by extracting essential features of data before applying fuzzy inference rules.

The rough set theory was first described by Pawlak in 1982 [Pawl82]. A rough set is presented by a formal approximation of classical set theory in terms of a pair of sets which are the lower and upper approximation of the original set [Pawl97]. A lower approximation is used to describe all features that the reference set certainly has and a lower approximation represents all features that the reference set surely and possibly has.

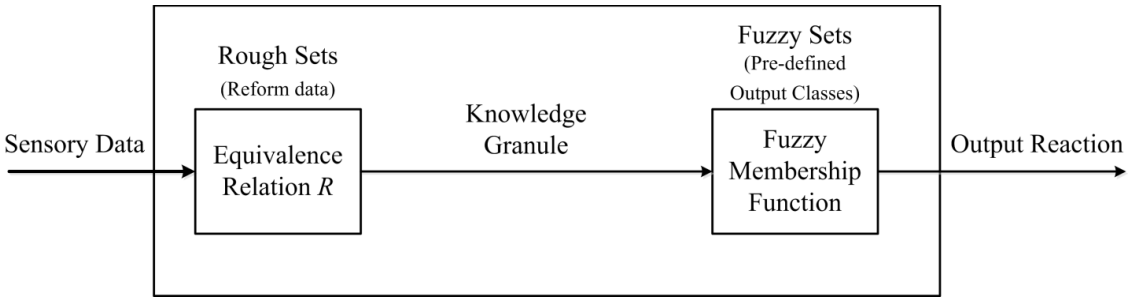


Fig. 1.2: Overview of a rough-fuzzy based control approach.

As shown in Fig. 1.2, the proposed *rough-fuzzy based control* (RFBC) approach creates knowledge granule from sensory data and interacts with the environment using intelligent behaviour. The rough set theory reforms sensory data and extracts important features to create knowledge granule by given equivalence relation rules, which is called data analysis stage. Then fuzzy set theory is applied at the reasoning stage. At this stage, pre-defined fuzzy sets and inference rule base are used to process given knowledge granule and make proper reactions.

The importance of the rough set operation can be explained from three aspects. First of all, let us assume a simple *single input single output* (SISO) control system, by applying the rough set theory the data processing pressure is reduced since impossible data are filtered out and irregular data are redressed at the analysis stage. Secondly, data

transferred to reasoning stage is properly categorized, lump processing procedure can be adopted and thus time is saved dramatically. Thirdly, compared with traditional adaptive control, the proposed perception-based computing contains extra data analysis stage. It means data are not purely regarded as numerical symbols, but information is extracted from it. In other words, the proposed rough-fuzzy based control approach steps closer to a human centric design.

1.4 Thesis Formulation

1.4.1 Thesis Statement

This thesis aims to present a new perception-based computing approach for a vision-based indoor navigation algorithm. The objective is to characterize essential properties of perception-based computing and a rough-fuzzy based control algorithm is proposed to improve uncertainty reasoning. The navigation behaviour is extracted using entropy-based information measures, and the presented navigation algorithm is evaluated with respect to performance of transient-state adaptability and desired path keeping behaviour.

1.4.2 Thesis Objectives

The thesis has two primary objectives:

1. Identify implementation measures for the performance of an indoor navigation algorithm, including:

(a) Identify and design an indoor environment interaction model (incomplete information system).

(b) Implement the CFBC algorithm.

(c) Experiment with different types of indoor environment to establish the performance criteria for different situations.

2. Design and implement a rough-fuzzy perception-based computing algorithm based on the CFBC, including:

(a) Identify and extract knowledge granules from the CFBC using rough set theory and entropy-based information measures.

(b) Implement the RFBC algorithm.

(c) Experiment with different indoor situations to evaluate performance and improvement of the RFBC algorithm compared with CFBC.

1.4.3 Research Questions

1. What are essential measures of a vision-based indoor navigation task?

(a) What are the characteristics of an indoor environment?

(b) What is the motivation of vision-based navigation?

(c) Why a wall-following behaviour is important to the navigation?

(d) What is the stability performance of current wall-following algorithms?

Comment: An indoor environment is characterized by specific features that are remarkably different from other types of environment. Vision is an efficient sensor as it can mimic human sense of vision and provide richer information of the navigation environment. The wall-following behaviour belongs to a task-oriented behaviour level, which is the highest level of the navigation behaviour. Furthermore, the wall-following behaviour can be derived into different distinct behaviours through variations of a wall. This question enables further research to identify the significance of an efficient wall-following algorithm, and establish performance evaluation methods.

2. Why perception-based computing approach is proposed?

   (a) What are essential properties of current wall-following algorithms?

   (b) What are primary advantages of perception-based computing algorithms?

   Comment: As a matter of fact, a number of wall-following algorithms have been proposed. They can be categorized on the basis of fundamental theoretical approaches. By analyzing of current algorithms, this question provides the reason of presenting perception-based computing. It also addresses the feasibility of perception-based computing for an indoor navigation algorithm.

3. What characterizes the performance of the rough-fuzzy based control algorithm?

   (a) What are the primary evaluation criteria of a perception-based computing algorithm?

   (b) How does the RFBC improve from the CFBC?

(c) How to assess the improvement in terms of uncertainty reasoning?

(d) What is the performance of the RFBC compared with other control algorithms?

Comment: The proposed RFBC is built on the basis of the CFBC. Consequently, the performance evaluation criteria should be consistent with the CFBC. Furthermore, since the RFBC is proposed to improve uncertainty reasoning, a proper evaluation method should be adopted. This question leads to the theoretical reason that why the rough set theory is proposed to filter and reform sensory data before applying fuzzy inference rules.

1.5 Thesis Organization

This thesis presents a perception-based computing for a vision-based indoor navigation robot. A rough-fuzzy theory is proposed to implement the perception-based computing. Ch. 2 presents an overview of the vision-based indoor navigation, and each critical component of the application is elaborated. The key characteristics of an indoor navigation algorithm are identified through analyzing previous related works. Ch. 3 provides a depth study of perception-based computing, and related introduction of fuzzy set and rough set theories. A comparison of fuzzy set and rough set theories is described, and the possibility of theory combination is raised. Furthermore, entropy-based information measures are discussed with respect to the behaviour extraction and evaluation of uncertainty reasoning. Ch. 4 presents a detailed methodology of implementing the perception-based computing algorithm. A rough-fuzzy approach is

fully described using the CFBC as a major comparison. Ch. 5 describes a system implementation design from two aspects, which are hardware and software. Important components are presented and discussed in terms of applicability and reliability. Software includes two primary parts, which are image processing that deals with sensory data and system software that coordinates hardware devices. Ch. 6 focuses on the simulation setup and experiment environment for the testing of the RFBC. The experiment results are depicted in Ch. 7 and conclusions are stated in Ch. 8.

Chapter 2

BACKGROUND ON INDOOR NAVIGATION

Development of a vision-based indoor navigation algorithm involves knowledge in a number of fields. Evaluation criteria can be generated from several critical components. To deeply understand the characteristics of the indoor navigation, this chapter presents background knowledge on these fields so that detailed analysis could be performed. The necessity of an intelligent navigation algorithm is emphasized through this chapter.

2.1 Autonomous Robot

Mobile systems or autonomous vehicles have gained large demand in both military and civilian sectors [LoHa07]. The majority of industrial robotic systems are semi-autonomous and requires guidance of human experts. By contrast, intelligent robotic systems demonstrate advantages in terms of autonomy and efficiency. However, such performance is achieved under certain circumstances due to immaturity. In other words, robustness is critical for intelligent systems development in order to execute tasks successfully against uncertain environments.

Current intelligent systems are promising under distinct conditions, and unquestioningly none of them is a perfect system for universal purpose. Nevertheless, it does not mean that pursuing more sophisticated systems is not worthy. Current research development of intelligent systems such as fuzzy logic, neural networks, genetic algorithms, and symbolic artificial intelligence illustrate great performance against practical environment [LoHa07]. One feasible future development is hybrid system of current intelligent control methods.

2.2 Indoor Navigation

This thesis focuses on indoor navigation tasks using perception-based computing. In general, navigation reference objects (*e.g.*, walls and tables) used in an indoor environment are different compared to reference objects (*e.g.*, road marks and trees) referred in an outdoor environment. Consequently, some techniques for an outdoor environment do not apply to an indoor environment. For instance, Global Positioning System (GPS) may not produce reliable sensory data for indoor navigation, and errors could result in unacceptable consequences. Furthermore, an indoor environment layout is characterized by irregularity, which leads to unpredictability. An indoor environment also imposes compact space, so that the near visual field limits capability of acquiring critical data. For example, an uneven wall makes a mobile robot difficult to follow, and a mobile robot could encounter corners that are less than 90 degrees, which means a robot has to make a large angle turning within limited space.

Noise is usually an unwanted process that contaminates signals. In general, sensory data collected from a practical environment are most likely to be contaminated by noise. As a result, a mobile robot may operate via imprecise data during navigation. For the purpose of maintaining control precision, denoising methods should be adopted according to different types of noise [Kins13]. In practice, it could lead to a situation that a mobile robot needs to make decisions frequently and properly in order to keep desired distance to reference objects.

Robot navigation in a practical environment has the following problems: (i) knowledge of the environment is partial, uncertain, imprecise, and approximate, (ii) the environment and obstacles are dynamic during navigation period, and (iii) collected sensory data are not completely reliable [PaGS05]. This thesis uses an indoor environment (*e.g.*, office and warehouse) for a vision-based autonomous navigation algorithm. An indoor environment demonstrates less common patterns and strictly constrained spaces, which gives rise to considerable increase in the complexity of autonomous vehicle motion coordination, planning and scheduling with the goal to be controlled efficiently and safely. Above mentioned problems can be identified clearly in an indoor environment, and limited knowledge of navigation environment is particularly critical to a navigation algorithm performance. To solve these problems, a mobile robot should have sufficient perception of the environment and the similar decision-making capability as human beings.

2.3 Vision-Based Navigation

Visual navigation for mobile robots has become a source of research contributions since navigation strategies based on vision can broad the scope of autonomous mobile robot applications. Vision servo control refers to the use of computer vision data to control the motion of a robot. Vision is an efficient sensor as it can mimic the human sense of vision and provide non-contact measurement of the navigation environment [HuHC96]. Up to date technologies and researches have raised a number of approaches to achieve the navigation goal in robust fashion. Their efforts cannot be neglected and some research results have been applied in industrial fields [DeKa02]. Nevertheless, research in this area is still far from its end since many key issues still need to be improved.

A common drawback of current autonomous robotics is the navigation sensory data. Popular approaches concentrate on sonar, laser scan, radar and even GPS. These devices are efficient but yet perfect choice since an important element of navigation has been omitted, which is vision. Vision-based navigation is regarded as critical applications in the area of intelligent navigation systems [JiBC06]. For instance, path in the indoor environment may not be created by flat reference objects, such as two chairs close to each other (Fig. 1.1). In this case, a laser sensor is likely to produce data that do not correctly reflect the situation and so does to an ultrasonic sensor. By contrast, vision-based navigation is able to create visual perception that represents the situation that a mobile robot encounters more accurately. This thesis uses visual sensory data as feedback information of the environment status to the controller.

2.3.1 Vision-based Control Techniques

The vision sensory data can be acquired by a camera either mounted directly on a mobile robot or fixed to a stationary platform. This thesis focuses on the former case, which means the motion of a camera induces the motion of a mobile robot. Vision-based control techniques are classified extensively into two categories: *image based visual servoing* (IBVS) and *position based visual servoing* (PBVS) [Wang11]. IBVS was proposed to correct real-time outputs of a mobile robot by estimating errors between current and desired features on the image plane. It does not include any measurement of the pose of the target [Gers09]. IBVS can be used for environment that there is little knowledge about observed scene. By contrast, PBVS estimates the pose of the object of interest with respect to the camera, and then issues corrected outputs [ChHu08]. Image features are also extracted and three dimensional information of the interested object is created.

2.3.2 Visual Landmark Tracking

Landmarks are recognizable natural or man-made feature used for vision-based navigation. Landmarks have one common feature that they can be identified in a scene due to specific characteristics. A mobile robot learns and stores these characteristics while it follows given reference objects. By applying image matching techniques, the mobile robot is able to recognize landmarks and determine its current location. Kim *et al.* [KiLy09] proposed a vision-based navigation algorithm using landmarks. Each landmark is installed on the ceiling along the experimental environment and contains reference

coordinate information. Because error by dead-reckoning localization increases with time, ceiling landmarks are utilized to correct such errors. The deployment of landmarks plays a key role in the navigation algorithm, which implies a very high familiarity degree of the experimental environment.

Landmarks can be categorized into two classes: natural and artificial [JiBC06]. Generally, robust extraction of natural landmarks is much more difficult than artificial ones. However, applications using artificial landmarks require more knowledge about region of interest than natural landmarks. The extraction performance is also influenced by geometrical scene variations for both types of landmarks. In other words, robust and efficient image processing techniques are essential for a visual navigation algorithm using a landmark tracking method.

2.3.3 Visual Target Tracking for Localization and Map Building

Localization could be defined as determining the position of an object within a reference coordinate system. By collecting consecutive localization information, the trajectory of an object can be constructed. It is important that localization and moving object tracking are regarded as the basis for scene understanding, which is the significant difference between adaptive and perceptual system. In the experiment of this thesis, a mobile robot follows and keeps tracking a given wall in an indoor environment. The mobile robot is able to perceive certain perception of the environment by constructing the trajectory of the given wall. Dao *et. al.* [DaYo03] presented a simple linear method for localizing an indoor mobile robot based on a natural landmark. Landmarks are identified

using Canny operator and Lucas-Kanade algorithm. The mobile robot then applies quick localization method based on corresponding line type landmarks. However, incorrect self-localization and tracking outputs can occur since feature recognition performance depends on vision quality, which is influenced by background appearance conditions such as view angle, light, and distance. Furthermore, natural landmarks may be deformed due to specific vision perspectives in the navigation period.

2.4 Wall-Following Behaviour

The indoor environment is considered as a structural space. Furthermore, a wall has the property of ubiquity in such environment, and it is reasonable and efficient to utilize walls as reference coordinates of indoor navigation. Therefore, the wall-following task is considered as an important evaluation method in terms of navigation behaviour of a mobile robot. It is described as moving a mobile robot along a given wall with desired heading angle and while also keep certain distance to the wall. In practice, a mobile robot can hardly follow a wall at a desired distance for a long period due to environmental interference. For example, a mobile robot cannot move forward precisely because of inhomogeneity and slip of wheels. Moreover, the sensory data may be interfered by background noise such as wall surface texture and light source.

In order to following a given wall in the indoor environment, a control algorithm is said to correct its advance path dynamically. By amending inputs for the actual state of the system through feedback functions, a closed feedback loop control strategy minimizes errors between system outputs and desired outputs [Peri05]. As a result, some closed loop

control systems such as proportional-integral-derivative (PID) control, fuzzy logic control were widely adopted as they were easy to be implemented for a highly nonlinear model. Ardiyanto [Ardi10] proposed a PID based control for low-cost mobile robots. The proposed PID controller is designed to (i) reduce reaction time if a mobile robot encounters obstacles, (ii) maintain stable in incomplete or unreliable environment. However, the property of PID based control is adaptive and it manipulates merely with data but not information embedded in the data. For instance, the integral part of PID calculates in the entire time span. This is not always appropriate because error values may be collected in various situations and they should be treated differently. In general, a wall-following task can be subdivided into two distinct situations.

2.4.1 Following an Unknown Wall

An unknown wall is expressed as knowledge about the environment is limited or even none before a mobile robot actually explores it. A robot is assigned with information about starting and ending position and it follows a given wall on its specific side until reaching the ending point. In other words, a wall-following robot in such environment deals with uncertain data and accuracy of outputs (*i.e.,* heading angle and distance to the wall) yet should be maintained. IBVS is usually selected to provide coordinate information based on image processing techniques of continuous images. To fulfill the requirement it needs real-time modelling and certain level perception of the environment. Raguraman *et al.* [RaTS09] showed an indoor navigation algorithm by wall-following and self-localization approach using an infrared sensor. A linguistic fuzzy

logic control is used to avoid obstacles and generate a path to the target. Kim *et al.* [KiLy09] proposed a vision-based navigation algorithm by tracking landmarks installed at the ceiling. A mobile robot corrects errors by calibrating its position with landmarks.

2.4.2 Following a Known Wall

A known wall is characterized by having information about the environment where a mobile robot navigates in. The trajectory of a robot is planned ahead of the wall-following task. The mobile robot corrects its heading angle and distance to the wall by determining its location along the path. Nevertheless, accumulative errors still could occur, thus requires corrections by tracking objects that a mobile robot has fully information about, so that the robot is able to minimize errors by referring desired outputs at a specific position.

Ishikawa [Kawa91] proposed a wall-following scheme by tracking stationary and dynamic obstacles. A planned path was consist of a list of coordinates of reference points and was assigned to a mobile robot. The final navigation path was a series of comprehensive outputs of planned path and obstacle avoidance path. The proposed algorithm has one critical problem that the mobile robot highly relies on coordinates of reference points, the wall-following task may fail if there is a major variation of the environment. Raudonis *et al.* [RaMa11] presented a trajectory following algorithm using vision-based sensor and markers. A camera is mounted above the region of interest and it corrects the robot's direction by analyzing orientation of markers installed on the mobile robot.

2.5 Uncertainty Environment

One of the fundamental problems in an intelligent system is the object classification problem. Especially in situations of pattern recognition and decision making, objects must be collected and refined to be associated to one of predefined sets in order to make reaction to the environment [PaSk94] [LeMB08]. It means information should be extracted through analyzing and understanding of data acquired from sensors [SkWa10] [Hayk05]. In the artificial intelligence (AI) field, each pattern is regarded as atom information for computing. We shall call the atom information as knowledge granule [PeSK08]. Patterns are identified and refined from data collected through interaction with the environment. From a practical perspective, for a particular pattern or object, due to limited knowledge of the environment and scant sensory techniques, complete and precise data collection is very difficult, which infers a knowledge granule is assessed and utilized based on limited data in most instances [LeCJ13]. In this case, uncertainty occurs when patterns are classified into given sets. Furthermore, these predefined sets may also contain uncertainty since they are created in the ideal circumstances. To deal with uncertainty in a human being's fashion is of importance to intelligent control systems.

In general, intelligent systems implemented by well-known AI techniques are mathematical in nature. They generally lack behavioural factors, which is called human-centric design. Conventional computing for such systems is still concentrating on manipulation of numbers and symbols, which is numerical computing [Zade99a], such as PID based control. As mentioned in Sec. 2.2, indoor navigation has the problem that knowledge about the environment is limited and vague. Consequently, patterns created

by a mobile robot have the feature of uncertainty since sensory data are partial and imprecise [LeJH11]. In order to reason accurately, it is necessary to refine and reform sensory data before identifying patterns. In this case, numerical computing or adaptive computing would not achieve satisfied results by imprecise and incomplete data.

The perception-based computing is a sub field of the intelligent computing. Perception is a particular form of interaction of an agent and its environment. Data are acquired from sensors and information is extracted through analyzing and understanding of data [SkWa10]. From an engineering perspective, the perception-based computing is an essential part of feedback mechanism since it perceives critical information from conducted operation data and then makes appropriate adjustments to following actions. Systems built using this theory are considered to be closer to a human-centric design if compared with current intelligent systems [Zade99b] [Kins07] [WaZK10] [GhHa04].

2.6 Kalman Filter for Localization

The Kalman filter is one of the most important and common data fusion algorithms and is named by its major developer Rudolf E. Kalman [Klee07]. It is a recursive estimator for linear systems with Gaussian error statistics. The Kalman filter consists of a set of mathematical equations that estimate the states of a process and minimize the mean square error of the estimated parameters [WeBi06] [Fara12].

Let **x** be a sensory data vector that is generated in real-time while a mobile robot follows a given wall. It contains critical data of a mobile robot at distinct time spot. This

sensory data vector **x** is used throughout this thesis. The Kalman filter model assumes the true state at time *t* is evolved from the state at *t*-1 using a state model

$$\mathbf{x}_t = \mathbf{F}_{t-1}\mathbf{x}_{t-1} + \mathbf{B}_{t-1}\mathbf{u}_{t-1} + \mathbf{q}_{t-1} \tag{2.1}$$

where $\mathbf{x}_t$ is the true state of a process at time *t*, $\mathbf{F}_{t-1}$ is the state-transition model, and $\mathbf{B}_{t-1}$ is the control input model at time *t*-1, $\mathbf{u}_{t-1}$ is the input vector at time *t*-1， and $\mathbf{q}_{t-1}$ is the process noise. It is assumed that the noise is white and with normal probability distributions

$$p(\mathbf{q}) \sim N(0, \mathbf{Q}) \tag{2.2}$$

The **Q** is the process noise covariance and it is assumed to be a constant. The measurement $\mathbf{z}_t$ of a state at time *t* is written as

$$\mathbf{z}_t = \mathbf{Z}_t\mathbf{x}_t + \mathbf{w}_t \tag{2.3}$$

where $\mathbf{Z}_t$ is the measurement model at time *t*. $\mathbf{w}_t$ is the measurement noise, which is also assumed as white and with Gaussian distribution

$$p(\mathbf{w}) \sim N(0, \mathbf{W}) \tag{2.4}$$

where **W** is the measurement noise covariance. Similar with **Q**, it is also assumed to be a constant in order to achieve optimal performance in the Kalman filter.

The Kalman filter estimates states of a process through two phases: prediction and update. The prediction of current state $\hat{\mathbf{x}}_{t|t-1}$ is made based on information at previous state

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_{t-1}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_{t-1}\mathbf{u}_{t-1} \tag{2.5}$$

The estimate of state at time *t* has two distinct representations. $\hat{\mathbf{x}}_{t|t-1}$ is the *a priori* state estimate at time *t* given measurements up to time *t*-1, and $\hat{\mathbf{x}}_{t|t}$ is the *a posteriori* state estimate at time *t* given measurements up to time *t*. In this case, a predicted measurement of current state can be generated using the observation model

$$\hat{\mathbf{z}}_{t|t-1} = \mathbf{Z}_t \hat{\mathbf{x}}_{t|t-1} \tag{2.6}$$

Thus, we have the estimate of measurement $\hat{\mathbf{z}}_{t|t-1}$ at time *t* using information from previous state. Furthermore, an actual measurement $\mathbf{z}_t$ at time *t* is collected through sensors. The difference between predicted measurement and actual measurement is called the measurement residual

$$\bar{\mathbf{z}}_{t|t-1} = \mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1} \tag{2.7}$$

The *a posteriori* estimate of state $\hat{\mathbf{x}}_{t|t}$ is a comprehensive result of prediction $\hat{\mathbf{x}}_{t|t-1}$ and measurement residual $\bar{\mathbf{z}}_{t|t-1}$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \bar{\mathbf{z}}_{t|t-1} \tag{2.8}$$

where $\mathbf{K}_t$ is called the Kalman gain defined in the state covariance estimation at time *t*.

The value of $\mathbf{K}_t$ depends on the *a priori* estimate error covariance $\mathbf{P}_{t|t-1}$ as is denoted by

$$\mathbf{K}_t = \frac{\mathbf{P}_{t|t-1}\mathbf{Z}_t^T}{\mathbf{Z}_t\mathbf{P}_{t|t-1}\mathbf{Z}_t^T + \mathbf{W}} \tag{2.9}$$

As observed, the smaller value of $\mathbf{W}$ means the measurement residual $\mathbf{z}^{-}_{t|t-1}$ is weighted more heavily. If the $\mathbf{P}_{t|t-1}$ approaches to zero, it means the prediction $\hat{\mathbf{x}}_{t|t-1}$ is closer to the estimate of current state $\hat{\mathbf{x}}_{t|t}$. As a fact, the primary goal of Kalman filter is to minimize $\mathbf{P}_{t|t-1}$ so that the prediction model in Eq. (2.5) is more accurate.

In order to apply Kalman filter to the indoor navigation, an initial state estimate $\hat{\mathbf{x}}_{0|0}$ and state error covariance $\mathbf{P}_{0|0}$ should be selected. The initial state error covariance matrix is hard to define, and inappropriate values would let the covariance matrix evolve in a way that decreases the estimation accuracy. Commonly, the starting parameters are chose from experience, which requires sophisticated modeling of the system and assume the noise is Gaussian distribution. From a mathematical point of view, Kalman filter demonstrates its tremendous advantage of control accuracy in a situation that a pre-study of the interaction of an object and its environment is well conducted. Furthermore, the goal of this thesis is to navigation in an indoor environment where knowledge is limited. It implies that a modelling of the environment is imprecise and partial. Though the principle is explicit, the Kalman filter is not easy to formulate and implement to a highly uncertainty environment. Therefore, the Kalman filter is not selected as a comparison algorithm in this thesis.

2.7 Spiking Neural Network for Wall-Following

Neural network is an information processing paradigm that is inspired by the biological nervous systems. It is well known by the novel data handling and processing

structure. A human's nervous system is comprised of a large number of neurons that are highly structured and interconnected [JaMM96]. Each neuron is considered as a simple processing element that processes meta-data and sends outputs to its adjacent neurons. In this case, these neurons work as a union to solve particular problems. Information is extracted from data through structural processing of neurons. Therefore, neural network has gained much interest by its ability of extracting information from complex and imprecise data.

In general, an artificial neural network consists of three layers: (i) the input layer, (ii) the hidden layer, and (iii) the output layer. The input layer accepts data collected from the environment and transfers it to the hidden layer. The detailed activity of the hidden layer depends on the activities of the input layer and weights of different neural connections between the input and hidden layer. Similarly, the results of the output layer are determined by the activities of the hidden layer and weights of neural connections between the hidden and output layer. In other words, the behaviour of the output layer is decided by the way hidden layer extract and reform information from data transferred from the input layer.

Artificial neural networks can be varied and classified into a number of types with respect to application purpose. For the particular case of vision-based indoor navigation, the application contains pattern recognition task which is extremely important to the control accuracy. One of the third generation neural networks named spiking neural network (SNN) is considered to optimize the pattern recognition results for an autonomous robot [WaHo09] [FlEp05]. A spiking neural network has the recurrent

network structure so that it has a loop between the input layer and the output layer. In other words, the control system using the spiking neural network contains feedback to correct errors in real-time tasks.

The spiking neural network consists of three types of neurons, which are sensory neurons, orientation neurons, and the steering neuron. A sensory neuron collects data from the environment and transfers them to the hidden layer. Orientation neurons contain distance limits information that a mobile robot has to take actions against the encountered situation if one of these limits is exceeded. The steering neuron determines the final actual heading angle of a mobile robot. The network is organized as a typical three-layer structure. The input layer is comprised of sensory neurons and orientation neurons. Each neuron sends out a spike if its own threshold is breached. The hidden layer collects spikes from the input layer and calculates the membrane potential. The neurons in the hidden layer send out spikes to the output layer based on calculated results. The steering neuron is the only type of neuron in the output layer and it finally determines the heading angle of a mobile robot at the next time frame.

The network architecture is illustrated in Fig. 2.1. Each sensory neuron value is encoded into spikes by frequency coding in the time window. The number of spikes in each time interval is proportional to the value. For instance, the minimum distance limit for a mobile robot is set to 400mm, and it is encoded as one spike in the corresponding time interval. By contrast, the maximum distance limit 4000mm is encoded as 10 spikes in the time window. The membrane potential of the hidden layer is calculated using temporal coincidence detection code. The hidden layer outputs specific number of spikes

if its membrane potential falls into corresponding threshold interval. The output layer calculates the membrane potential based on the number of spikes from the hidden layer in the time interval. The heading angle adjustment value of a mobile robot is updated if and only if the membrane potential exceeds its resting threshold. Otherwise, the adjustment value keeps the same configuration as previous time interval.

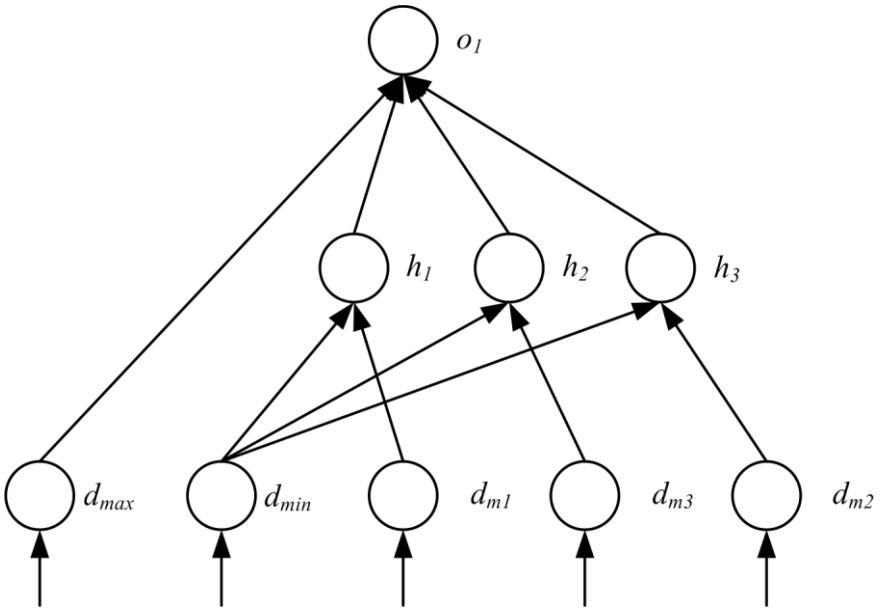


Fig. 2.1: Overview of the spiking neural network.

h_1, h_2 and h_3 calculate membrane potential of each distance limit. Furthermore, h_1 generates spikes that increase membrane potential of neuron o_1 whereas spikes from h_2 decrease the membrane potential. The temporal coincidence coding is applied to determine the number of spikes $S_{ns}(h_1)$ and $S_{ns}(h_2)$ as follows

$$S_{ns}(h_1) = w_{h_1} * \frac{d_{m1}}{d_{min}}$$
$$S_{ns}(h_2) = w_{h_2} * \frac{d_{min}}{d_{m3}} \tag{2.10}$$

where w_{h_1} and w_{h_2} are connection weights of h_1 and h_2. d_{m1} and d_{m3} are the minimum and maximum distance limits that a mobile robot should keep. The membrane potential of h_3 is written as follows

$$MP(h_3) = \frac{d_{min}}{d_{m2}} \tag{2.11}$$

where $MP(h_3)$ is the membrane potential of h_3 and d_{m2} is the desired distance between a mobile robot and a wall. h_3 generates spikes if its membrane potential exceeds the threshold as is denoted by

$$S_{ns}(h_3) = \begin{cases} w_{h3}, & MP(h_3) > MP_{h3\_thr} \\ 0, & \text{otherwise} \end{cases} \tag{2.12}$$

where w_{h3} is the connection weight of h_3, and MP_{h3_thr} is the membrane potential threshold of h_3. The neuron o_1 updates the amount of heading angle adjustment based on incoming spikes and it is denoted by

$$MP(o_1) = \sum_{i=1}^{3} S_{ns}(h_i) \tag{2.13}$$

where $MP(o_1)$ is the membrane potential of o_1. The o_1 fires the activation function if its membrane potential falls in the threshold interval

$$\begin{cases} MP(o_1), & MP_{m1} \le MP(o_1) \le MP_{m2} \\ V_{rest}, & \text{otherwise} \end{cases} \tag{2.14}$$

where MP_{m1} and MP_{m2} are lower and upper bound of its threshold interval, respectively. V_{rest} represents the output neuron is the rest state, which means the heading angle adjustment is not updated in this time interval.

The performance of SNN is tested through simulations against various indoor environments. A mobile robot uses SNN achieves great control results. The mobile robot is able to react to encountered situations through a smooth approaching or avoidance trajectory. This is important in the indoor navigation as stated that stability is one of critical evaluation criteria. However, the convergence rate of SNN is not optimal when comparing with other perceptual algorithms. It implies that the performance would decrease dramatically in a situation that the concentration of layout variations is high enough so that a mobile robot is said not able to keep desired distance during the most of time. The performance of SNN is explicitly discussed in Ch. 7.

2.8 Summary

This chapter elaborates critical components that constitute to a vision-based indoor navigation. The ability of autonomous robots demonstrates promising application prospects. Furthermore, a vision-based approach could be applied to overcome difficulties posed by the indoor navigation environment. In addition, the significance of an efficient wall-following algorithm is emphasized. Two common control algorithms are studied explicitly in terms of advantages and disadvantages. The indoor navigation

problems are then summarized and the perception-based computing is proposed to improve drawbacks of current indoor navigation algorithms.

Chapter 3

PERCEPTION-BASED COMPUTING

3.1 Overview

Perception is defined as the process of organization, interpretation and explanation of acquired data in the brain [ScGW10]. It is a particular form of interaction of an agent and its environment. The literature of perception includes thousands of papers and books in many disciplines such as psychology, linguistics, philosophy, and brain science [Zade99b] [Kins04] [Kins07] [WaZK10] [GhHa04]. With the proposed machine perception, it is possible to improve intelligent control systems and let them step closer to human like [ShNa06]. The proposed approach to robotic perception is being developed to store, analyze and summarize existing experience in given environment so that the machine could grasp the current situation and respond optimally. Perception makes humans be able to communicate with vague, uncertain concepts, and process multiple real-time tasks [SkWa12] [Bela06]. In general, it is possible to summarize the knowledge gained from sensors and implement it to intelligent control applications.

A general perception-based computing could be implemented through a number of approaches. As proposed by James Peters [Pete07], a near-set theory can be used to create perception of situations through identification of similarity among various objects. Near sets are disjoint sets that resemble each other. The similarity of objects is determined by comparing lists of feature values, and disjoint sets are said to be able to resemble each other if there are observable similarities between objects in the sets [Pete09]. Furthermore, similarities represented by a tuple of matching feature values are called descriptions [Pete13]. Near sets approach shows promising performance for pattern identification problems such as image processing and data mining.

The goal of this thesis is to recognize patterns from real-time sensory data rather than creating patterns from database. Therefore, this thesis introduces fuzzy set theory, which is another promising way [Rutk03]. Fuzzy set theory was introduced by Lotfi A. Zadeh in 1965 [Zade99a] as an extension of the classical notion of set. In the classical set theory, membership of elements in a set is binary, it means an element either belongs or does not belong to a set. By contrast, fuzzy set allows degree of assessment of the membership of elements in a set. The value interval for an element's membership is [0, 1]. Therefore, the fuzzy based control is well suited for intelligent systems since it has the capability of classification in uncertain conditions.

3.2 Fuzzy Set Theory

3.2.1 Definitions and Terminology

The fuzzy set theory is a mathematical tool that is used to model the vagueness classification situation of human beings. Let a be a variable which takes values from a universe of discourse U and is denoted by

$$a = u,\ u \in U \tag{3.1}$$

Let F be a fuzzy subset of U and be described with fuzzy membership function μ_F [Zade99]. a is then in the fuzzy restriction of F, and therefore, a also has a value from membership function represents the degree of satisfaction of a to the fuzzy set F. The form can be written as

$$a = u : \mu_F(u) \tag{3.2}$$

Generally, a fuzzy set F is defined by a set of ordered pairs of each variable a in the universe of discourse U associated with F:

$$F = \{(a, \mu_F(u)) \mid u \in U\} \tag{3.3}$$

In brief, let $l(a)$ denotes a fuzzy constriction associated with a. The relationship of F and a can be expressed as

$$l(a) = F \tag{3.4}$$

Furthermore, a is not limited to one value domain, it could be the name of an object, a variable or a proposition, and F is said to have restrictions on values extracted from

features of a. Let $t(a)$ be features of a which takes values from U and the expression is modified as

$$l(t(a)) = F \tag{3.5}$$

For example, let a be a proposition "Tom is young", the expression is induced as

$$\text{Tom is young} \rightarrow l(t(\text{Tom})) = \text{young},\ t = \{\text{age}\},\ l = \{0 \le \text{age} \le 40\} \tag{3.6}$$

The construction of a fuzzy set depends on two aspects: (i) the identification of a suitable universe of discourse and (ii) the specification of an appropriate membership function. They are usually generated by human experts with sufficient knowledge of fuzzy system and the experiment goal. Furthermore, the fuzzy set theory exhibits properties of subjectivity and non-randomness that differs it from the probability theory.

3.2.2 Fuzzy Membership Functions

A fuzzy set is completely parameterized by its membership function. Each object associated with a fuzzy set has membership value thorough membership function (Eq. (3.2)). It indicates the satisfaction degree of an object to the given fuzzy set. In general, the primary types of fuzzy membership functions include triangular, trapezoidal, Gaussian and polynomial. In practice, triangular and Gaussian membership functions are suitable for control algorithms with continuous inputs. A triangular membership function provides powerful uniform control feedback against system changes whereas control signal generated by a Gaussian membership function varies in terms of degrees of system changes.

3.2.2.1 Triangular Membership Function

A triangular membership function is characterized by three parameters $\{m_{i-1}, m_i, m_{i+1}\}$ as follows

$$\mathrm{T}(a; m_{i-1}, m_i, m_{i+1}) = \begin{cases} 0, & a < m_{i-1} \\ \dfrac{a - m_{i-1}}{m_i - m_{i-1}}, & m_{i-1} \le a \le m_i \\ \dfrac{m_{i+1} - a}{m_{i+1} - m_i}, & m_i \le a \le m_{i+1} \\ 0, & m_{i+1} < a \end{cases} \tag{3.7}$$

where parameters $\{m_{i-1}, m_i, m_{i+1}\}$ is in ascending order as shown in Fig. 3.1 and define the X axis of three vertices of a triangular membership function.

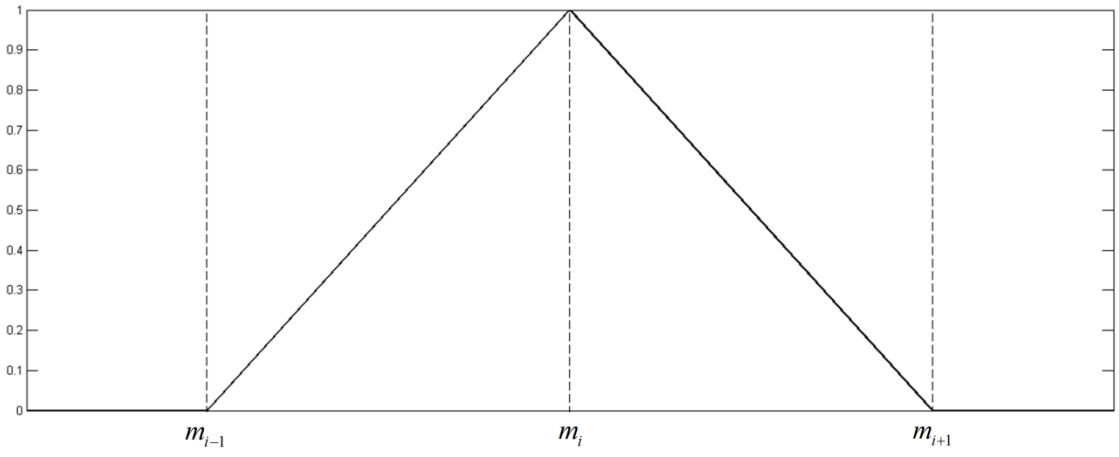


Fig. 3.1: Triangular fuzzy membership function.

3.2.2.2 Gaussian Membership Function

A Gaussian membership function shapes as a bell and it is specified by four parameters $\{m_{i-1}, m_i, m_{i+1}, \sigma\}$. The subset $\{m_{i-1}, m_{i+1}\}$ defines the shape of the bell and m_i is the center. Parameter σ determines flatness of the bell.

$$\mathrm{G}(a; m_{i-1}, m_i, m_{i+1}, \sigma) = e^{-(\frac{a-m_i}{\sigma})^2} \tag{3.8}$$

where parameter σ is written as follows

$$\sigma = \beta(m_{i+1} - m_i) \tag{3.9}$$

where β can be adjusted. The Gaussian membership function is shown in Fig. 3.2.

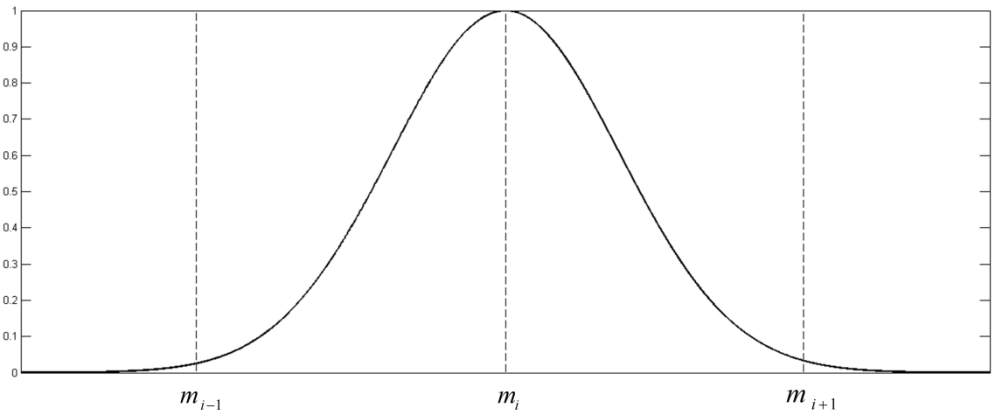


Fig. 3.2: Gaussian fuzzy membership function.

3.2.3 Fuzzy IF-THEN Rules

In the numerical computing, variables are described using numerical numbers or values. In 1973, Lotfi A. Zadeh proposed the concept of linguistic variables. By contrast with numerical variables, linguistic variables are described by words that represent

satisfaction degree to a certain object or requirement. For instance, if the input is temperature for a thermal control, it can be described using words like “high”, “low, “positive”, and “negative”, which is “high/low/positive/negative temperature”. In addition, a variable is able to be completely described by simply using “positive”, “zero”, and “negative” on the value span instead of countless numerical numbers. Similarly, linguistic values are also used to represent output variables.

The rules of a fuzzy inference system can be formulated once the linguistic variables and values range are defined. For a SISO system, an inference relation $input \rightarrow output$ can be interpreted in linguistic terms as

IF *input* is *a* THEN *output* is *b* (3.10)

For example, “IF *temperature* is *high* THEN *evaporation rate* is *high*”. Inputs are called antecedent variables and outputs are called consequent variables. In most instances, inputs *a* and outputs *b* do not belong to the same fuzzy set such that $a \subseteq F_a$ and $b \subseteq F_b$. Eq. (3.10) is defined in a generalized format as follows

IF *a* is $\mu_{F_a}(a)$ THEN *b* is $\mu_{F_b}(b)$ (3.11)

3.2.4 Fuzzy Membership Functions with Overlap

If data acquired from the environment are regarded as an input, the property of fuzzy set determines that the input is able to have satisfaction degree values in more than one related fuzzy sets. Therefore, a fuzzy control system can have multiple fuzzy sets outputs

for a single input. It implies that an input a can have multiple fuzzy inference rules such that

$$\begin{aligned} &\text{IF } a \text{ is } \mu_{F_{a1}}(a) \text{ THEN } b \text{ is } \mu_{F_{b1}}(b) \\ &\text{IF } a \text{ is } \mu_{F_{a2}}(a) \text{ THEN } b \text{ is } \mu_{F_{b2}}(b) \end{aligned} \tag{3.12}$$

Obviously, these fuzzy sets are said to be mutually overlapped in terms of the input a. In this case, the system output corresponding to the input a is a composition value of multiple fuzzy membership functions. If consider a fuzzy control system with continuous inputs in the time domain, in order to keep entire system working smoothly, it may be necessary to equally overlap fuzzy membership functions so that a series composite outputs would not show dramatically change [EsVa04]. Equally overlapped membership functions demonstrate two important properties.

1. The overlap is 1/2.

2. $\mu_{F_{a2}}(a) = 1 - \mu_{F_{a1}}(a)$ The values of a variable a in two overlapping fuzzy sets are summed up to 1. This feature is extremely important for the system stability.

A composite output value of an input a can be written as follows

$$f(a) = \frac{\mu_{F_{b1}}(b)\mu_{F_{a1}}(a) + \mu_{F_{b2}}(b)\mu_{F_{a2}}(a)}{\mu_{F_{a1}}(a) + \mu_{F_{a2}}(a)} \tag{3.13}$$

For the case of Gaussian fuzzy membership functions with equally overlapping as shown in Fig. 3.3, there are three fuzzy sets centered at $m_{i-1}, m_i,$ and m_{i+1}, respectively. Every two consecutive fuzzy membership functions are equally overlapped. In this case, the composite output of an input a in the interval of $[m_{i-1}, m_i]$ is written as

$$f(a) = \frac{y_{m_{i-1}} \mu_{m_{i-1}}(a) + y_{m_i} \mu_{m_i}(a)}{\mu_{m_{i-1}}(a) + \mu_{m_i}(a)} \tag{3.14}$$

where $y_{m_{i-1}}$ and y_{m_i} are fuzzy outputs of Gaussian membership functions centered at m_{i-1} and m_i, respectively. $\mu_{m_{i-1}}(a)$ and $\mu_{m_i}(a)$ are corresponding Gaussian membership values of the input a as in Eq. (3.8).

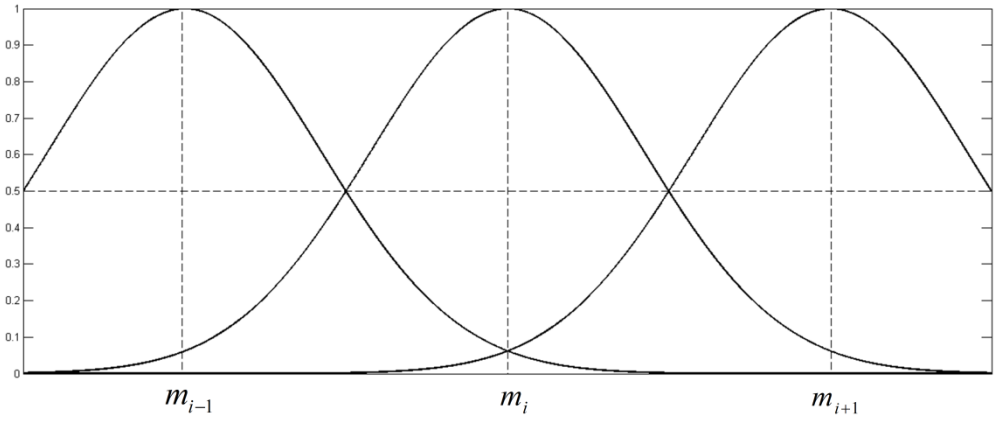


Fig. 3.3: Gaussian fuzzy membership function with 1/2 overlap.

3.3 Fuzzy Inference System

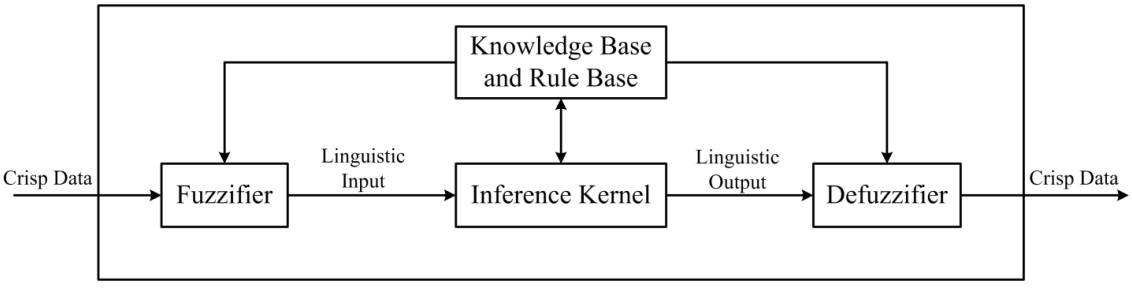


Fig. 3.4: Overview of a fuzzy inference system.

As shown in Fig. 3.4, a fuzzy inference system contains four parts: fuzzifier, knowledge base and rule base, inference kernel, and defuzzifier. A fuzzifier transforms sensory data from crisp measured signals into fuzzy quantities. The transformation is performed using antecedent membership functions stored in the knowledge base. In practice, the amount and type of fuzzy membership functions are determined by human experts with adequate knowledge of fuzzy inference system and the experiment goal. The rule base consists of inference rules (*i.e.*, IF-THEN rules) that specify actions to linguistic inputs. An inference kernel is the critical part of a fuzzy inference system, it simulates human decision making based on non-crisp concepts and fuzzy logic rules. The inference kernel evaluates linguistic inputs via rule base (the set of IF-THEN rules) and reasons linguistic outputs. The linguistic outputs are then transformed into crisp data through the defuzzifier. Consequent membership functions are used to generate crisp outputs in order to control a mobile robot using numerical signals.

3.4 Previous Work Based on Fuzzy Set Theory

Raudonis *et al.* [RaMa11] proposed a trajectory following algorithm using fuzzy based control. A camera is mounted above the region of interest and it corrects a two wheels robot's direction by analyzing orientation of markers installed on the mobile robot. Though fuzzy based control performs great results in terms of trajectory following, the mobile robot's path is not optimized. For a desired indoor navigation algorithm, a mobile robot's path should be smooth, which means the minimization of vibration amplitude.

In general, the optimization for a fuzzy based control system can be achieved through two primary methods. As shown in Fig. 3.4, a result quality depends on two major aspects: the accuracy of crisp inputs, and the precision of knowledge and rule base. To improve the precision of knowledge and rule base requires tremendous pre-study about the navigation environment. It may decrease the overall efficiency of a navigation algorithm, and narrow the algorithm's application range. Therefore, common solutions more concentrate on enhancement of sensory data accuracy. Li *et al.* [LiCF08] proposed an obstacle avoidance algorithm by tracking multiple stationary objects using ultrasonic sensor. The algorithm aims to use multiple sensory data to compensate the lack of sensory accuracy. Another similar approach is to apply multiple sensory devices. Amanatiadis *et al.* [AmCh10] presented a fuzzy based control system using multiple sensors. To achieve accurate sensory data, the proposed algorithm integrates data from an inertial sensor, a digital camera and a radio frequency identification device using a sophisticated fuzzy logic system. These solutions improved the fuzzy based control accuracy. However, in order to further optimize the performance, it is more appropriate to utilize additional filter algorithms.

A filter method should correct imprecise data from logical point of view. It should be able to identify and minimize errors from a series of sensory data. Budiharto *et al.* [BuJa10] proposed a development of adaptive neural fuzzy inference system controller for humanoid servant robot designed for a vision-based robot. A servant robot is designed to serve customers following line landmarks. A digital camera and OpenCV image processing techniques are used to recognize human faces. A fuzzy based navigation

system is formed and optimized based on composition information of landmarks and recognition status. Karambakhsh *et al.* [KaKh11] presented a fuzzy Kalman filter control algorithm to optimize navigation algorithms in terms of the indoor environment. To explore unknown environment, a sonar sensor and a laser sensor are applied to create a virtual model where a mobile robot currently is. The Kalman static filter is used to smooth the sensory input of a fuzzy based control system. Both two methods demonstrate great performance in terms of data filtering. However, the data processing efficiency is not optimized, which implies considerable time consumption for a data filtering task (See Sec. 2.6). The goal of an additional data filtering method is to regulate input data, as well as keep the efficiency of a navigation algorithm. A rough set theory is then proposed for this reason.

3.5 Rough Set Theory

The rough set theory proposed by Zdzislaw Pawlak is seen as a new mathematical tool to deal with imprecision, vagueness and uncertainty [Pawl97]. In this work, the usual information table-based attributes from rough set theory are replaced by sensor-based features. Objects of interest are characterized by features and each feature is associated with a sensor. Elements with the same feature values belong to the same equivalence class. Such elements cannot be distinguished. In other words, a knowledge granule [JaSk08] is formed from such class since each element in the specific class exhibits the same feature values, which is closer to the human way of analysis than the numerical computing. This is extremely important since human centric design is the trend for the AI development,

and knowledge granule is one indispensable property of human-like thinking behaviour [PePS02]. Therefore, the rough approach is essentiality of fundamental importance to AI and cognitive sciences, such as decision analysis, machine learning and pattern recognition [Pawl97].

In general, there are two types of sets in terms of features. A set that its members have completely the same features in terms of type, amount and value called crisp set. Otherwise, a rough set is then formed [Pawl97] [Pawl82]. In the real world, a rough set is common whereas crisp set is limited; it is because the knowledge to the most of existing objects is limited. The vagueness comes from incomplete information about objects we are interested. For instance, people love to buy drinks in summer, if people with drink are grouped as a set, the details of drink such as brand, type and total capacity can lead to further classification. However, information may not be completely available if we require deeper data such as residual volume, temperature, and even people's feeling. In this form, in order to classify objects into distinct sets properly, each set should be represented using knowledge granule that is extracted and reformed from essential features of the set.

3.5.1 Definition and Terminology

Two crisp sets are introduced to describe a rough set, called the upper approximation and lower approximation space [Pawl97]. The lower approximation set represents concepts or elements that certainly belong to the rough set and the upper approximation set contains all elements that surely or possibly belong to the rough set. The difference

between the lower and upper approximation space is called the boundary region of the rough set.

Let U be a certain set called the universe. R is the equivalence relation on U. Thus we have a pair $O=(U,R)$ called an approximation space. By applying R, it is said that elements in O are indiscernible. Furthermore, classes created by R are called elementary sets or knowledge granules in O [PeSK08], and the set of all such knowledge granules in O is denoted by U/R. Every finite union of knowledge granules in O called a composed set. Let M be an arbitrary reference set $M \subseteq U$ and a is an arbitrary element in U. The set M can be described by a pair of its lower and upper approximations as follows

$$\begin{aligned} \underline{R}(M) &= \{a \mid a \in U, [a]_R \subseteq M\} \\ \overline{R}(M) &= \{a \mid a \in U, [a]_R \bigcap M \neq \phi\} \end{aligned} \tag{3.15}$$

where $\underline{R}(M)$ and $\overline{R}(M)$ are lower and upper approximation spaces of M, respectively. $[a]_R$ is an equivalence class in terms of R. It also stands for a knowledge granule created by R. A knowledge granule is the basic information atom that represents particular feature independently. $\underline{R}(M)$ is used to describe all features that the reference set M certainly has and $\overline{R}(M)$ represents all features that M possibly has. The boundary region of M is the set $BN_R(M)$ [SaYe98]

$$BN_R(M) = \overline{R}(M) - \underline{R}(M) \tag{3.16}$$

As shown in Fig. 3.5, a closed shape formed by a random line is the arbitrary set M. The universe U is the union of negative region (*i.e.*, elements in U that surely do not belong to M), boundary region and lower approximation. The upper approximation is

comprised of boundary region and lower approximation. It can be noticed that if $BN_R(M) = \phi$, the set M becomes a crisp set in terms of *R*. Otherwise it is a rough set in terms of *R*.

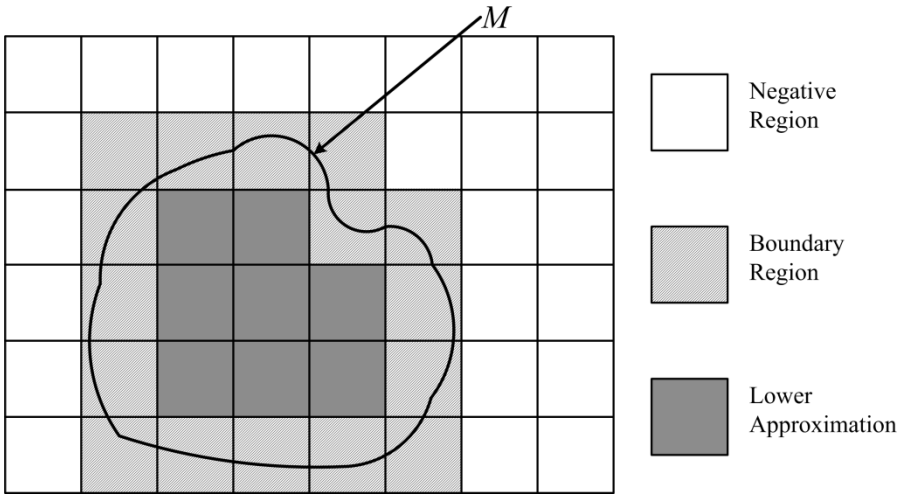


Fig. 3.5: Overview of a rough set.

The vagueness degree of a rough set *M* is evaluated by

$$\alpha_R(M) = \frac{|\underline{R}(M)|}{|\overline{R}(M)|} \tag{3.17}$$

where $|M|$ represents the cardinality of the set M. It is intuitive that $0 \le \alpha_R(M) \le 1$, and M is crisp in terms of *R* if $\alpha_R(M) = 1$. This equation can also be used for verification of accuracy of approximation of the set M.

3.5.2 Rough Membership Functions

The rough membership function of a rough set *M* given equivalence relation rule *R* is defined by

$$\lambda_M^R(a) = \frac{|[a]_R \bigcap M|}{|[a]_R|} \tag{3.18}$$

where $\lambda_M^R(a)$ is the membership value of an object a to the rough set M in terms of equivalence rule base R. $[a]_R$ is the equivalence class of a based on R as shown in Fig. 3.6. The value range is $0 \le \lambda_M^R(a) \le 1$. An object certainly does not belong to M if $\lambda_M^R(a) = 0$, which means $a \in U - \bar{R}(M)$. It is inferred from the equation that elements in the same elementary set $[a]_R$ are equally regarded as a knowledge granule and they have the same membership value through a rough membership function. The degree of an object's belongingness to M is the complementary of the degree of this object's belongingness to $U - M$ in terms of total degree of 1.

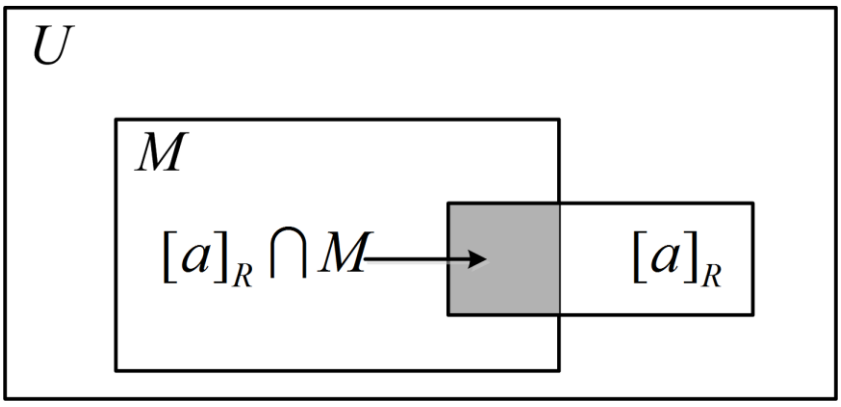


Fig. 3.6: Overview of rough membership function.

A composed rough set consists of multiple rough sets and the membership value of an object to a composed rough set is calculated as [PaSk94]

$$\begin{aligned} \lambda_{M \cup J}^R(a) &= \max(\lambda_M^R(a), \lambda_J^R(a)), a \in U \\ \lambda_{M \cap J}^R(a) &= \min(\lambda_M^R(a), \lambda_J^R(a)), a \in U \end{aligned} \tag{3.19}$$

where M and J are rough sets defined by R. $\lambda^{R}_{M\cup J}(a)$ and $\lambda^{R}_{M\cap J}(a)$ are membership functions of the union of M and J, the intersection of M and J, respectively.

3.6 Comparison of Rough Set and Fuzzy Set

In a set theory, in addition to the various approaches of manipulations and principles, the basis of a set consists of rules and elements. The rules, or is named characteristics, is used to determine the belongingness of an element to a given set. In general, rules are summarized mathematically or logically, and each element is able to be classified according to specific characteristics. For the case of non-crisp set theories (*e.g.*, fuzzy set and rough set), the most significant property that differs them from crisp set theories is the expression of belongingness. The belongingness of an element to a given set is determined by the degree of matching between attributes of the element and rules of the set. It is observed that the entity of a set depends on rules of the given set and attributes of elements. In other words, in order to define a set, a fixed rule base should be pre-defined. Furthermore, in order to define belongingness of an element, attributes of the element should be extracted.

The rough set theory deals with incomplete information, and it aims to extract essential attributes to classify elements into indiscernible oriented sets with limited starting knowledge base. The unique property of a rough set is the extraction of attributes in order to enrich knowledge base. Consequently, more precise membership value of a set can be calculated and elements are then further classified by induced knowledge base. In

applications, the rough set can be used for data refining and feature extraction, the results are treated as reformed input for a decision making system.

The fuzzy set theory deals with crisp knowledge base, it means the rule base of a fuzzy set is pre-defined. Whether an element belongs to a fuzzy set depends on its attributes and rules of that fuzzy set. A fuzzy set theory can be used for decision making purpose in applications.

As shown in Table 3.1, a SISO system is considered as an example [SaYe98]. The comparison demonstrates the major difference of a rough set and a fuzzy set from both aspects of input and output.

Table 3.1: Comparison of a fuzzy set and a rough set.

	Input	Output
Rough Set	Rule base of a set is determined upon equivalence relations are applied to candidate elements.	Output is a value processed by post-defined rule base of the set.
Fuzzy Set	Candidate elements are classified according to pre-defined rule base of a set.	Output is a comprehensive value of rule base of the set and membership values of elements.

In general, an input is sensory data captured from the environment; it is then processed and classified into output sets. Since input data can be plentiful and inaccurate, they may not be exactly matched to fixed output sets, it is appropriate to name such processing as approximation. On the one hand, if classification of data is not properly performed, then the roughness appears. On the other hand, the difference between an input and classified output sets cannot be ignored, so the fuzziness exists. Thus for an

intelligent control system, we need a method that can integrate roughness and fuzziness into one systematic function. This is essential for this thesis since in the vision-based indoor navigation algorithm, sensory data including distance, speed and heading angle vary continuously. In the meanwhile, reactions to a prospective situation are limited to turning, forwarding, and stop. In this case, an accurate and efficient classification method is critical to the proposed control method.

When compared with the fuzzy set theory, the rules of a set created by rough set theory are adjusted according to features of sensory inputs. In other words, the rule base formed by rough set theory reflects the situation of input data from a practical point of view. As a matter of fact, compared with particular information collection, environment shows much more diversity in the aspects of data. Such comparison is consolidate not only regarding data type and amount, but also related to potential combination and structure variation. Therefore, a classification is conducted directly by a pre-defined rule base may not be a good choice since knowledge base is so limited and mismatch may occur. For instance, if an autonomous robot encounters an obstacle right in the front, to turn right or turn left appear to share equal possibility. It is necessary for a mobile robot to acquire more information rather than randomly picks one direction.

As introduced in Sec. 3.5, each equivalent set represents a knowledge granule. An equivalent set with higher percentage of dominant objects means more solid information it expresses, and of course less vagueness. In an ideal case, the output of a rough set can be seen as a universal result of every element in this rough set. In other words, elements in a rough set are regarded as equal in terms of a specific rule base. Furthermore, fuzzy

set theory is able to synthesis multiple reasoning results into one ultimate reaction to the environment. As a result, if we treat the output of a rough set as a composed element, it is possible to use the composed candidate element as an input of the fuzzy set, which may produce more appropriate classification and reasoning results. The optimality of RFBC over CFBC is observed through comparisons of both simulations and experiments with respect to energy-based metrics and information-based measures, respectively. Therefore, this thesis proposes the rough-fuzzy approach for the implementation of the perception-based computing.

3.7 Information Measures

The RFBC proposed in this thesis aims to improve reasoning results for a vision-based indoor navigation in the uncertainty environment, as stated in Sec. 2.5. Consequently, evaluations of algorithm robustness towards uncertainty conditions should be conducted in a proper way. In the thesis, when a mobile robot follows a wall, a real-time sensory input data vector **x** (see Sec. 2.6) contains measured distance between a mobile robot and a given wall (*i.e.*, feedback of interaction of the robot and the wall), and a series of such data form a trajectory of this mobile robot. Traditional energy-based metrics could evaluate the trajectory with respect to numerical performance. It reflects the central value of measured distance and how close data are to the mean. However, the proposed perception-based computing is designed to grasp information from the environment and respond optimally. In order to further analyze the information

quantification performance against the uncertainty environment, methods that evaluate the information that embedded in control results should be adopted [Kins04] [Wang02].

3.7.1 Shannon's Entropy

The information theory developed by Ralph Hartley [Hart28] and Claude E. Shannon [Shan48] is a mathematical theory of communications on the basis of probability theory [XiBe84]. The importance of information theory has made it a key theory in both research and application fields. In information theory, entropy is a measure that is used to quantify the expected value of the information contained in a message. Furthermore, Shannon entropy is also a quantitative measure of the uncertainty of a random variable [DiSJ06]. A random variable with less uncertainty of potential values contains lower entropy (*i.e.*, less information) than a variable with more uncertainty. Furthermore, the number of all potential values of a random variable also has an impact on its entropy [Titc00]. In other words, the amount of information depends on the uncertainty of the variable. For instance, if a variable is likely to be a specific value with probability of 0.9, the amount of information is less since the probability distribution of values is relatively concentrated.

In this thesis, the value of input data vector **x** could be chosen from a set of all possible values. In general, the probability of each potential value to be assigned varies along with different navigation algorithms. In an ideal case, the range should be narrowed to around a desired value. In other words, the value of next measured distance is most likely to be in a specific interval. That is to say, though the uncertainty (*i.e.*, information)

of the environment is high, the entropy of control results (*e.g,* information of measured distance) is low, which means the algorithm performance is stable against uncertainty situations.

In general, a set of all possible values of an input data vector $\mathbf{x}$ is denoted by a bag of N potential values $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. In order to describe the probability that $\mathbf{x}$ is to be a particular potential value, a *probability mass function* (pmf) is used. A pmf characterize the probability that a discrete variable is exactly equal to some values. By contrast, a *probability density function* (pdf) is used to define the probability distribution of a continuous variable. Since the $\mathbf{x}$ is discrete, the pmf is used throughout the navigation period as denoted by

$$\mathrm{P} = [\, p_{\mathbf{x}_1}(\mathbf{x}_1),\ p_{\mathbf{x}_2}(\mathbf{x}_2), ...,\ p_{\mathbf{x}_n}(\mathbf{x}_n)] \tag{3.20}$$

where P is the pmf, and notice that the probability of each potential value could be the same, such as $p_{\mathbf{x}_1}(\mathbf{x}_1) = p_{\mathbf{x}_2}(\mathbf{x}_2)$. The trajectory formed from the measured distance value in $\mathbf{x}$ is denoted by $\mathbf{J} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m]$. The number of measures in $\mathbf{J}$ is denoted by M. Notice that the bag notation $[\bullet]$ allows equality of elements, while the bag notation $\{\bullet\}$ precludes the equality of elements. According to Shannon's information theory, the self-information $I(\mathbf{x}_i)$ of each possible value is defined as

$$I(\mathbf{x}_i) = -\log_c p_{\mathbf{x}_i}(\mathbf{x}_i),\ i \in \mathrm{N} \tag{3.21}$$

where c is the size of the coding alphabet [Kins04]. For a binary coding alphabet $\{0,1\}$, the size is $c = 2$, and the unit of entropy is bit. N is the size of a bag of all potential

values. Therefore, the entropy $H(\mathbf{J})$ of the trajectory is a weighted sum of self-information in the $\mathbf{J}$ as denoted by

$$H(\mathbf{J}) = -\sum_{i=1}^{N} p_{\mathbf{x}_i} \log_c p_{\mathbf{x}_i}(\mathbf{x}_i) \tag{3.22}$$

The entropy reflects the amount of new information generated by the trajectory, and the goal of this thesis is to minimize such new information so that the measured distance would stay close to the desired distance. In other words, the uncertainty of control outputs should be minimized so that they are most likely to be in an expected value interval. The uncertainty reasoning quality of this thesis is analyzed through several common situations in the indoor environment, such as transient-state at different types of wall, path keeping, and emergency avoidance behaviour.

As shown in Fig. 3.7, the illustrated curve is one of experiment results conducted in this thesis. What we can infer is that the trajectory is around the desired path, and a Shannon's entropy measure is able to represent the overall uncertainty reasoning quality. However, an entropy measure for the entire trajectory does not reflect the path keeping accuracy. In other words, a lower value of entropy measure does not mean the expected value interval would be around the desired path. In this case, a conditional entropy measure should be used [Kins04].

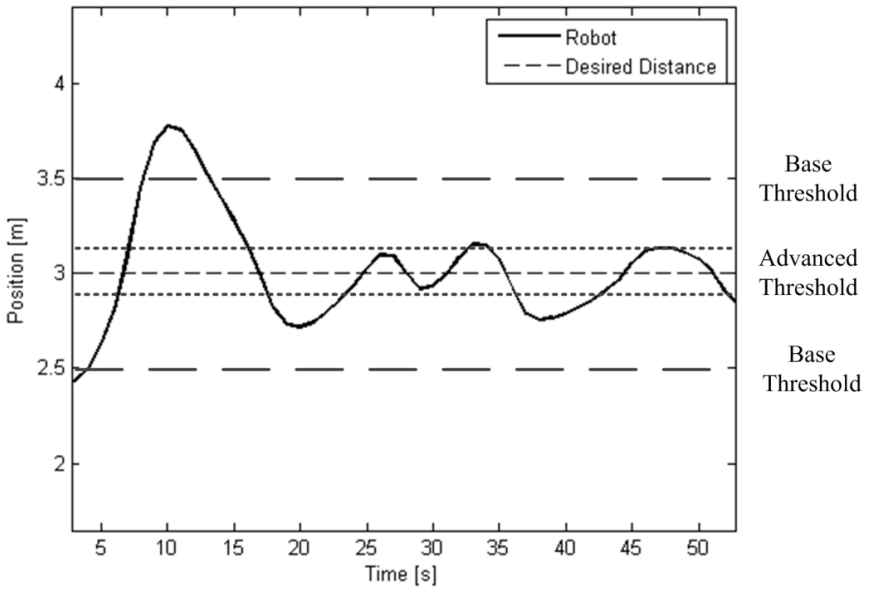


Fig. 3.7: The base threshold and advanced threshold.

A conditional entropy measure is used to further assess the uncertainty of a variable or an event given certain restrictions. If a base threshold of a trajectory is set and values within the threshold is considered as good control results along the trajectory, which means an algorithm performance is said to be good if most measures of its trajectory lie in the threshold interval. Moreover, another threshold named as advanced threshold is set and values within this threshold is considered as excellent control results. The relationship between these two thresholds is shown in, and.

The performance of all algorithms stated in this thesis meet the requirement of base threshold. However, the advanced threshold leads to different navigation quality. The conditional entropy is the uncertainty measure of the advanced threshold given the base threshold. Let *ThB* be an event that the result at a specific time spot is within the base threshold, and *ThA* is the event that this result falls in the advanced threshold interval.

Then the conditional probability of an excellent performance given the base threshold is denoted by $p(ThA|ThB)$. The conditional entropy is written as

$$H(ThA|ThB) = -p(ThB)\cdot p(ThA|ThB)\cdot \log_c p(ThA|ThB) \tag{3.23}$$

where the operation "$\cdot$" is a algebraic multiplication, and c is the same as in Eq. (3.21). The probability of each event is induced from the pmf of Eq. (3.20). Since each event is binary (*i.e.*, either happen or not happen), the above conditional entropy equation removes the notation of summation. The combination of Shannon's information entropy and conditional entropy can be used to evaluate the stability and control accuracy of control outputs of navigation algorithms.

3.7.2 Rényi's Entropy

Rényi's entropy is a generalized form of Shannon's entropy that could discern the spread of probabilities in the pmf [FaKi05]. The Rényi's entropy is written as

$$H_q(\mathbf{J}) = \frac{1}{1-q}\log_c \sum_i^{\mathrm{N}} p_{\mathbf{x}_i}^q(\mathbf{x}_i),\ -\infty \le q \le +\infty \tag{3.24}$$

where q is the moment order [Kins05]. For $q=1$, the Rényi's entropy is known as the Shannon's information entropy. Using different values of q, different entropy characteristics are revealed from each pmf P_q. A higher order of q can be understood as the probability that q measures to be the same potential value. Entropy-based fractal dimensions deals with non-uniform distributions in the fractals. A Rényi's entropy-based fractal dimension could be raised to multi-scale analysis of a trajectory $\mathbf{J}$. In other words,

the nature of **J** can be extracted through analysis at difference scales. The Rényi's dimension is written as

$$D_q = \lim_{k \to \infty} \frac{H_q}{\log(1/s_k)} \tag{3.25}$$

where s_k is the size of a potential value interval. The size of s_k should be limited since scales beyond the range may not produce reliable results [FaKi05]. In this thesis, the size of s_k is calculated using $s_k = 2^{-k}$ meter, and $k = 0,1,2,\ldots$. As shown in Fig. 3.8, a Rényi's dimension spectrum of trajectory **J** in Fig. 3.7 with $-10 \le q \le 10$ and $k = 4$ is a monotonically decreasing [Kins05] [KiDa06].

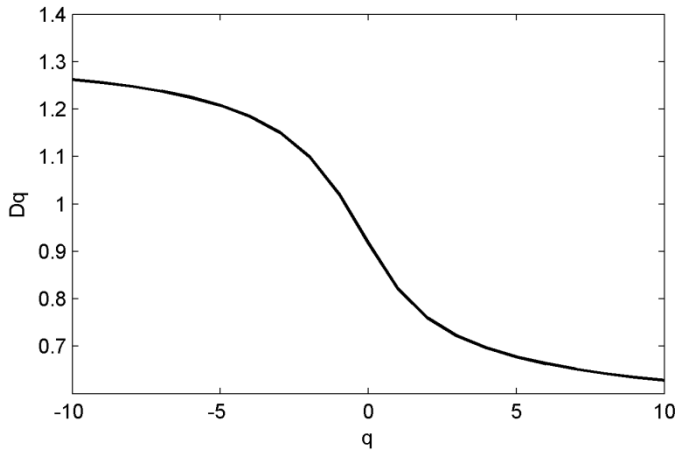


Fig. 3.8: The Rényi's dimension spectrum.

The significance of the Rényi's dimension spectrum is that it reveals the nature of the trajectory, and it is a bounded signature of the trajectory. Multiple "S" curves in Fig. 3.8 using the same algorithm would converge to an identical pattern, and this proposition is more solid when the number of measures M in a trajectory **J** is larger than a specific threshold. In other words, the pattern of a trajectory could be identified and the behaviour

analysis of an algorithm is feasible. This is extremely important since denoising techniques can be applied to separate environmental interference from desired sensory data. In practice, the threshold value depends on the particular properties of an algorithm and the indoor environment [KiSo09]. For instance, the trajectory **J** in Fig. 3.7 tends to converge while the number of measures increases as illustrated in Fig. 3.9. In this case, the curve is stable when the M is larger than 750.

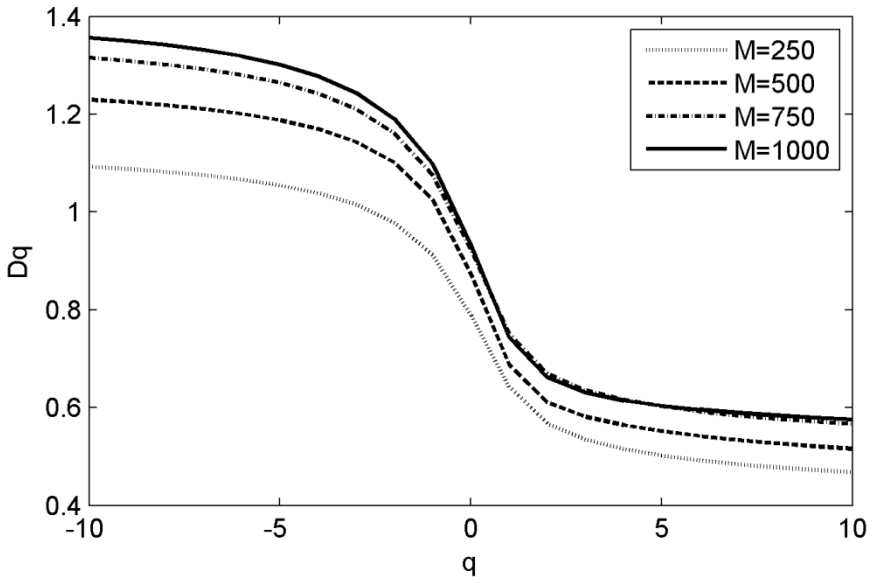


Fig. 3.9: The Rényi's dimension spectrum with distinct values of N.

A Shannon's information entropy is a single-valued measure that indicates the uncertainty of a navigation trajectory. By contrast, the Rényi's dimension spectrum is able to exhibit the nature of an algorithm's control behaviour through multi-scale analysis. Therefore, in this thesis, the Rényi's entropy can be used to evaluate the performance of an algorithm via analysis of its Rényi's dimension spectrum.

3.7.3 Learning Entropy

The Shannon's entropy demonstrates its advantages over uncertainty evaluation of overall trajectory performance. However, it does not consider consistency of data with the system behaviour. The Rényi's entropy is used to identify the behaviour pattern of an algorithm. In order to further assess the consistency of each measured input data vector $\mathbf{x}$ to the trajectory, a *sample-by-sample* (SBS) entropy-based monitoring method named *Learning Entropy* (LE) is introduced to measure the new information embedded in a specific sensory data [BuKo13]. More precisely, since measured data may contain errors, the influence of noise to predicted value (*i.e.*, the amount of new information in the sensory data) is analyzed through the consistency evaluation.

In an ideal case, the new information of an arbitrary sensory data should be the minimum. In other words, it is expected that the measured data match the control algorithm results. In practice, the difference between predicted data and measured data is not foreseen by the control model. Therefore, this thesis integrates a rough set theory to regulate the sensory data. For a particular sample that is perturbed by environmental interference, the performance of redressing data should be quantified so that an intuitive understanding of damage is obtained. The inconsistency of measured data to the system behaviour is incrementally learned with respect to the temporary system governing law. In this case, it facilitates the adjustment of equivalence relation rule, and a better behaviour extraction can be conducted. By real-time adaptation of a short-term predictor and by observing the behaviour of adapted parameters, we are able to cognitively monitor and evaluate every new measured sample or even whole intervals of behaviour with

varying complexity. The proposed LE does not operate with residuals that are adopted in some learning algorithms. Furthermore, the LE does not require accurate prediction model since the inconsistency is mainly evaluated through multi-scale analysis of consecutive sensory data series.

In this thesis, a mobile robot continuously collects data while it navigates in an unknown indoor environment. A navigation algorithm generates outputs by given sensory data, and the prediction model estimates the heading angle and position based on outputs of the navigation algorithm. Let $\tilde{\mathbf{x}}_{t+n}$ be the predicted value at time $t+n$ using sensory data collected at time $t+n$-1, and at time $t+n$ the corresponding measured data are $\mathbf{x}_{t+n}$. Therefore, the difference between two values is

$$\mathbf{e}_{t+n} = \mathbf{x}_{t+n} - \tilde{\mathbf{x}}_{t+n} \tag{3.26}$$

For short-term navigation duration, a partial trajectory $\mathbf{J}_{t,t+n}$ in the time interval $[t, t+n]$ can be formed as

$$\mathbf{J}_{t,t+n} = [\mathbf{x}_t, \mathbf{x}_{t+1}, ..., \mathbf{x}_{t+n}]^T \tag{3.27}$$

where $[\bullet]$ is the same notation as Eq. (3.20), and $[\bullet]^T$ represents the vector transposition operation. A quadratic multiplication $\mathbf{C}$ is conducted from the $\mathbf{J}_{t,t+n}$ as denoted by

$$\mathbf{C}_{t,t+n} = [\mathbf{x}_i \mathbf{x}_j ; i = t, ..., t+n,\ j = i, ..., t+n]^T \tag{3.28}$$

In this case, we can have a vector of weights $\Delta\mathbf{w}$ that stand for parameter increments of the predictor for each element in $\mathbf{C}$ using the following equation

$$\Delta \mathbf{w}_{t+n} = \frac{\tau}{1+\mathbf{C}^T \cdot \mathbf{C}} \cdot \mathbf{e}_{t+n} \cdot \mathbf{C}^T \tag{3.29}$$

where τ is the learning rate, and the weight vector reflects the relationship of the measured data $\mathbf{x}_{t+n}$ and its former *n* measured data. The weight vector is compared with the recent average weight vector $\overline{\Delta \mathbf{w}}$. If a weight exceeds its corresponding average value, it is called a unusual magnitude in $\Delta \mathbf{w}$. The total number of such unusual magnitudes in a weight vector determines the degree that a measured data $\mathbf{x}_{t+n}$ differ from its former *n* measured data.

$$\left|\Delta \mathbf{w}_{t+n}\right| > \gamma \cdot \left|\overline{\Delta \mathbf{w}}\right| \tag{3.30}$$

where γ is the detection sensitivity parameter. We can see that the value of γ is critical when determining the amount of new information in the measured data. As a result, the learning entropy $L(\mathbf{x}_{t+n})$ of a measured data at time *t+n* is approximated as

$$L(\mathbf{x}_{t+n}) = \frac{1}{n \cdot n_\gamma} \sum_{i=1}^{n_\gamma} N(\boldsymbol{\gamma}_i), \ \ \boldsymbol{\gamma} = \{\gamma_1, \gamma_2, ..., \gamma_{n_\gamma}\} \tag{3.31}$$

where n_γ is the number of a set $\boldsymbol{\gamma}$ that consists of detection sensitivity parameters, and $N(\bullet)$ is the number of unusual magnitudes in $\Delta \mathbf{w}$ when applying a specific sensitivity value. For the purpose of better approximation, elements in $\boldsymbol{\gamma}$ should be around the detection sensitivity value that make the first unusual magnitude appear.

By analyzing above equations, the LE of a newly measured data depends on two primary parameters, which are the number of former measured data *n* and the detection sensitivity value γ. In general, these *n* measured data should fully reveal the nature of the

partial trajectory, otherwise misjudgement may happen. However, it is difficult to achieve the appropriate value of *n* in an environment where knowledge is limited. Nevertheless, the value of *n* could be approximated through multi-scale analysis using Rényi's entropy, and it leads to the study of granular computing [Kins05]. In this case, instead of finding the perfect *n*, various orders of LE is conducted so that the contrast of true inconsistent data and false detected data can be enlarged. In this thesis, a maximum of 4^{th} order of weight vector $\Delta\mathbf{w}$ is performed to evaluate the LE of a newly measured data as is denoted by

$$\left|\Delta^{i}\mathbf{w}_{t+n}\right| = \left|\Delta^{i-1}\mathbf{w}_{t+n} - \Delta^{i-1}\mathbf{w}_{t+n-1}\right|,\ i = 2,3,4 \tag{3.32}$$

The LE value of a true inconsistent data is enlarged when orders increase, which is regarded as multi-scale analysis. This approach substantially decreased the difficulty of finding the perfect *n* former measured data (*i.e.*, the appropriate partial trajectory). For instance, a trajectory is consisted of 800 measured data ranging from 100^{th} to 900^{th} second. The LE of the trajectory with various orders is illustrated in Fig. 3.10, and measured data at time 501^{st} second is perturbed. Due to the approximated value of *n* (*n* is 4 for any situation along the path), and the detection sensitivity parameter γ of the perturbed data that the first unusual magnitude appear is 9.4. The rest γ of other samples are around 1.6, it implies the inconsistency of measured data at time 501^{st} second is very high. However, the perturbed data cannot be identified through the 1^{st} order LE due to inappropriate value of *n*. Furthermore, the contrast is increasing when the number of order is increasing, and the perturbed data could be identified clearly from the 4^{th} order

LE. In this case, the misjudgement is optimized so that the embedded new information of a newly measured data can be evaluated through multi-scale analysis of its LE.

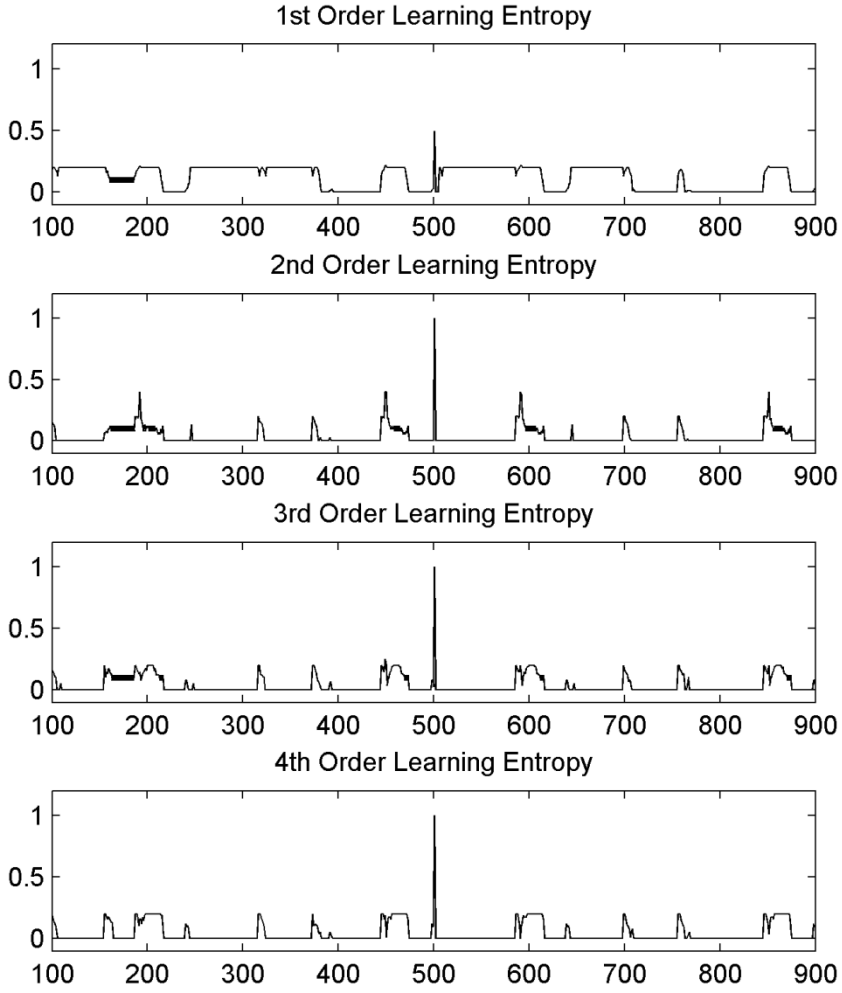


Fig. 3.10: The LE with various orders.

The LE is used to evaluate the performance of redressing sensory data using a rough set theory, and results are compared with the conventional fuzzy based control algorithm. The amount of new information of a perturbed data is quantified in order to apply the corresponding level of redressing method.

3.8 Summary

This chapter presents the fundamental knowledge of the perception-based computing. A fuzzy set theory is first described including important characteristics and terminology. The previous related work using fuzzy based control is reviewed and the drawback of the CFBC is summarised. A rough set theory is then introduced to filter inaccurate sensory data. Furthermore, through comparison of the rough set and the fuzzy set, the feasibility of implementing perception-based computing by combining rough set theory and fuzzy set theory is discussed. The Shannon's information measure of uncertainty is introduced, and it is considered as an important measure for the RFBC performance evaluation in terms of improving uncertainty reasoning. A conditional entropy measure is described to further evaluate the control accuracy and stability. Furthermore, in order to identify the control behaviour of an algorithm, the significance of Rényi's entropy measure is discussed. The possibility of multi-scale analysis for behaviour extraction is raised. For a perturbed data, the learning entropy is used to quantify the amount of new information with respect to its former measured data. The next chapter explicitly explain the mathematical implementation procedure of the RFBC.

Chapter 4

ALGORITHM IMPLEMENTATION

4.1 Proposed Method

This chapter presents a methodology of the proposed perception-based computing for a vision-based indoor navigation algorithm. It is implemented by a combination of rough and fuzzy theories. Prior to details of the control algorithm, a mathematical modelling of the indoor navigation behaviour is conducted in Sec. 4.2. A mathematical interaction model refines objects and parameters that are essential for the perception-based computing implementation. The interaction model also facilities the description of the RFBC algorithm. Based on the given model, the Sec. 4.3 details the construction of a CFBC design. In addition, the schematic model of a mobile robot and a wall is illustrated in Sec. 4.3. An explicit description of linguistic variables used throughout this chapter is also provided.

As stated in Sec. 3.4, a CFBC is widely used in a number of applications [Dhar13]. It provides a basic concept of linguistic pattern classification, which does not employ any precise mathematical formulation [SaYe98]. The CFBC is able to produce excellent

performance in uncertainty environment. However, the property of fuzzy theory determines that a fuzzy logic design focuses on correcting outputs based on given inputs. It does not have the capability of redressing inputs. In practice, sensory data may not be fully accurate, in other words, any input in a series of inputs could contains errors due to unexpected environment interference [PaGS05]. In this case, an additional filtering algorithm that regulates inputs is necessary in order to improve uncertainty reasoning.

The rough set theory is introduced for this reason in Sec. 4.4 of this chapter, and it is able to refine sensory data in order to identify and redress inaccurate inputs. In the RFBC algorithm, in contrary with the CFBC algorithm, a rough-fuzzy logic design uses corrected inputs for reasoning tasks. The proposed RFBC algorithm requires a set of historical data named database including inputs and their corresponding outputs [SkWa12]. This can be done by collecting operation record of the CFBC. The database is then analyzed using rough set theory. Data contain similar patterns are logically grouped. In this case, sensory records are categorized into equivalence classes based on its patterns. Since the history data reflect the practical environment, these equivalences are used as a middle layer between a real-time input and pre-defined fuzzy output classes. In other words, the real-time sensory data of the RFBC are corrected according to general patterns of the equivalence class. Furthermore, the accuracy of an input is optimized before it is transferred to the fuzzy inference system (Sec. 3.3). Details of the proposed RFBC algorithms are expressed and related pseudo codes are illustrated in Sec. 4.4.

4.2 System Modelling

The interaction between a mobile robot and the indoor environment is modelled as an incomplete information system [ZhMi04]. This is because sensory data are limited compared with data related to objects of interest [Pawl82].

$$S = \langle X, A, V, \rho \rangle \tag{4.1}$$

where X is an arbitrary subset of discourse U, A is the set of features associated with X. ρ is an information function: $\rho : X \times A \rightarrow V$, it can be seen as information about an object in X. V is the union of sets of feature values of each object in X, denoted by

$$V = \bigcup V_{atr}, \quad atr \in A \tag{4.2}$$

For the wall-following task in this thesis, three types of data are measured constantly: (i) the shortest distance from a robot to a wall, (ii) a mobile robot's heading angle, and (iii) a wall's orientation. In this case, X contains position information of a mobile robot at each time spot. A includes distance, heading angle, and wall's orientation. ρ is the information induced from feature values, such as the robot is too close to a wall, distance is moderate, or heading angle towards the wall. The proposed rough-fuzzy controller is built based on a conventional fuzzy controller to enhance the uncertainty reasoning. Consequently, a CFBC is described first and then followed by the RFBC.

4.3 Conventional Fuzzy Based Control

In order to design a CFBC algorithm, a knowledge base is generated by human experts with sufficient experience for this particular fuzzy inference system. The goal of the design is to make a mobile robot follow given walls at a desired distance while the robot also maintains a constant maximum velocity. The Gaussian membership function (Sec. 3.2.2.2) is selected as the fuzzy membership functions for the purpose of better control results. Two antecedent variables are (i) measured distance *d* from a robot to a given wall and (ii) orientation difference θ between the robot's heading angle and the wall's orientation. The consequent variable is the mobile robot's heading angle ω.

The Fig. 4.1 demonstrates a mobile robot follows a given wall. The distance between the mobile robot and the wall is denoted by *d*. The orientation difference θ is divided in two parts. A θ_*^+ means the orientation difference is positive, in other words, the mobile robot is driving away from the given wall. It is denoted by θ_*^- if the mobile robot is moving towards the given wall. The subscript of θ represents the linguistic degree of orientation difference on either direction (*i.e.,* facing away from wall or towards wall). In practice, it is replaced by three distinct linguistic labels. Furthermore, ω is the mobile robot's actual heading angle.

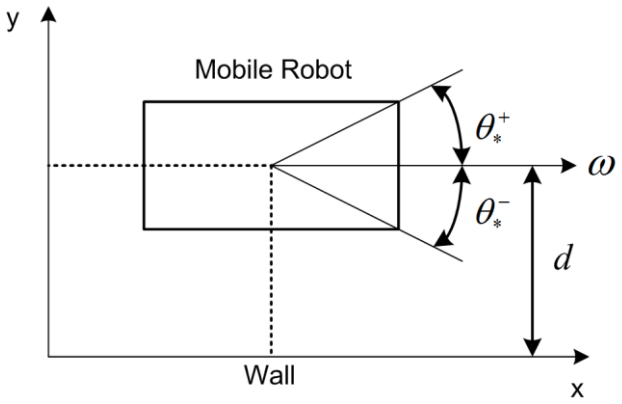


Fig. 4.1: Schematic model of a mobile robot and a given wall.

The distance d is the first input and it is described as five linguistic variables as shown in Fig. 4.2: d_{VN} (very near), d_N (near), d_M (moderate, it can be further subdivided as d_{MN} moderate on the near side and d_{MF} moderate on the far side), d_F (far), and d_{VF} (very far). They are equally partitioned within the range of [0, 5] meters. The orientation difference is the second input and it is described as five linguistic variables: θ_H^- (high angle towards wall), θ_L^- (low angle towards wall), θ_Z (parallel angle with the wall), θ_H^+ (high angle facing away from wall), θ_L^+ (low angle facing away from wall). They are also equally partitioned within the range of $[-50, 50]$ degrees.

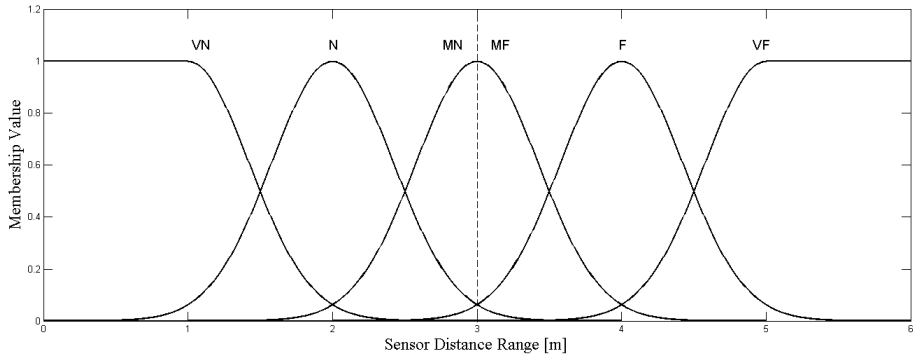


Fig. 4.2: Fuzzy membership functions for distance inputs.

The heading angle of a mobile robot is the output and it is described as five linguistic variables: ω_H^+ (high angle facing away from wall), ω_L^+ (low angle facing away from wall), ω_Z (parallel angle with the wall), ω_L^- (low angle towards wall), and ω_H^- (high angle towards wall). They are equally partitioned within the range of [-25, 25] degrees.

The CFBC is the basis of the RFBC and it is verified through experiments. The rules table of the CFBC is illustrated in the Table 4.1.

Table 4.1: Fuzzy inference rules.

	θ_H^+	θ_L^+	θ_Z	θ_L^-	θ_H^-
d_{VF}	ω_H^-	ω_H^-	ω_H^-	ω_H^-	ω_Z
d_F	ω_H^-	ω_H^-	ω_H^-	ω_H^-	ω_Z
d_{MF}	ω_H^-	ω_H^-	ω_L^-	ω_L^-	ω_Z
d_{MN}	ω_Z	ω_L^+	ω_L^+	ω_H^+	ω_H^+
d_N	ω_Z	ω_H^+	ω_H^+	ω_H^+	ω_H^+
d_{VN}	ω_Z	ω_H^+	ω_H^+	ω_H^+	ω_H^+

If the fuzzy inference rules table in Table 4.1 is considered to cover actions to all possible conditions (*i.e.,* distance and orientation difference) of a wall-following task, each real-time input data vector **x** then is said to be classified into one subfield in order to react. An input **x** is represented and reasoned using fuzzy inference rules as denoted by

$$\text{IF } d(t) \text{ is } d_j \text{ AND } \theta(t) \text{ is } \theta_k \text{ THEN } f(\mathbf{x}) \text{ is } y^{jk} \tag{4.3}$$

where $d(t)$ is the measured distance contained in **x** and $\theta(t)$ is the measured orientation difference contained in **x** at time *t*. The value of *j* is assigned from the set

{very near, near, moderate, far, very far}, and k is assigned from the set {zero, low, high}. y^{jk} is the linguistic output of the fuzzy inference kernel (see Sec. 3.3).

In order to achieve smooth control performance, 1/2 overlap of fuzzy sets is adopted in the design (as stated in Sec. 3.2.4). For the case of distance, it is implemented using Gaussian membership function with 1/2 overlap as shown in Fig. 4.2. The fuzzy inference rules are illustrated as follows

$$\text{IF } d(t) \text{ is } d_j \text{ THEN } f(d(t)) \text{ is } y^j \tag{4.4}$$

Generally, the Gaussian membership function (see Sec. 3.2.2.2) for a given fuzzy set is written as

$$\begin{aligned} \mu^i(d(t)) &= \mathrm{G}(m_i) \\ \mu^{i+1}(d(t)) &= \mathrm{G}(m_{i+1}) \end{aligned} \tag{4.5}$$

where m_i and m_{i+1} represent vertical dash line in Fig. 4.3, which is a zoomed in figure of Fig. 4.2.

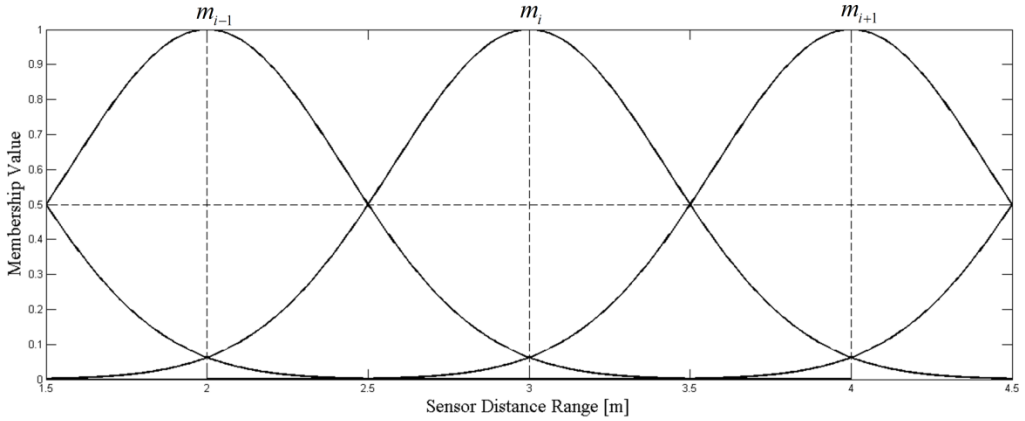


Fig. 4.3: Zoomed in Gaussian membership function with 1/2 overlap.

Similarly, we can also have two membership function values in the orientation difference domain. In this case, the comprehensive fuzzy membership values are

$$\begin{aligned}\mu^{i}(\mathbf{x}) &= \mu^{i}(d(t)) \cdot \mu^{i}(\theta(t)) \\ \mu^{i+1}(\mathbf{x}) &= \mu^{i+1}(d(t)) \cdot \mu^{i+1}(\theta(t))\end{aligned} \tag{4.6}$$

Therefore, the fuzzy inference system generates the follow interpolation in the interval $[i, i+1]$

$$f(\mathbf{x}) = \frac{y^{i}\mu^{i}(\mathbf{x}) + y^{i+1}\mu^{i+1}(\mathbf{x})}{\mu^{i}(\mathbf{x}) + \mu^{i+1}(\mathbf{x})} \tag{4.7}$$

where y^{i} and y^{i+1} are linguistic outputs for each fuzzy inference rule.

In brief, the CFBC algorithm can be summarized as the following procedure.

1. A knowledge base is built by human experts with sufficient fuzzy inference system experience.
2. A fuzzy inference rules table is generated according to the knowledge base.
3. While a mobile robot follows a given wall:
   (a) Classify the input *x* into one of the subfield of the fuzzy inference rules table.
   (b) Calculate fuzzy membership values (Gaussian membership function with 1/2 overlap) of each feature of the input data vector *x*.
   (c) The heading angle of a mobile robot is determined by comprehensive membership values of distance *d* and orientation difference θ.

4.4 Rough-Fuzzy Based Control

The proposed rough-fuzzy based control is built to improve uncertainty reasoning of the CFBC. To implement the proposed algorithm, a certain amount of samples were collected from CFBC to build an incomplete information system in Eq. (4.1) [Pawl82]. These data samples are denoted by X. According to simulation condition, particular adjustment was conducted to ensure samples are equally distributed along the navigation path. The set of features A of each data sample contains the measured distance and the measured orientation difference of a mobile robot at distinct location.

In the RFBC, equivalence classes are created according to behaviours induced from data samples. Because of environment interference, a potential measured value interval is set up so that measures in the specific interval would represent the same behaviour (*i.e.*, an equivalence class with identical features). Consequently, the selection of interval size is significant to the accuracy of information granules (Sec. 3.5.1). As described in Sec. 3.7.2, the Rényi's entropy is used to determine a proper interval size for equivalence classes. If the trajectory **J** in Fig. 3.7 is considered, for each moment order q ($-10 \le q \le 10$), the log-log plot of corresponding Rényi's entropy of the trajectory **J** at specific scales k ($0 \le k \le 12$) is illustrated in Fig. 4.4. If the scale is fixed to a particular value, value differences between curves are much smaller when q is close to ± 10, which is consistent with Fig. 3.8.

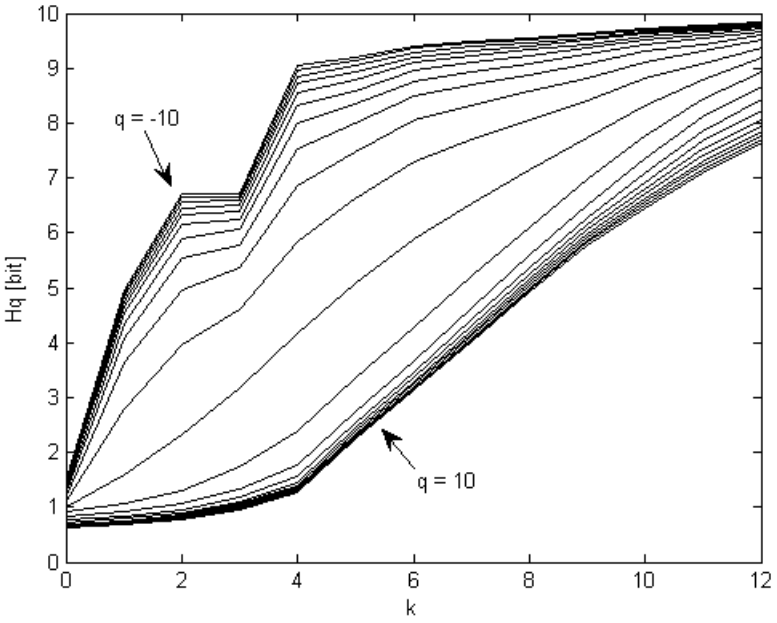


Fig. 4.4: Log-log plot of the Rényi's entropy versus scales of value interval size.

In general, more attentions are paid to curves with $q \geq 0$. As we can tell, the curves (*i.e.*, q is close to 10) begin to diverge when $k > 4$, which implies behaviours are further subdivided. However, behaviours created at theses scales may not reflect reliable results [FaKi05]. In other words, the behaviour of a trajectory **J** is stable when $0 \leq q \leq 10$ and $0 \leq k \leq 4$ through multi-scale analysis. Therefore, in order to achieve robust behaviour extraction, the value interval of potential measured distance for an equivalence class is set to $s_4 : 2^{-4} = 0.0625$ meter (as described in Sec. 3.7.2).

The collected data samples are partitioned into corresponding equivalence classes based on measured feature values V (Eq. (4.1)). In this case, an input data vector **x** is generated in real-time while a mobile robot follows a given wall using RFBC, and the learning entropy of **x** is also updated (detailed detection method is stated in Sec. 3.7.3).

If **x** is determined as perturbed data, the value would be redressed using the average value of recent measured data. Furthermore, each real-time input data vector **x** then can be categorized into specific equivalence class by calculating the nearest distance to equivalence classes in the input space. The rough-fuzzy membership value of an input data vector is then determined by the cardinality of corresponding equivalence class and the fuzzy membership values of features. It is calculated as follows [SaYe98] [PaSk94]

$$r_F^R(\mathbf{x}) = \text{cardinality}([\mathbf{x}]_R \cap F) \tag{4.8}$$

where $\mathbf{x}$ is an input data vector and $[\mathbf{x}]_R$ is the equivalence class that $\mathbf{x}$ belongs to. F is the given fuzzy set of features. The algebraic minimum operator is used for intersection operation. The cardinality(F) means the cardinality of a given set and the method in [Zade99a] [Dhar13] can be used

$$\text{cardinality}(F) = \sum_{atr \in A} \mu_F(V_{atr}) \tag{4.9}$$

where the μ_F is a fuzzy membership function for a given fuzzy set F. V_{atr} is the value of particular feature of the input data vector **x** (Eq. (4.2)). Since **x** contains two features (measured distance and measured orientation difference), the product operation in [SiBo04] can be used

$$\mu_F(V_{d(t)\cdot\theta(t)}) = \mu_{F_d}(V_{d(t)}) \cdot \mu_{F_\theta}(V_{\theta(t)}), \quad d, \theta \in A \tag{4.10}$$

where d and θ represent distance and orientation difference, respectively. F_d and F_θ are distinct fuzzy sets for features d and θ. By calculating the rough-fuzzy membership value of an input data vector $\mathbf{x}$ for each fuzzy set, the corresponding fuzzy inference rule

is fired only if the given fuzzy set has the maximum rough-fuzzy membership value among all fuzzy sets in Table 4.1. In other words, by calculating and comparing rough-fuzzy membership values in every subfield of Table 4.1, the system chooses the fuzzy inference rule that is the most appropriate one for given input $\mathbf{x}$.

Some notations are expressed before pseudo code is given. A data vector sample $\mathbf{x}_{sample}$ is a historical input data vector $\mathbf{x}$ acquired from operation of CFBC. It contains two features, which are distance d_{sample} and orientation difference θ_{sample}. *EC* is a set of equivalence classes. Algorithm 4.1 shows the cardinality calculation of each equivalence class. N_{ec} depends on the number of elements in a particular equivalence class. According to Table 4.1, each feature has five fuzzy sets. *dis* stands for five distance fuzzy sets and *agl* represents five fuzzy sets of orientation difference (Table 4.1). The detailed algorithm is expressed in Eq. (4.9) and Eq. (4.10).

Algorithm 4.1: Calculation of cardinality of a given equivalence class

1 : **for** $i = 1$ to n **do**
2 : **for** $j = 1$ to m **do**
3 : $$EC(i,j) = \sum_{ec=1}^{\mathrm{N}_{ec}} \sum_{dis=1}^{5} \sum_{agl=1}^{5} \mu_{dis}(d_{sample}^{ec}) \cdot \mu_{agl}(\theta_{sample}^{ec})$$
4 : **end for**
5 : **end for**

While a mobile robot follows a given wall, an input vector $\mathbf{x}$ is generated and partitioned into an equivalence class by calculating the nearest distance. As shown in Algorithm 4.2, the rough-fuzzy membership value (Eq. (4.8)) of each subfield in Table 4.1 for this equivalence class is calculated and compared. $CARD_{EC(i,j)}$ denotes the set of

all rough-fuzzy membership values of a particular equivalence class $EC(i, j)$. As a result, the subfield of Table 4.1 with the maximum rough-fuzzy membership value is selected.

Algorithm 4.2: Calculate and compare rough-fuzzy membership values for a given equivalence class

1: $max\_d \leftarrow 0$
2: $max\_g \leftarrow 0$
3: $max\_val \leftarrow 0$
4: **for** $dis = 1$ to 5 **do**
5: **for** $agl = 1$ to 5 **do**
6: $CARD_{EC(i,j)}^{dis,agl} = \sum_{ec=1}^{N_{ec}} \mu_{dis}(d^{ec}) \cdot \mu_{agl}(\theta^{ec})$
7: **if** $CARD_{EC(i,j)}^{dis,agl} > max\_val$ **do**
8: $max\_d \leftarrow dis$
9: $max\_g \leftarrow agl$
10: $max\_val \leftarrow CARD_{EC(i,j)}^{dis,agl}$
11: **end if**
12: **end for**
13: **end for**

Based on selected subfield in Table 4.1, a corresponding fuzzy inference rule is applied to an input data vector *x*. In CFBC, an output is resulted directly from an input *x*. By contrast, an output of RFBC is reasoned from the specific equivalence class. Since elements in an equivalence class are mutually indiscernible [Pawl82] [Pawl97], features of equivalence class are able to represent any element belongs to it. The practical meaning of this is to utilize historical data to redress current heading angle to be more accurate. In this case, if an input data vector is inconsistent with former inputs due to environment interference, RFBC is able to minimize errors and maintain steady status.

4.5 Summary

This chapter presents mathematical implementation procedure of the proposed rough-fuzzy based navigation algorithm. The idea is to elaborate the implementation method hierarchically so that the CFBC and RFBC can be properly linked. A system modeling is also conducted that facilities the description. The RFBC improves input accuracy by utilizing equivalence classes before applying fuzzy inference rules. In order to achieve robust behaviour extraction, a multi-scale analysis using the Rényi's entropy measure is conducted to determine the equivalence relation rules. The next chapter presents the role of the perception-based computing algorithm in a vision-based indoor navigation system.

Chapter 5

SYSTEM IMPLEMENTATION

5.1 System Architecture

The proposed vision-based indoor navigation system is implemented using a multi-layer design. The entire system consists of five subsystems: (i) vision sensory, (ii) image processing, (iii) control unit, (iv) interpretation, and (v) reaction. These five subsystems are located at four pieces of hardware: (i) a Kinect sensor, (ii) a portable computer, (iii) a Arduino Uno R3 board, and (iv) a mobile robot. The four pieces of hardware are connected logically as demonstrated in Fig. 5.1.

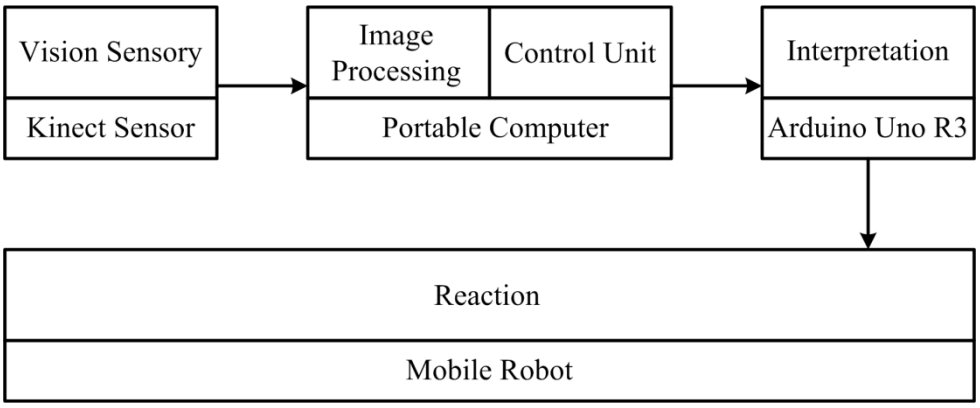


Fig. 5.1: System architecture.

The vision sensory subsystem collects via a Kinect sensor installed in the front of a mobile robot. The angle of the Kinect sensor is properly adjusted so that objects are observed with natural geometrical form. Data collected from the Kinect sensor creates depth maps and is transmitted to the portable computer. Each depth map is processed in the image processing subsystem, and a set of maximum and minimum depth information is acquired. Two proper points are extracted and a wall is created. The wall in this form could be either a real wall or a virtual wall. A real wall is referred as both points are extracted from the same object. By contrast, a virtual wall is created based on two points that are located at different objects. A virtual wall is able to prevent the mobile robot from unnecessary exploration and potential trap. Additionally, certain restrictions are applied so that a virtual wall would not block or mislead the navigation path.

The control unit subsystem guides the heading angle based on the extracted wall and related information. Both the image processing and the control unit are stored at the portable computer. The combination of image processing and control unit subsystems can be seen as a brain of the whole navigation system. However, they are independent from each other since their roles are different. The proposed navigation algorithm in Ch. 4 is the control unit subsystem. It reasons proper reactions to the environment based on vision sensory data. The image processing subsystem is critical since it is the only module that reforms and transfers the environment information to the control unit.

The interpretation subsystem translates outputs of the control unit into appropriate commands for a mobile robot. A microcontroller board named Arduino Uno R3 is

utilized to accomplish this task. The mobile robot is the final reaction device that executes reasoning results of the control unit.

The rest of this chapter is organized as follows. An explicit description of a Kinect sensor is provided in Sec. 5.2.1. The Sec. 5.2.2 presents details of other hardware including Arduino Uno R3 and the mobile platform. Furthermore, the image processing and the interpretation subsystems are presented in terms of functionalities and implementation methods in Sec. 5.3.

5.2 Hardware Description

5.2.1 Kinect Sensor

The first version of visual interactive device Kinect developed by the Microsoft Company was released into the market in November of 2010 [KaEG12]. It allows users to control and interact with their console or computer without the need of traditional interaction devices (*e.g.*, a gaming controller and mouse), through a natural user interface using gestures and spoken commands. The first generation of Kinect sensor was initially developed to improve the game experience of Xbox 360 gaming console. Shortly after that, a great amount of research interests had raised to connect a Kinect sensor to computers to explore more potentiality. An official *software development kit* (SDK) was also released in order to facilitate the exploration of Kinect sensors.

A Kinect sensor has an RGB camera, an infrared (IR) light source and an infrared light sensor as shown in Fig. 5.2. In this form, the device is able to estimate the distance

to objects observed in the environment [CoSc12]. Besides, it contains a microphone array which helps improve the recording of three-dimensional movement of objects in front of it. The motorized tilt enables a Kinect sensor to point at a proper view angle. A three-axis accelerometer is integrated in Kinect sensor and it is possible to use the accelerometer to determine the current orientation of the sensor.

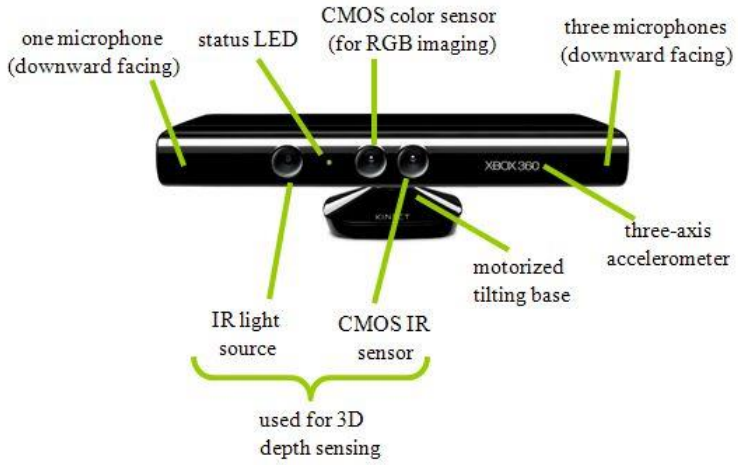


Fig. 5.2: Kinect with sensors labelled.

The minimum effective detection range for a Kinect sensor is 0.4 meter and maximum range is 3.5 meters. The resolution is 10 millimeters at 2 meters range [OlKa12]. The field of view is 58 degrees. A Kinect sensor provides multiple frame formats for both depth image and standard color image ranging from 640×480 to 1280×960 pixels at maximum 30 frames per second (PFS). By combining both hardware (IR sensor) and software, a Kinect sensor has the ability of building depth maps within quite short period (4 milliseconds for depth image with resolution of 640×480 pixels). A Kinect sensor is easy to install and integrate since it provides the standard SDK that would facilitate the use of a Kinect sensor. The comparison of a Kinect sensor and

another vision sensor GOPRO Hero3 is elaborated in Appendix C.3. Furthermore, in order to enhance the mobility of a Kinect sensor, the power supply modification method is also stated in Appendix C.3.

5.2.2 Other Hardware Components

In the workflow of the proposed vision-based indoor navigation system, the control unit subsystem guides the movement of a mobile robot by correcting the heading angle. In order to translate outputs of the control unit into proper inputs of the mobile platform, this thesis proposes the utilization of a single-board microcontroller called Arduino [ArCa12]. Arduino is an open hardware/software platform. An Arduino is able to receive inputs from a variety of sensors and then affect its surroundings by controlling attached components and actuators. Open source software named Arduino programming language and corresponding Arduino development environment are provided to program the on-board microcontroller freely [Busa12] [ArPo13]. Besides, the low expense is another consideration of selecting the interpretation device. The well balance between cost and efficiency makes it the ideal solution for academic purpose [JuLu13].

During the navigation period, the control unit subsystem communicates with the Arduino Uno R3 through serial communications. The interpretation subsystem stored at the device receives inputs from USB port and interprets them into proper signals. The general motors and servo motors are then controlled by the Arduino Uno R3 through I/O pins. Two digital I/O pins are used for forward and backward movement, respectively. One pulse width modulation (PWM) I/O pin is connected to a servo motor (see Appendix

C.1) which is in charge of the steering function. The power of the servo motor is also supplied by 5V output voltage pin on the Arduino board. The details of software are described in Sec. 5.3.2 of this chapter. More technical specifications and parameters of the Arduino Uno R3 are stated in Appendix C.2.

The reaction subsystem is implemented using a remote control (RC) mobile car. The RC car is able to perform basic movements (*i.e.,* moving forward, backward, and steering). The original steering mechanism is modified using a servo motor as described in Appendix C.1. In this case, it is able to steer in the range of 25 and -25 degrees with step precision of 1 degree. The steering speed is relative fast so that the time cost can be omitted. The maximum load of the RC car is 3 kg. In a real situation, given the same command duration, the moving condition of a RC car is influenced by a number of interference. Therefore, the command duration should be adjusted according to the navigation environment

5.3 Software Description

5.3.1 Image Processing

The image processing subsystem is implemented in C# programming language using Microsoft Visual Studio 2010. The running OS environment is Windows 7. The software primarily includes two sections. First of all, a *graphical user interface* (GUI) is provided (Fig. 5.3). A user can observe runtime system information (*e.g.,* both RGB and greyscale frames, depth data, and system settings) displayed in the panel. The GUI also enables a

user to adjust system settings during the operation. Secondly, the core of image processing subsystem receives frames from a Kinect sensor, and transfers data to both the display panel and the control unit subsystem. The rest of this section describes these two parts in a sequential order.

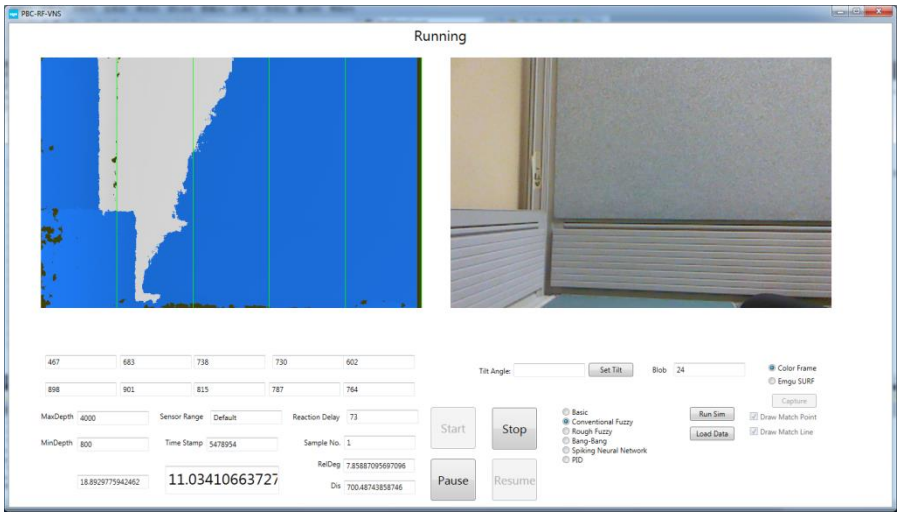


Fig. 5.3: The software GUI display panel.

As illustrated in Fig. 5.3, the software GUI display panel contains two screens for videos streamed from a Kinect sensor. The one on the left is allocated for depth frames and the other screen displays color frames. Both video resolutions are 640×480 pixels. The depth image is divided vertically into five parts. The corresponding maximum and minimum depth information are illustrated below each frame part. The maximum and minimum depth sensory range setting of a Kinect sensor is listed below the depth information area. The decision made by a control algorithm is shown in a bold text below the Kinect sensory range setting. The total processing time duration of each video frame varies depends on the complexity of frame and is shown on the left side of start button. If the sampling function is active, then the current sampling number is shown. A mobile

robot begins to move once the start button is pressed. It stops either by clicking the stop button or approaching too close to objects. There is a set of control algorithms options for user to choose, which is listed on the right side of the stop button. The tilt angle of the Kinect sensor can be adjusted in the range of -27 to 27 degrees. Besides, there is a status label in the top center of the panel indicating the current system condition.

The image processing subsystem is responsible for reformation and transformation of incoming frames. A dispatcher timer is set up with constant time interval (33 milliseconds). An event handler is raised whenever the timer expires. The event handler is designed to fetch both color and depth frames from the cache and distribute them to corresponding sub-functions. A primary goal is to divide the depth frame into five vertical parts and calculate minimum and maximum depth information of each sub frame. The relative angle between a mobile robot and a wall can be calculated by the depth information using computer graphics. Each frame is flipped horizontally before sending to software GUI. It is because the Kinect sensor captured mirrored frames for the purpose of better gaming performance.

The interface of control unit subsystem is used to transfer sensory data to the specific RFBC algorithm. Since Matlab exhibits advanced ability in terms of mathematical calculation, the control unit is written in Matlab language and is packaged as dynamic-link library (DLL) files. A Matlab Compiler Runtime is used to support the calling of DLL files in a C# project. The measurement of distance and orientation difference is combined as one sensory input. According to a user's preference, the input is passed to

distinct type of control unit (*e.g.*, CFBC or RFBC). The control unit utilized the sensory input to reason steering angle and then transfers the result to the interpretation subsystem.

Besides the reaction behaviours reasoned from the control unit, the navigation system has basic obstacle avoidance scheme. It stops whenever more than half subsections of a depth image have distance less than minimum tolerance. It implies that, under general circumstance (*e.g.*, facing wall with large orientation difference), a mobile robot cannot get out without hitting the wall.

5.3.2 Interpretation

The interpretation subsystem is stored at the Arduino Uno R3. It is responsible for translation outputs of the control unit subsystem. The Arduino programming language is used to write functions, which is basically C language and follows most C programming syntax formats. As shown in Algorithm 5.1, the program mainly contains three sub-functions. The definition of I/O pins, serial communications baud rate, and initial state of outputs are defined in the first sub-function “setup()”. Once parameters and settings are written to the Arduino Uno R3 board, the second sub-function is performed as a loop, called “loop()”. The serial event handler watches the port. It receives incoming data and invokes decoding sub-function “execution()” whenever the control unit produces reasoning results. To move the mobile robot forward or backward, the corresponding I/O pin is set to low. According to outputs of the control unit, an *application programming interface* (API) for the Arduino programming environment named “Servo” is used to guide the servo motor to move to desired position. For each new command, a timer is

setup to ensure the moving operation is executed properly. The duration of the timer is adjusted according to the practical experiment environment.

Algorithm 5.1: Arduino mid-ware interpretation code

```
1:  interval_Millis ← 500
2:  void setup()
3:  {
4:      pin_forward ← HIGH
5:      pin_backward ← HIGH
6:      Serial.begin ← 9600
7:      servo ← INIT_POSITION
8:  }
9:  void loop()
10: {
11:     current_Millis ← millis();
12:     if Serial.available() > 0
13:         previous_Millis ← current_Millis
14:         buffer ← Serial.read()
15:     endif
16:     execution(buffer)
17:     if current_Millis − previous_Millis > interval_Millis
18:         pin_forward ← HIGH
19:         pin_backward ← HIGH
20:         servo ← INIT_POSITION
21:   endif
22: }
```

5.4 Summary

This chapter presents the vision-based indoor navigation system architecture, and followed by important hardware and software description that constitute a fully

functional autonomous robot. Five subsystems are elaborated and the relationship between them and hardware components are also stated. The approach of linking control algorithms to the entire navigation system is discussed, and a pseudo code of the interpretation subsystem is listed.

Chapter 6

DESIGN OF EXPERIMENTS

6.1 Simulation Setup

In this thesis, the wall-following task is presented to test the proposed rough-fuzzy based indoor navigation algorithm. As described in Sec. 4.3, two parameters are measured constantly during the simulation, which are the shortest distance *d* from mobile robot to the wall, and the orientation difference θ between the mobile robot's heading angle and the wall's orientation. These two parameters exhibit two important features such that *d* changes along with θ and θ is determined according to *d*.

By summarizing from Fig. 1.1 in Sec. 1.2, the indoor environment tackled by this thesis can be generalized into three types of walls: (i) straight wall, (ii) curved wall, and (iii) wall with sharp corner. A straight wall has the feature of fixed orientation, therefore θ only varies along with the mobile robot's heading angle. Consequently, it is selected as primary algorithm applicability testing. A curved wall is used for algorithm robustness testing purpose since the change rate of its orientation is not fixed. Furthermore, a curved wall is derived into several conditions by modifying the orientation changing rate, such as

U-turns and round corners. A sharp corner is common in an indoor environment and it is usually subdivided into concave corners and convex corners. This thesis uses both types of corner with 90 degrees. Those three types of situations are considered to cover the majority of an indoor environment, and corresponding following behaviours are important to an intelligent navigation algorithm.

The velocity of a mobile robot in the simulation is set to a constant value. In order to achieve better illustration result, the desired distance from robot to the wall is set to 3 meters so that the trajectory fluctuation can be clearly observed. At the beginning of each test round, the mobile robot is placed at 1 meter away from the wall with the same orientation as the wall at that location. The purpose of such configuration is to test RFBC more efficiently.

Three comparison algorithms are also tested in the same indoor environment. They are boundary based control (BBC), Spiking neural network (SNN), and CFBC. The boundary based control or so called Bang-Bang controller is applied to make a mobile robot move within the particular distance range. The robot turns its direction to pre-defined 15 or -15 degrees if it hits the minimum or maximum distance boundary. The boundaries in the experiment are set to 0.3 meter on each side of the desired distance. The spiking neural network is a comparison of perceptual computing algorithm and is elaborated in Sec. 2.7. The conventional fuzzy based control is fully described in Sec. 4.3.

For the Shannon uncertainty measures in the simulation and experiment, the base threshold and advanced threshold stated in Sec. 3.7 is set to 0.3 m and 0.1 m on each side,

respectively. Since the overall possible value is in the interval of $[0,5]$ m (See Sec. 4.3), the control precision of each threshold is $\pm 6\%$ and $\pm 2\%$, respectively.

6.2 Experiments Environment

The experiment of the proposed perception-based computing for vision-based indoor navigation algorithm is performed using a mobile robot described in Ch. 5. A straight wall and a wall with 90 degrees corner are used in the experiment. Results are compared and discussed with the same type of results in the simulation. Compared with the simulation environment, the *signal to noise ratio* (SNR) is much higher and more complicated. Consequently, results from experiment show more fluctuation than results from simulation. Nevertheless, the same pattern is still able to be identified. Furthermore, deeper discussion of uncertainty reasoning improvement using RFBC is conducted. It proofs that the conducted experiment follows the same theoretical approach as the simulation. The mobile robot used for the vision-based indoor navigation system is shown in Fig. 6.1. It is designed to navigate automatically along given walls and try to keep desired distance. The Kinect sensor is pointing at a wall on its left side with a horizontal view angle. The turning angle of the mobile robot is $[-25, 25]$ degrees, and the speed is adjusted according to floor condition and control accuracy. It can be done by modifying signal interval generating from the Arduino mid-ware to the RC platform. Currently, each round of test lasts about 60 seconds and a mobile robot is designed to navigate through 7.5 meters wall, which implies the average speed is around 0.12 meter per second. In addition, images taken from a Kinect sensor is cropped in order to focus on

interested area (*i.e.*, given walls), thus the interference of unnecessary objects is eliminated.

Fig. 6.1: The mobile robot.

The experiment is taken place along the hallway on the fifth floor of Engineering Information and Technology Complex 1 at University of Manitoba as shown in Fig. 6.2. The hallway has the feature that the left side wall is consistent in terms of wall type, texture, and flatness. Furthermore, the right side wall is comprised of walls, doors, and glasses. The environmental interference is more complicated and vision sensory is said to be appropriate for glass materials. The left side wall of the hallway is used as the straight wall. More challenging situations can be made based on the straight wall. The corner test is conducted using two perpendicular straight walls with the same texture to ensure equal distance measurements.

Fig. 6.2: The experiment environment.

6.3 Summary

This chapter presents the simulation setup and the experiment environment. Parameters and comparison algorithms are emphasized. The type of selected wall is discussed and it is suggested that some horizontal comparisons can be done between simulations and experiments. The next chapter shows results conducted based on the settings described in this chapter.

Chapter 7

RESULTS AND DISCUSSIONS

The proposed rough-fuzzy based control is tested through simulation using Matlab 2012b in the Windows 7 operating system. Based on property of algorithms, they can be separated mainly into two categories, adaptive (BBC) and perceptual (SNN, CFBC and RFBC) algorithms. Experiments are also conducted and some comparisons with simulation results are analyzed.

7.1 Simulation Results

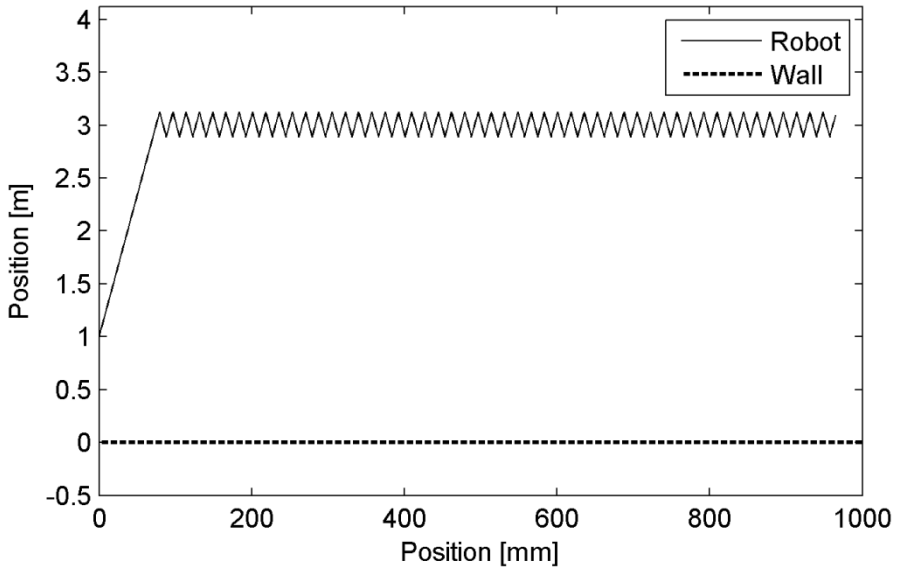
7.1.1 Straight Wall Simulation

The simulation results of straight wall are illustrated in Fig. 7.1 and Table 7.1. The BBC (Fig. 7.1(a)) moves a mobile robot to desired position with slower rate than other algorithms. The SNN (Fig. 7.1(b)) shows smooth approaching curve, but the convergence rate is not optimal. The CFBC (Fig. 7.1(c)) shows similar results as the proposed RFBC (Fig. 7.1(d)) in terms of overall performance. However, as illustrated in Fig. 7.2, the steady-state of CFBC (Fig. 7.2(a)) fluctuates larger than RFBC (Fig. 7.2(b)).

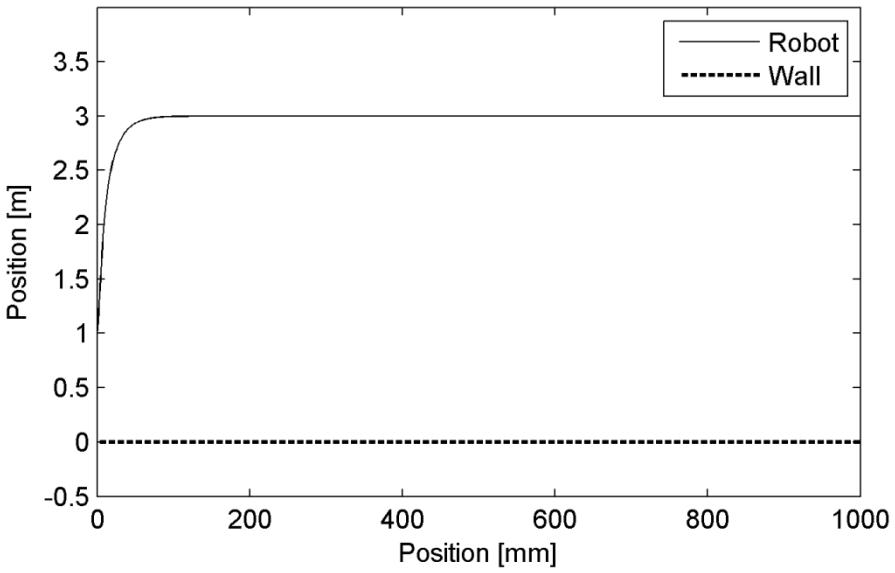
Though the fluctuation amplitude of BBC is within base threshold according to parameter settings (see Sec. 6.1), the entropy of BBC of shows the worst case among all algorithms. It means the BBC results contain much more new information, so that the expected value interval is wider than other algorithms. The entropy of SNN shows the best value in terms of overall performance, which mainly because of its smooth trajectory. The RFBC slightly improved the performance compared with the CFBC. However, the RFBC demonstrates the best conditional entropy of the advanced threshold given the base threshold (as stated in Sec. 3.7). It indicates that the RFBC trajectory is closer to a desired path, and the period that a mobile robot stays on a desired path is the longest.

Table 7.1: Simulation data using straight wall.

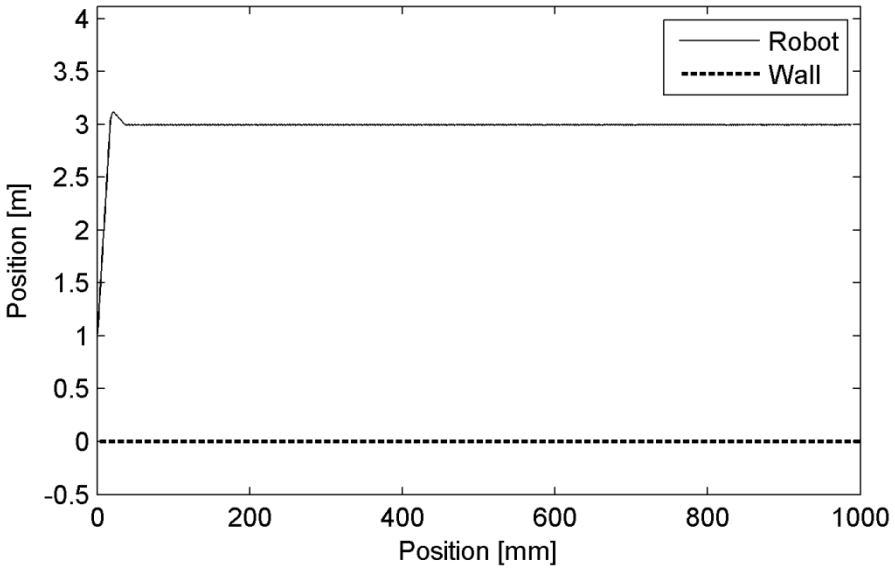
Straight Wall	Mean [m]	Variance [m]	Entropy [bit]	Conditional Entropy [bit]
BBC	2.9275	0.1041	2.0633	0.1490
SNN	2.9630	0.0405	1.0071	0.0257
CFBC	2.9713	0.0379	1.2014	0.0115
RFBC	2.9838	0.0389	1.1644	0.0043



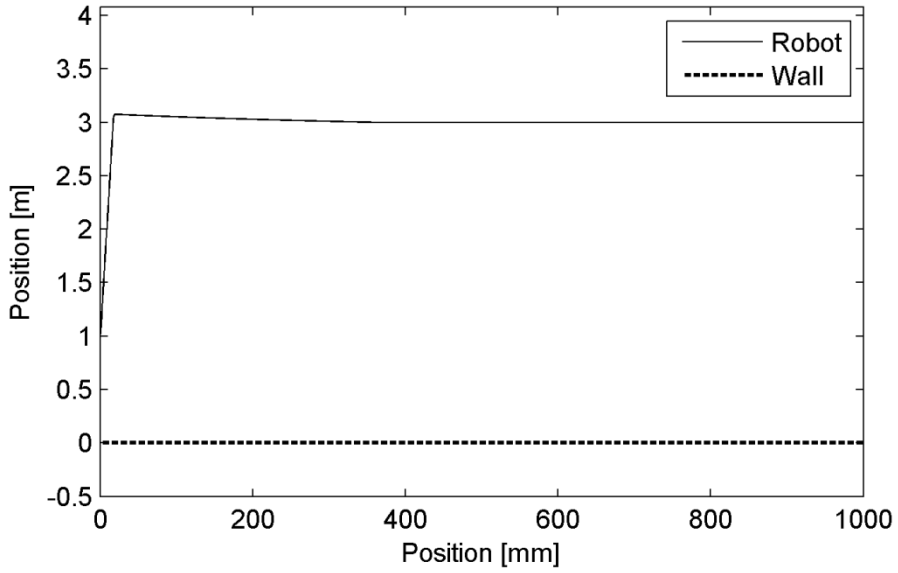
(a)


Robot
Wall
Position [m]
Position [mm]
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
200
400
600
800
1000

(b)

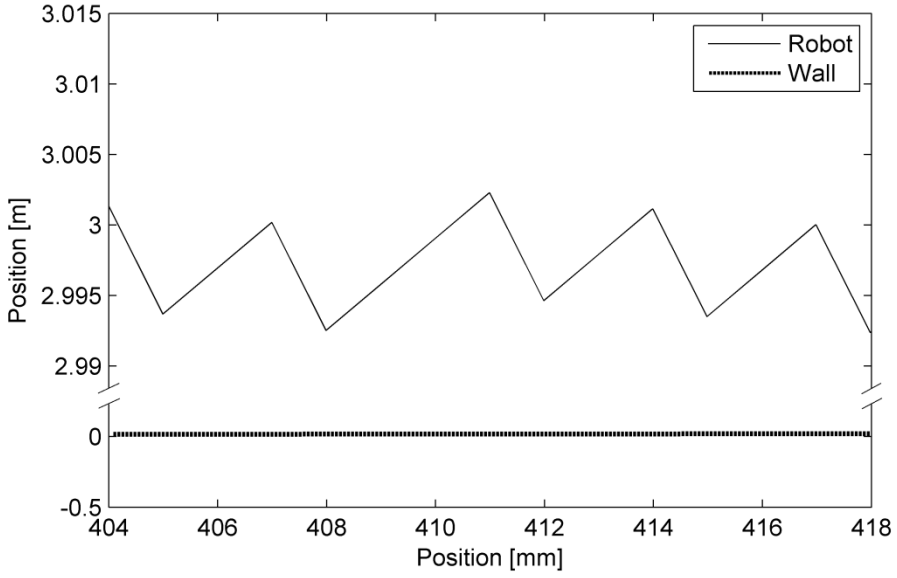

Robot
Wall
Position [m]
Position [mm]
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
200
400
600
800
1000

(c)

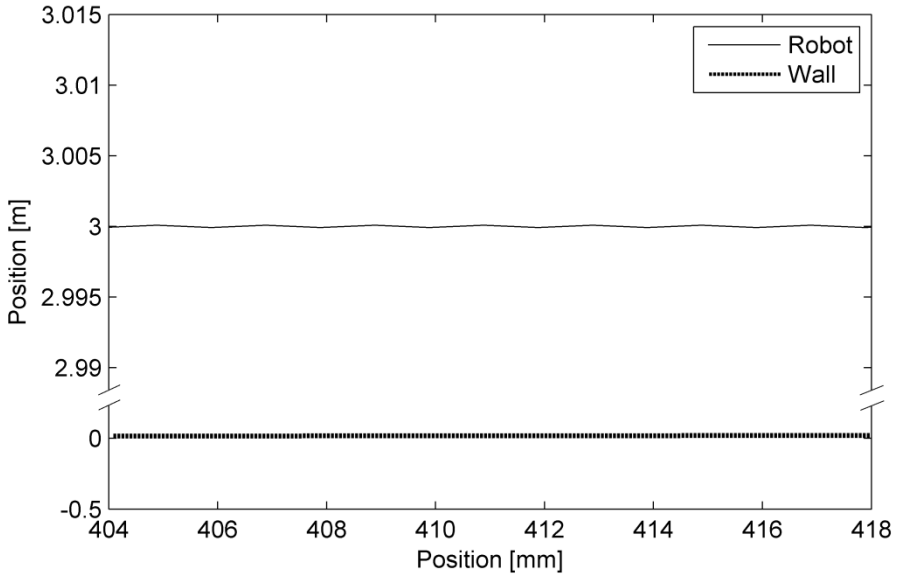


(d)

Fig. 7.1: System performance with straight wall using (a) BBC, (b) SNN, (c) CFBC, and (d) RFBC.



(a)



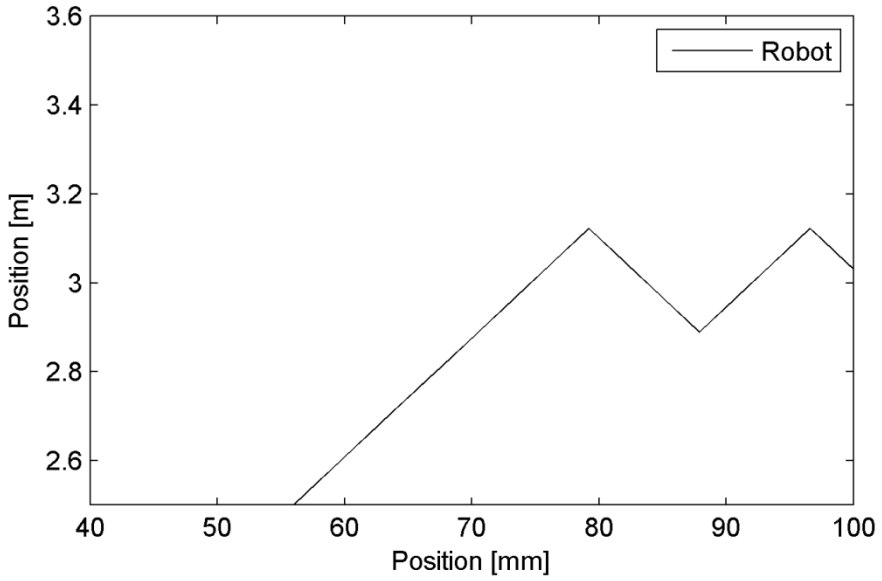
(b)

Fig. 7.2: Zoomed in comparison between (a) CFBC and (b) RFBC.

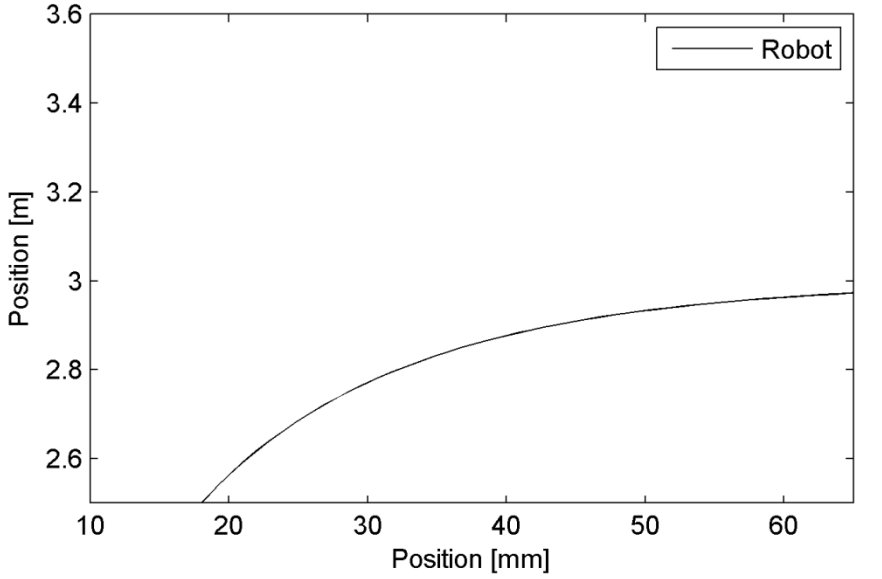
In the early stage of the simulation, a mobile robot is placed at 1 meter away from wall. Compared with desired distance of 3 meters, the robot is very close to a given wall. Consequently, it approaches the desired distance with large heading angle facing away from wall. All algorithms except the BBC (Fig. 7.3(a)) perform well in terms of the approaching rate. The approaching curve of the SNN (Fig. 7.3(b)) controller is smooth and stable. However, its response rate to the environment is slower than other perceptual computing algorithms. It is not competitive if there is a time restriction.

The proposed RFBC (Fig. 7.3(d)) minimizes overshoot while a mobile robot approaches the desired distance. On the one hand, in the indoor navigation, the time of stay on a desired path could be very limited due to complex layout. In other words, a mobile robot may always try to approach a desire distance. An efficient response to

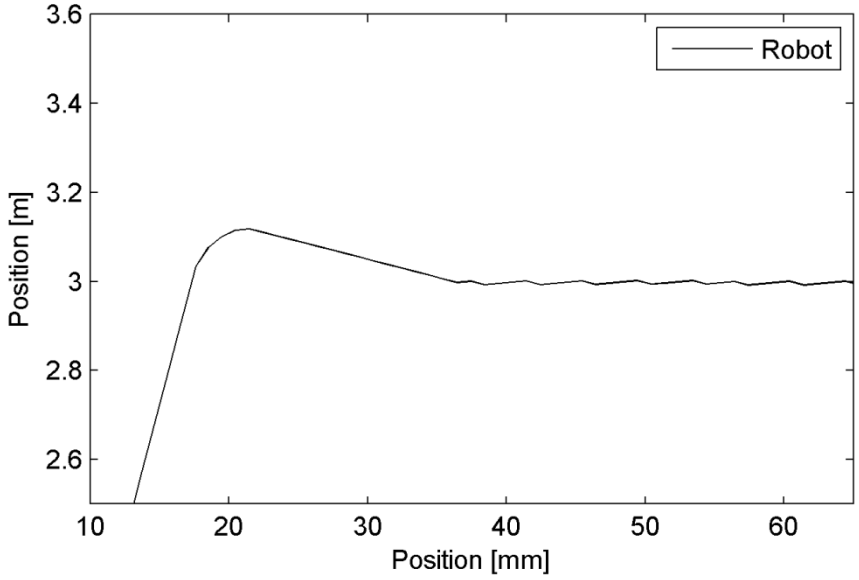
environment is obviously beneficial to the system performance. On the other hand, overshoot should be minimized since it may lead a mobile robot to unstable status. Compared with CFBC (Fig. 7.3(c)), RFBC enhances the control behaviour, and makes the approaching curve smoother.



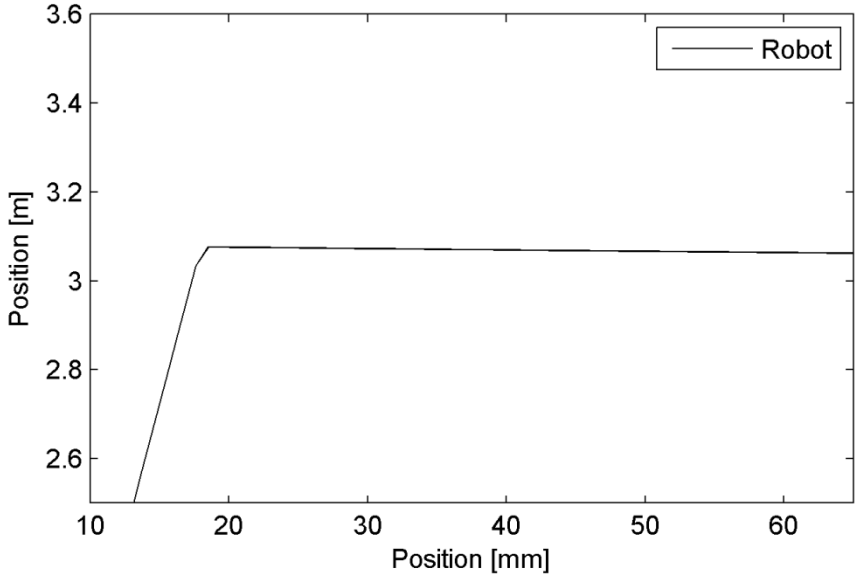
(a)



(b)



(c)



(d)

Fig. 7.3: Zoomed in comparison of transient-state of (a) BBC, (b) SNN, (c) CFBC, and (d) RFBC.

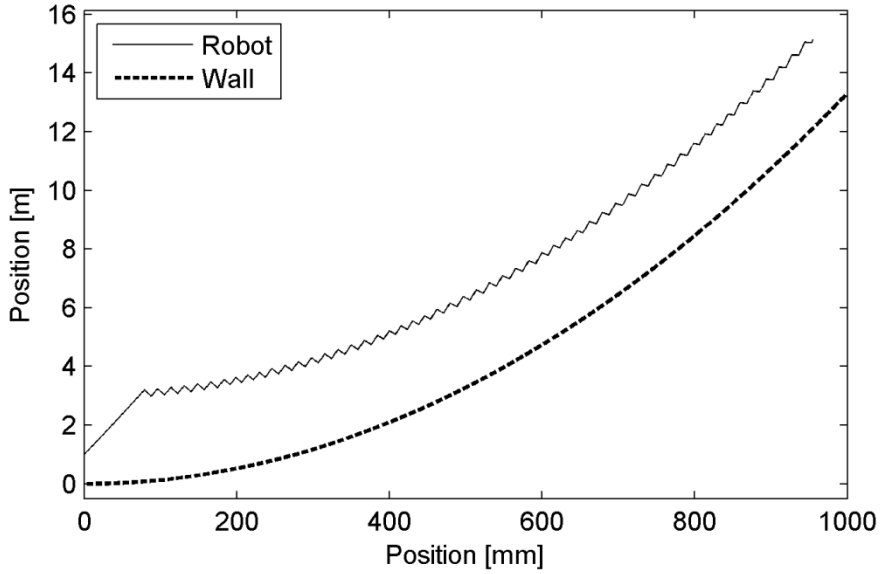
7.1.2 Curved Wall Simulation

The second simulation is conducted using curved wall and results are illustrated in Fig. 7.4 and Table 7.2. The trajectory of BBC (Fig. 7.4(a)) contains more obvious fluctuation than other algorithms. The SNN controller (Fig. 7.4(b)) shows smooth approach behaviour, but the average distance is lower than CFBC (Fig. 7.4(c)) and RFBC (Fig. 7.4(d)). The Fig. 7.5 demonstrates the zoomed in comparison of CFBC (Fig. 7.5(a)) and RFBC (Fig. 7.5(b)) in terms of their steady-state period, it shows that RFBC is smoother and more stable.

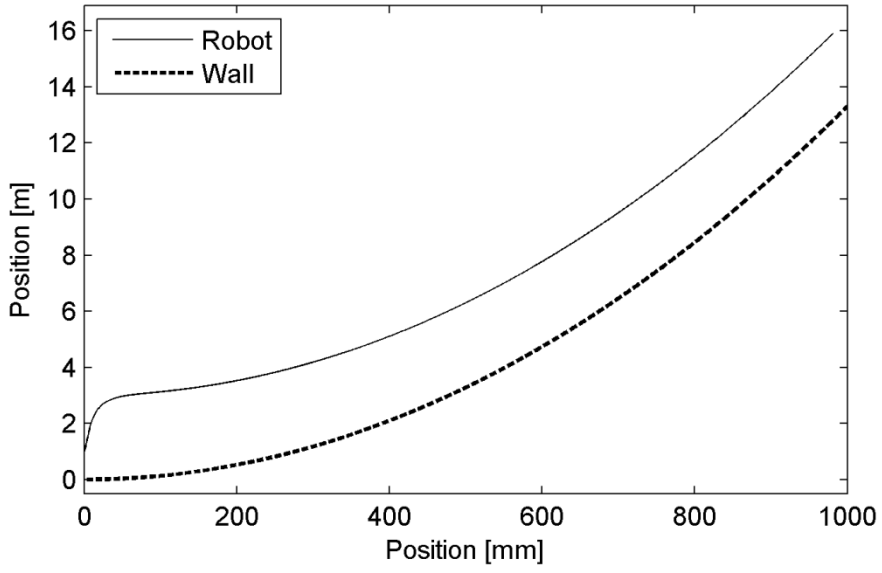
The entropies of SNN in Table 7.2 are the smallest among all algorithms. However, the conditional entropy of SNN is worse than CFBC and RFBC. The reason should be explained through a comprehensive analysis of mean and entropy. We can observe that the average distance of SNN is lower and the uncertainty is low. It is because the distance measures of SNN indeed fall into specific value interval, but the value interval does not meet the requirement of excellent performance (*i.e.*, in the advanced threshold). From a graphical point of view, a mobile robot using SNN is always almost "catch up" with the desired path during the navigation period. Therefore, the conditional entropy of SNN is not as good as CFBC and RFBC. Furthermore, the stability of RFBC is verified since the conditional entropy value is similar in both Table 7.1 and Table 7.2. This is important because it indicates the RFBC is able to maintain consistent performance if the orientation changing rate of a wall is low.

Table 7.2: Simulation data using curved wall.

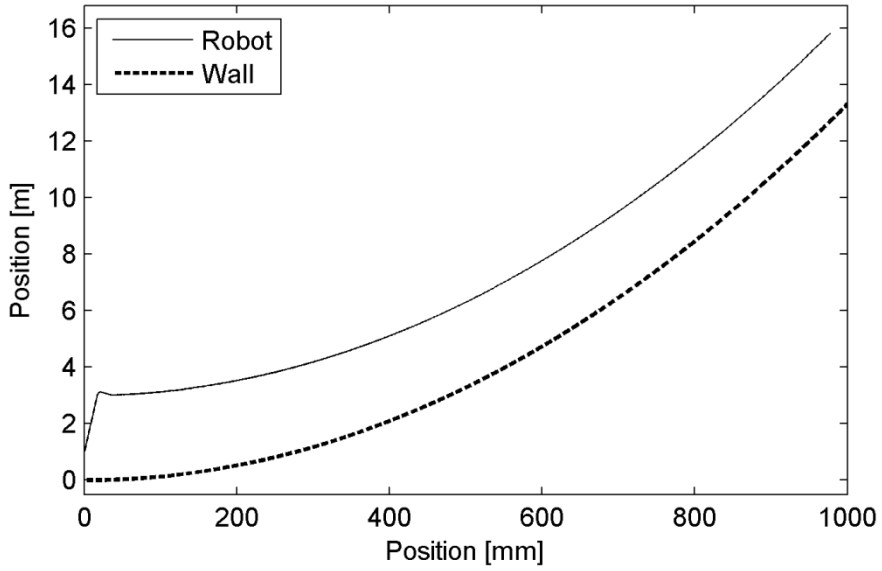
Curved Wall	Mean [m]	Variance [m]	Entropy [bit]	Conditional Entropy [bit]
BBC	2.9217	0.1031	2.0779	0.1552
SNN	2.9590	0.0402	0.4761	0.0257
CFBC	2.9721	0.0381	1.2944	0.0115
RFBC	2.9818	0.0388	1.2283	0.0043



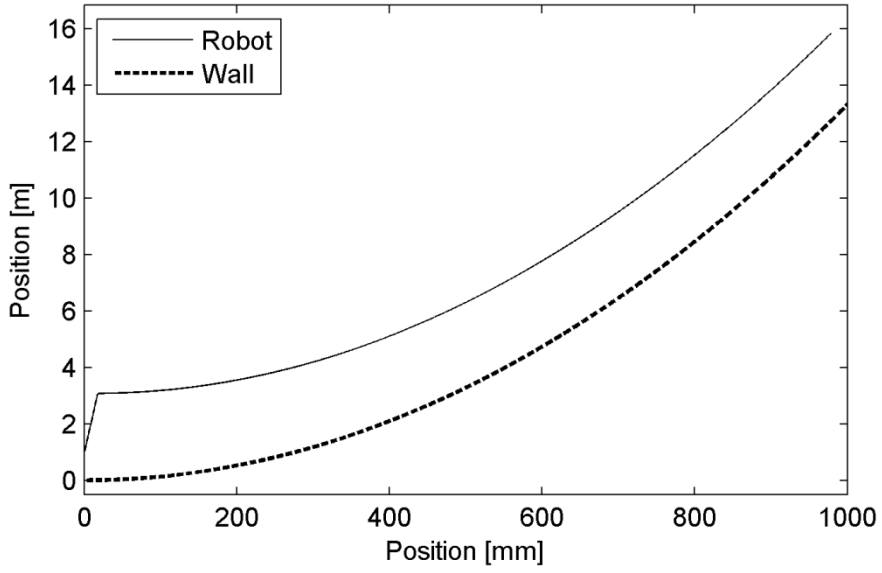
(a)



(b)

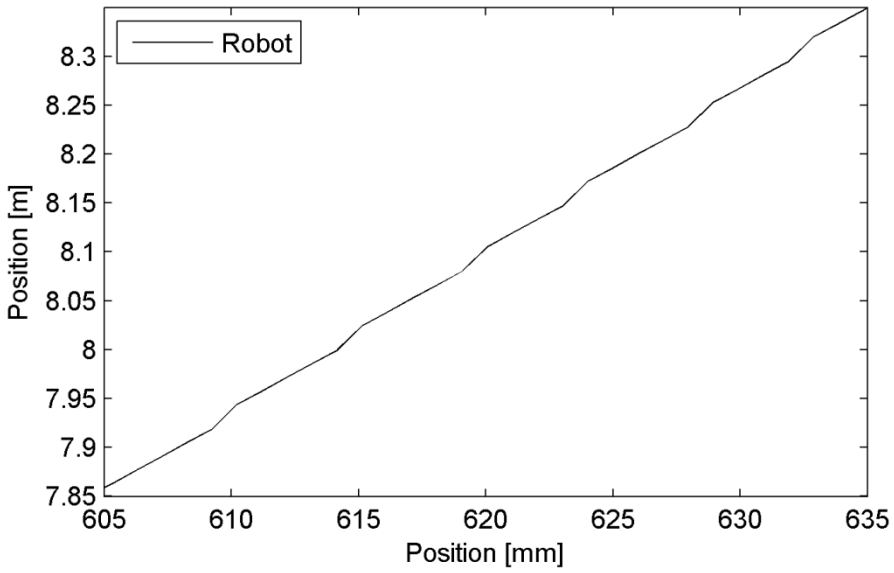


(c)

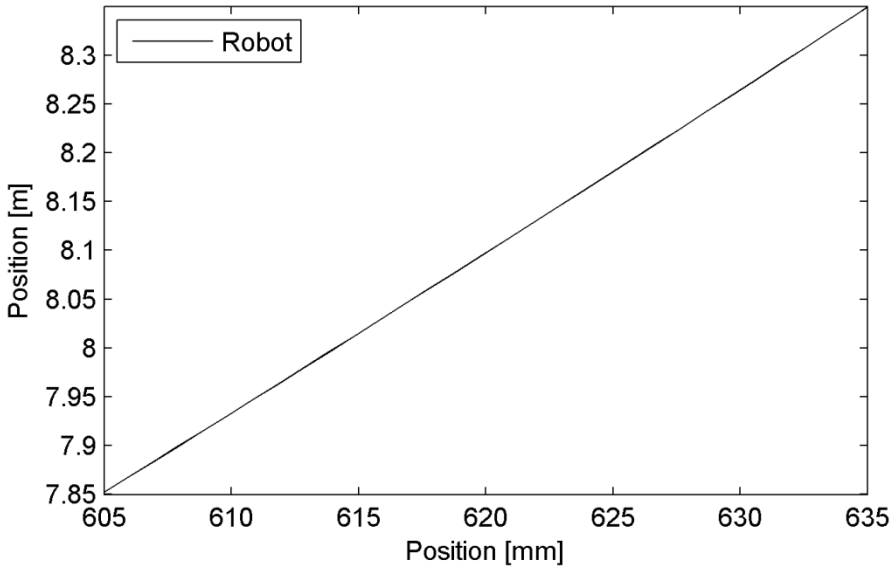


(d)

Fig. 7.4: System performance with curved wall using (a) BBC, (b) SNN, (c) CFBC, and (d) RFBC.



(a)



(b)

Fig. 7.5: Zoomed in comparison between (a) CFBC and (b) RFBC.

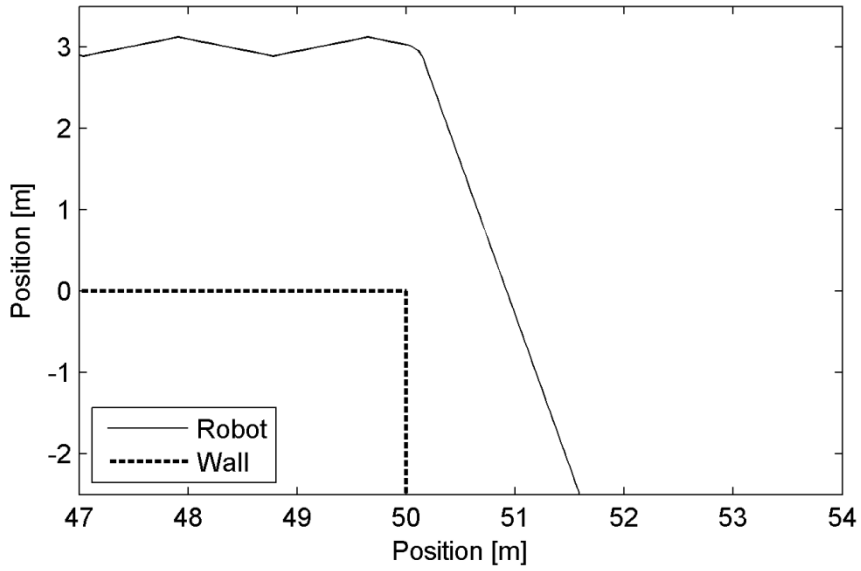
7.1.3 Wall with Corner Simulation

The simulation using a wall with 90 degree corner is subdivided into two conditions, which are a convex corner shown in Fig. 7.6 and a concave corner shown in Fig. 7.7. The control results are listed in Table 7.3. This comparison focuses on the turning performance of selected algorithms. The transient-state when a mobile robot encounters a convex corner is shown in Fig. 7.6, the BBC (Fig. 7.6(a)) approaches the desired distance with the lowest rate. The results can be improved by adjusting parameters (*e.g.,* modify the pre-defined heading angle), but fierce heading direction change may bring more unstable factors. SNN (Fig. 7.6(b)), CFBC (Fig. 7.6(c)), and RFBC (Fig. 7.6(d)) demonstrate similar turning behavior, but SNN does not exhibit optimal approach behavior after passing the convex corner. The trajectory of RFBC is more flat during turning, and RFBC slightly improves the overshoot minimization compared with the CFBC.

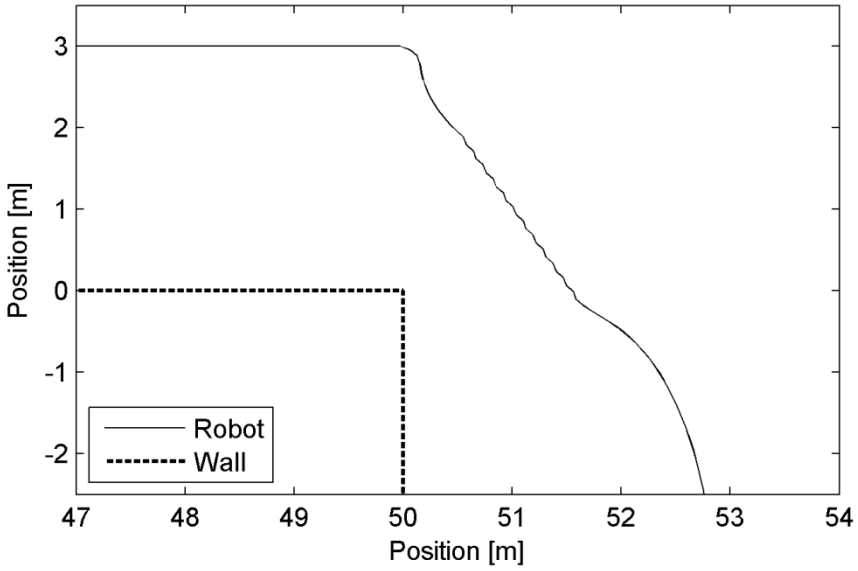
As shown in Table 7.3, the similar turning behaviors of SNN, CFBC, and RFBC is reflected in the entropy measures. The advantage of RFBC results from the improved approaching behavior, and it enters the steady-state earlier than CFBC and SNN. In the conditional entropy measures, the high control accuracy of RFBC can be clearly observed. The drawback of SNN (*i.e.,* lower response rate to environmental change) causes obvious difference in control accuracy. When compared with CFBC, the RFBC enhances uncertainty reasoning in a situation that the orientation changing rate of a wall is high. The goal of overshoot minimization and transient-state optimization is primarily achieved.

Table 7.3: Simulation data using a 90 degree corner.

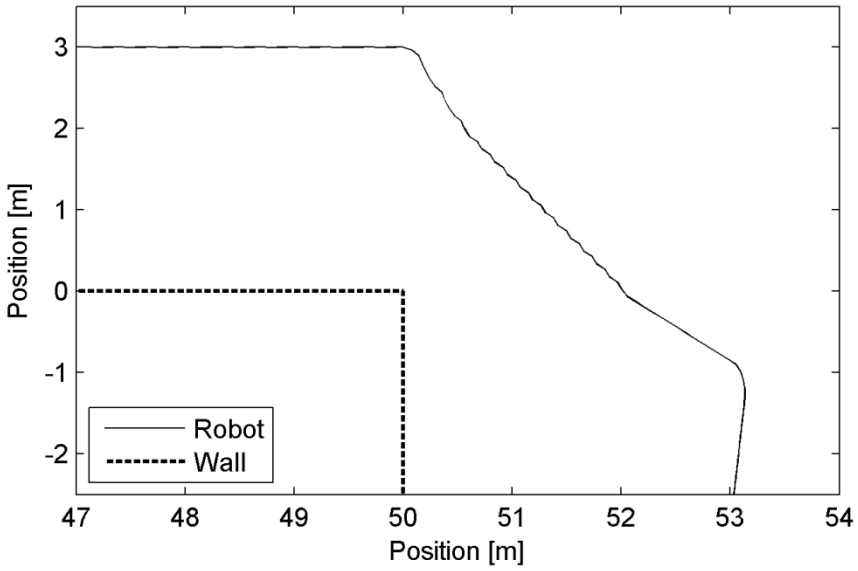
	Mean [m]	Variance [m]	Entropy [bit]	Conditional Entropy [bit]
Convex Corner				
BBC	2.8013	0.2516	2.6777	0.1515
SNN	2.9000	0.1034	1.7919	0.0537
CFBC	2.9296	0.0800	1.5527	0.0285
RFBC	2.9521	0.0725	1.2736	0.0115
Concave Corner				
BBC	2.9267	0.1040	2.0691	0.1515
SNN	2.9583	0.0408	1.3601	0.0496
CFBC	2.9672	0.0381	1.0559	0.0286
RFBC	2.9652	0.0392	1.3159	0.0186



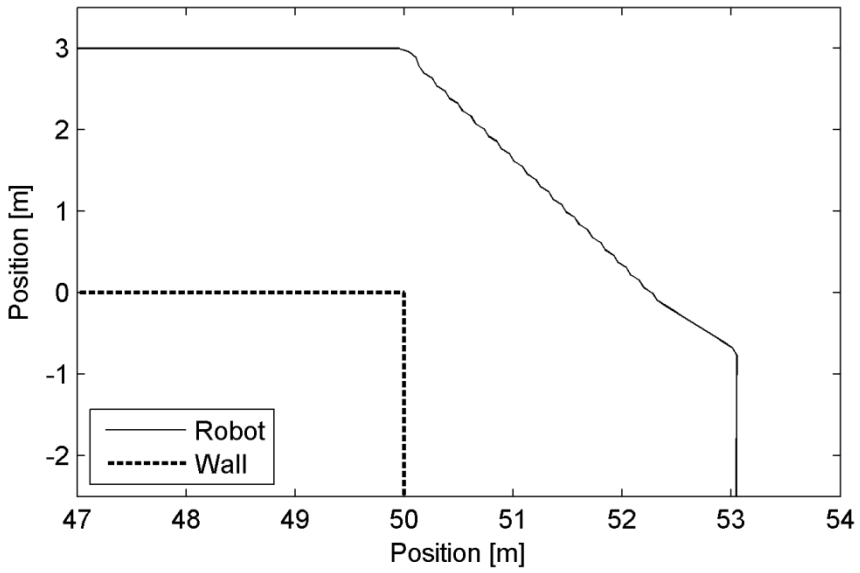
(a)


Position [m]
Position [m]
Robot
Wall
3
2
1
0
-1
-2
47
48
49
50
51
52
53
54

(b)


Position [m]
Position [m]
Robot
Wall
3
2
1
0
-1
-2
47
48
49
50
51
52
53
54

(c)

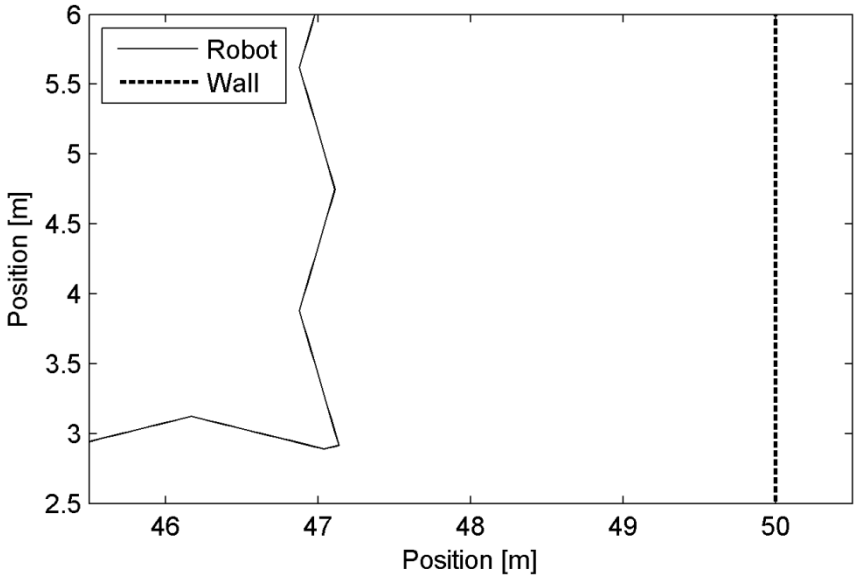


(d)

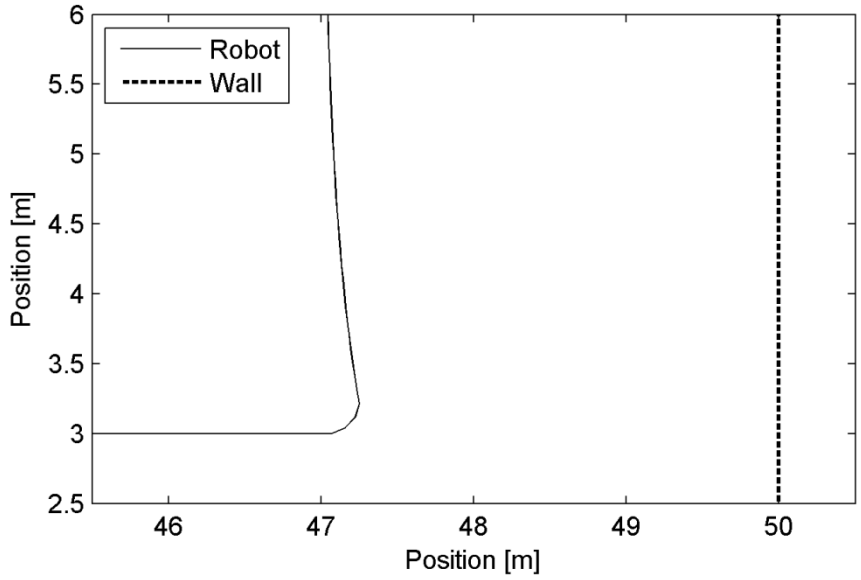
Fig. 7.6: Zoomed in transient-state comparison of (a) BBC, (b) SNN, (c) CFBC, and (d) RFBC at a 90 degree convex corner.

The transient-state when a mobile robot encounters a concave corner is shown in Fig. 7.7. The BBC (Fig. 7.7(a)) illustrates too much trajectory fluctuation, which implies the reaction to environmental change is too strong. The SNN (Fig. 7.7(b)) performs smoother turning results than CFBC ((Fig. 7.7(c))). Compared with the SNN, the proposed RFBC (Fig. 7.7(d)) further optimizes the turning performance. However, the convergence rate of RFBC is not as good as CFBC and SNN. It results from imprecise pattern classification, which means the generation of equivalence classes is not precise enough. In the appendix C, an overall trajectory figure shows it converges to the desired distance with lower rate than CFBC and SNN. Therefore, proper equivalence relation rules are important to the RFBC design and this could be one of the future research goals.

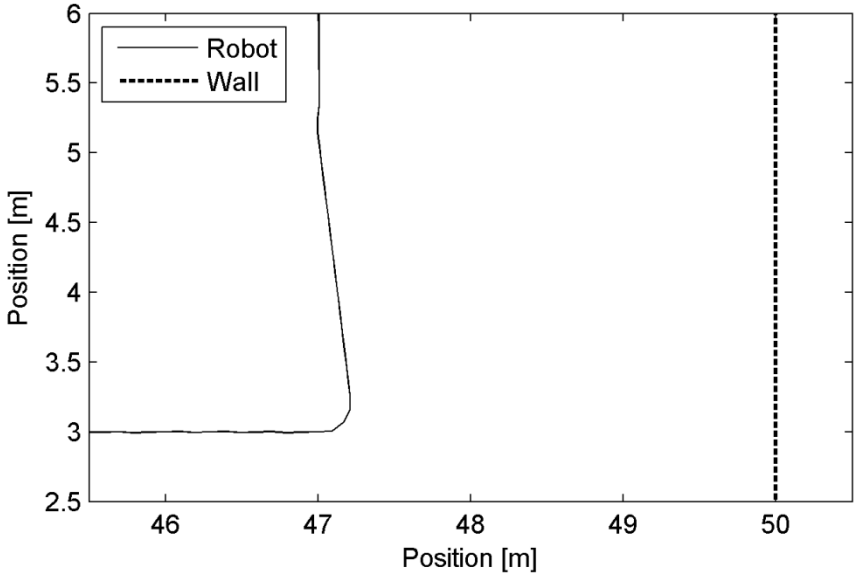
In Table 7.3, the RFBC is not the best in terms of overall performance. Though the conditional entropy shows that its control accuracy is still stable, the convergence rate of RFBC drops once the mobile robot is close enough to the desired path. From a mathematical point of view, the property of an equivalence class makes both the measured distance and the measured orientation difference indiscernible with their desired values. As a result, the selection of fuzzy rules is not optimized. On the one hand, it makes a mobile robot steady especially while it is close to the desired path since the small fluctuation (*e.g.,* Fig. 7.2) around the desired path can be minimized. On the other hand, the dramatic convergence rate decrease indeed lowers the control accuracy. This issue can be improved through further following behaviour analysis in order to refine equivalence relation rules.



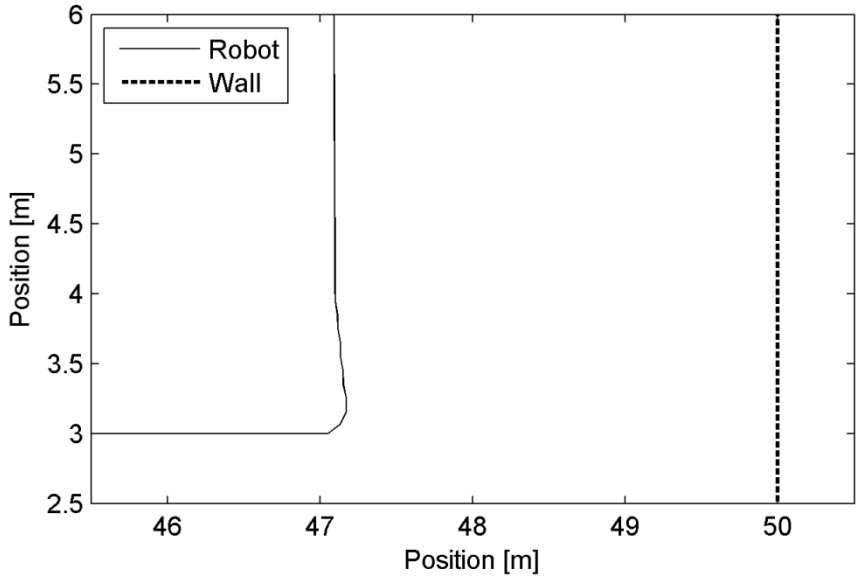
(a)


Robot
Wall
Position [m]
Position [m]
6
5.5
5
4.5
4
3.5
3
2.5
46
47
48
49
50

(b)


Robot
Wall
Position [m]
Position [m]
6
5.5
5
4.5
4
3.5
3
2.5
46
47
48
49
50

(c)



(d)

Fig. 7.7: Zoomed in transient-state comparison of (a) BBC, (b) SNN, (c) CFBC, and (d) RFBC at a 90 degree concave corner.

7.2 Experiment Results

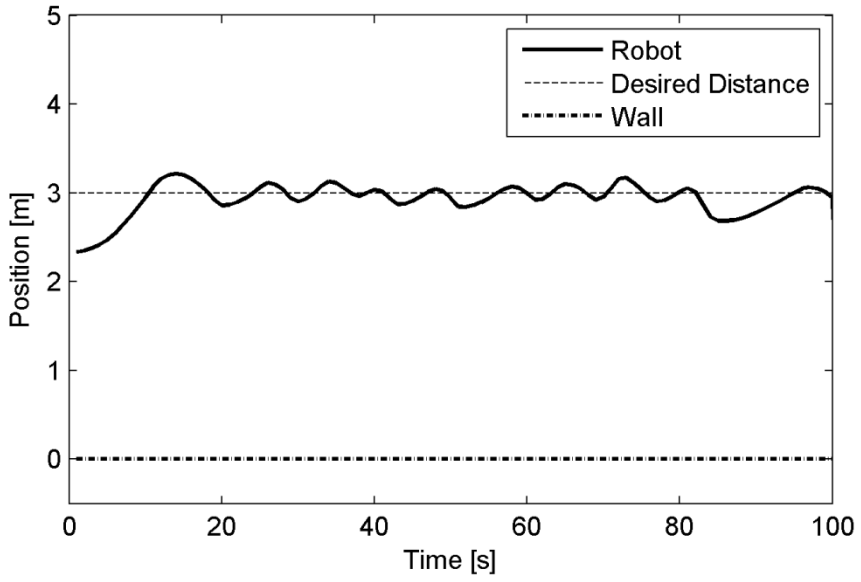
Compared with the simulation, the experiment environment contains much more noise. In addition, the practical kinematic behaviour of a mobile robot is not as ideal as it is in the simulation. However, the similar pattern of each algorithm still can be identified through charts. The actual desired distance between a mobile robot and the wall is 0.9 meter. For the purpose of better illustration, the desired distance of experiment is magnified to be equal to simulation setup. The goal is to verify the robustness of the proposed RFBC in the practical environment, and observe the performance against noisy sensory data.

7.2.1 Straight Wall Experiment

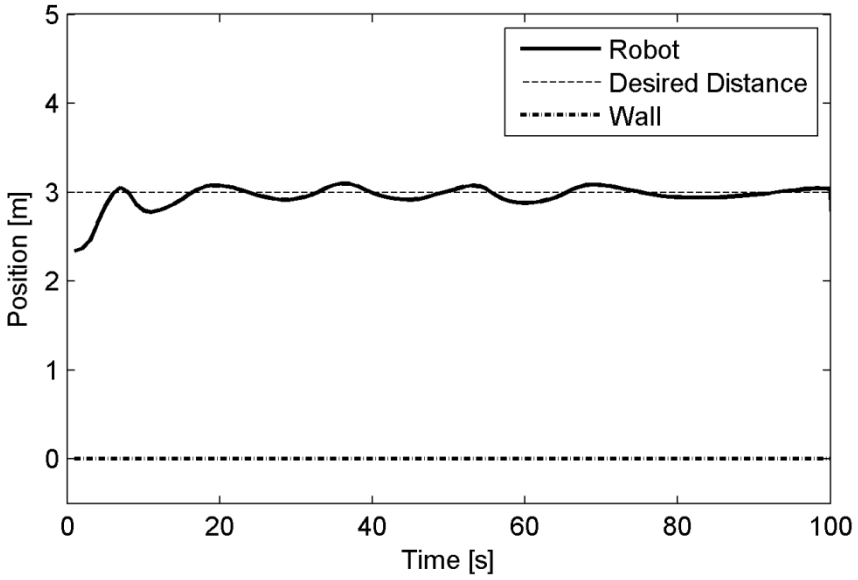
The experiment results of a straight wall are shown in Fig. 7.8 and Table 7.4. Obviously, the BBC (Fig. 7.8(a)) result is not competitive with other algorithms. The SNN (Fig. 7.8(b)) significantly minimized the overshoot, but the response rate to distance variation is slower than CFBC and RFBC. It is reflected in the slightly larger fluctuation amplitude in the experiment. The CFBC (Fig. 7.8(c)) and RFBC (Fig. 7.8(d)) improved stability with respect to desired distance keeping. Compared with the CFBC, the RFBC further enhanced the stability in terms of fluctuation frequency.

Table 7.4: Experiment data using a straight wall.

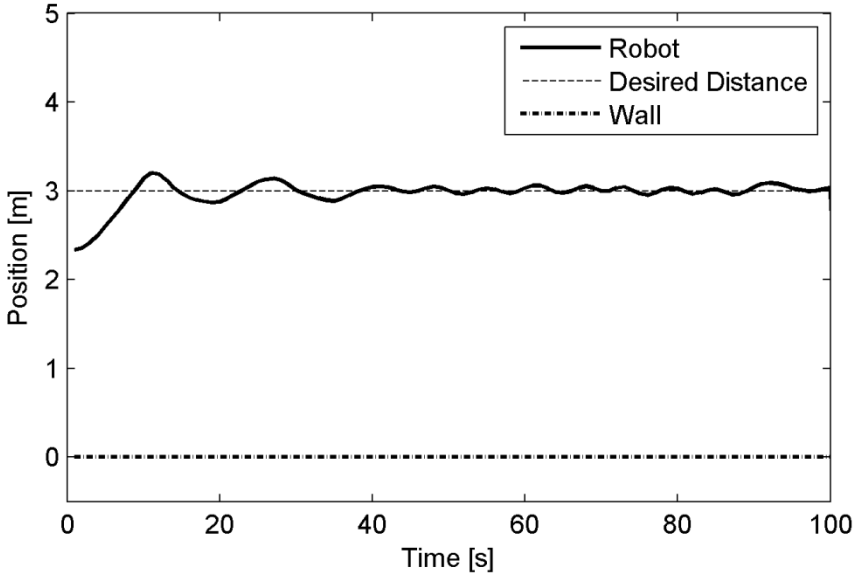
Straight Wall	Mean [m]	Variance [m]	Entropy [bit]	Conditional Entropy [bit]
BBC	2.9359	0.0322	2.6023	0.3510
SNN	2.9577	0.0162	1.7584	0.1618
CFBC	2.9753	0.0208	2.0186	0.2102
RFBC	2.9762	0.0200	1.5522	0.1236



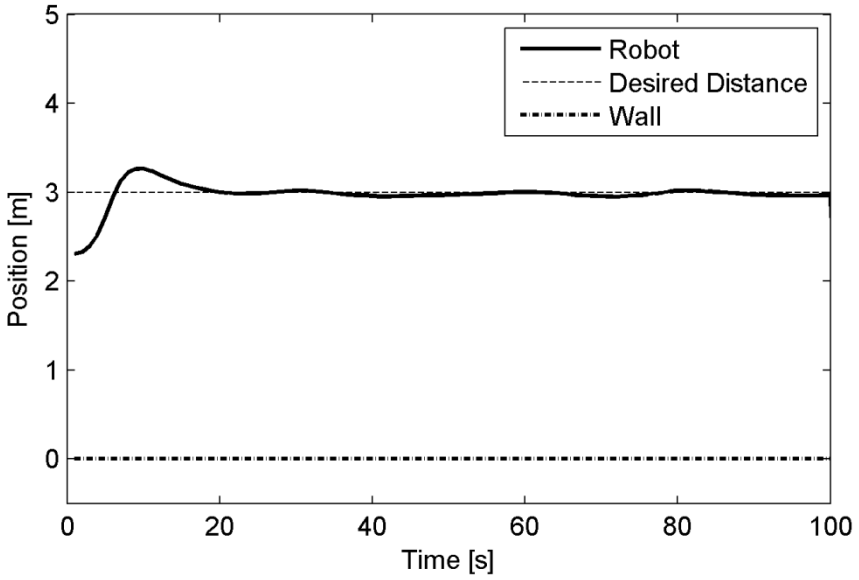
(a)


Robot
Desired Distance
Wall
Position [m]
Time [s]
0
1
2
3
4
5
0
20
40
60
80
100

(b)


Robot
Desired Distance
Wall
Position [m]
Time [s]
0
1
2
3
4
5
0
20
40
60
80
100

(c)



(d)

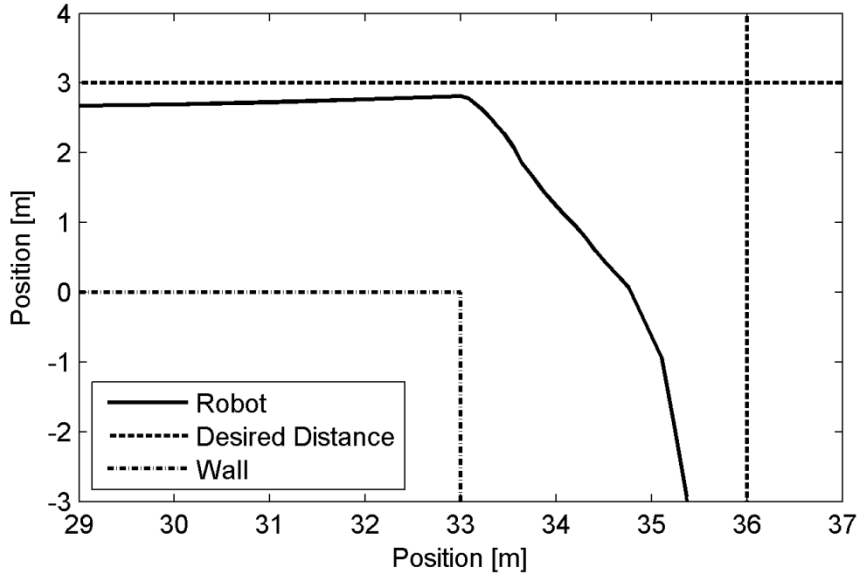
Fig. 7.8: Straight wall experiment comparison of (a) BBC, (b) SNN, (c) CFBC, and (d) RFBC.

7.2.2 Wall with Corner Experiment

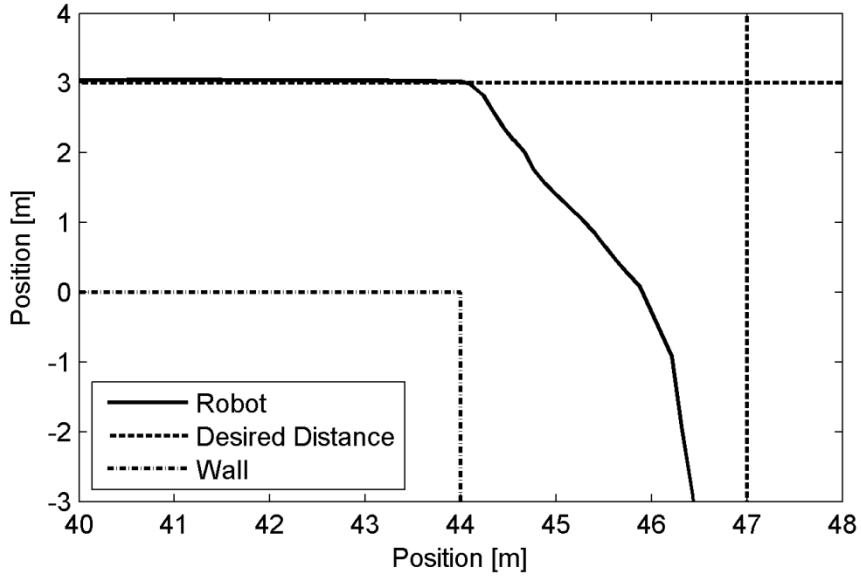
The experiment results of a convex corner wall are shown in Fig. 7.9 and Table 7.5. Prior to encounter a convex corner, all control algorithms except BBC (Fig. 7.9(a)) performs well in terms of desired distance keeping. The turning performance of BBC and SNN (Fig. 7.9(b)) exhibits smooth curve. However, compared with the CFBC (Fig. 7.9(c)) and RFBC (Fig. 7.9(d)), the second half of turning period illustrates lower convergence rate. As a result, the CFBC and RFBC locked at the desired distance faster than the BBC and SNN. As observed, the overshoot of RFBC is slightly minimized compared with the CFBC.

Table 7.5: Experiment data using a 90 degree convex corner.

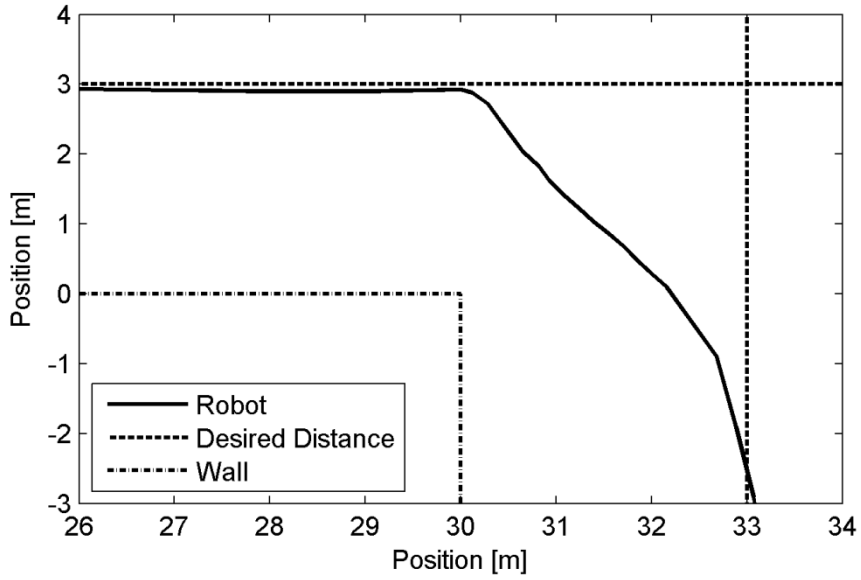
Convex Corner	Mean [m]	Variance [m]	Entropy [bit]	Conditional Entropy [bit]
BBC	2.8896	0.2319	3.0715	0.2146
SNN	2.8336	0.1772	2.4836	0.1832
CFBC	2.9439	0.1935	2.9222	0.0824
RFBC	2.8923	0.1957	2.5042	0.1347



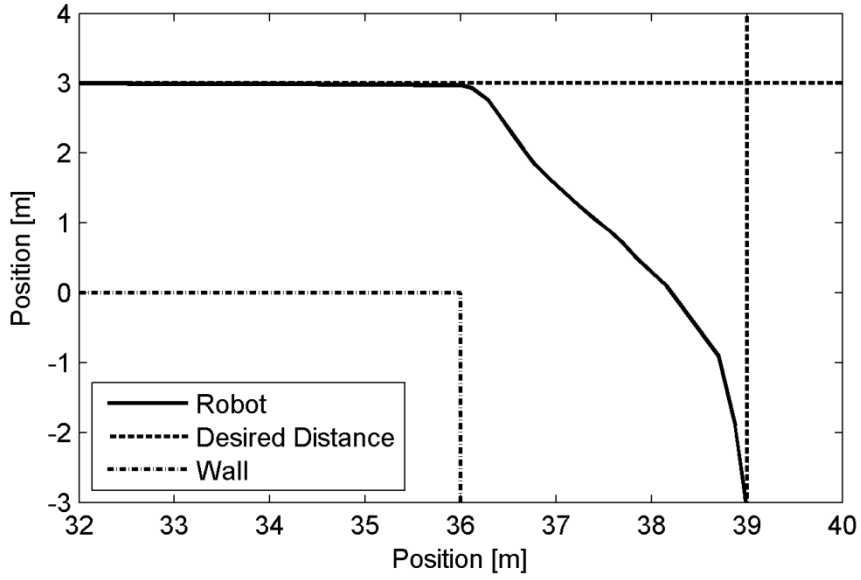
(a)



(b)



(c)



(d)

Fig. 7.9: Experiment results of a 90 degree convex corner using (a) BBC, (b) SNN, (c) CFBC, and (d) RFBC.

7.3 Summary

This chapter presents simulations and experiments used to test the performance of the RFBC and other comparison algorithms. The simulation is considered as a known environment without noise, and its goal is to test the optimal performance and validate the theory of RFBC. The experiment is considered as an unknown environment with much more noise and uncertainty. Its goal is to test the robustness of RFBC in practical environments, and provides some comparisons with simulation results to verify the algorithm efficiency.

Chapter 8

CONCLUSIONS

8.1 Overview

This thesis presents a new perception-based computing for vision-based indoor navigation algorithm. An indoor environment is defined by a complex layout in a compact space, and indoor navigation behaviour is summarized as a pattern classification problem. The perception-based computing is used to improve uncertainty reasoning results as described in Ch. 2. A combination of rough and fuzzy set theories is proposed as a new implementation approach for perception-based computing. This is supported through a study in Ch. 3 of rough set and fuzzy set with respect to individual features and analytical comparison. In Ch. 7, simulations and experiments are conducted to compare capabilities of navigation algorithms. The uncertainty improvement is analyzed using information measure. The results indicate the RFBC is able to minimize overshoot and optimize transient-state period while follows a given wall. The response rate and turning performance is optimized so that unnecessary trajectories are avoided. The stability of the RFBC is enhanced compared with common control algorithms.

8.2 Thesis Conclusions

This thesis attempts to answer many interesting research questions about the vision-based autonomous indoor navigation as outlined in Sec. 1.4.3. This section concludes answers through experiment results and observations.

An indoor environment is defined as a complex layout in a compact space. It leads to a number of issues that need to be properly handled. A rapid response to environmental change is one of primary evaluation criteria due to the irregularity of space configuration, and a stable behaviour is also critical since the indoor space could be too narrow to allow imprecise displacement. Furthermore, collected sensory data may not be reliable and be able to reflect the situation correctly. The contours of objects like walls and furniture are the major reference coordinate while a mobile robot navigates in such environment. The importance of a wall-following behaviour for the indoor navigation is emphasized in Sec. 2.4. Consequently, an indoor navigation is described as a wall-following behaviour in a restricted and noisy environment, as well as limited and partial knowledge. In order to achieve rapid response and stable control, the efficiency of control algorithms should be improved with respect to computing approach. In other words, the algorithm should not only compute with a huge amount of sensory data, but also perceive information from the environment so that be able to interact optimally.

Prior to a study of computing method, an efficient way of collecting data is discussed. Motivation of vision sensors and drawbacks of other sensors in terms of the specific environment are reviewed in Sec. 2.3. An indoor navigation algorithm is able to

perceive more accurate information of the environment based on visual data. Furthermore, a systematic analysis of the influence of environment noise and uncertainty situations is elaborated. In this case, this thesis then evaluates current control algorithms using findings related to indoor navigation problems. The advantages and disadvantages of reviewed navigation algorithms could be identified, and a perception-based computing approach is proposed to improve these drawbacks.

To achieve a human-centric design for the indoor navigation, a particular form named perception is introduced to represent the interaction of an agent and its environment. The proposed approach aims to improve control performance by grasping information embedded in the sensory data. It is designed to analyze and summarize exiting experience of the environment so that the machine is able to interact with the environment more efficiently. Furthermore, a widely accepted conventional fuzzy based implementation method is described. The features and performance is discussed through mathematics and previous works. A CFBC is easy to implement for a non-linear system and has the capability of stabilizing navigation trajectory in a highly uncertainty environment. However, the CFBC lacks the ability of stabilizing antecedent variables (*i.e.,* measured data) before applying fuzzy inference rules. This is important since sensory data could contain errors in a number of situations, and vagueness may exist between real-time inputs and pre-defined fuzzy rule base. It is said that the performance of CFBC can be further improved if antecedent variables are more reliable. For this reason, a rough set theory is introduced to regulate sensory inputs.

The rough set theory shows its unique ability of characterizing elements in a set by indiscernible feature values. By applying proper equivalence relation rules, each sensory input is classified into corresponding equivalence class according to a short-term behaviour analysis. An equivalence class is an information granule that independently represents a specific situation of the environment. In this case, the input is redressed based on features of this particular equivalence class. In other words, since equivalence rules are induced from existing experience, and equivalence classes created from such rules is able to lower vagueness and uncertainty of a real-time sensory data. A rough-fuzzy membership function is introduced so that an appropriate fuzzy inference rule to the particular equivalence class is selected.

In addition to traditional evaluation methods, in order to measure the information refinement degree of perceptual computing algorithms, an entropy based information measure is introduced (see Sec. 3.7). If the perception of the environment is properly conducted during the navigation, the uncertainty of control outputs should be low. It implies that a mobile robot perceives accurate information of the situation, and outputs would not have too much surprise, which means values are most likely to fall into a specific interval. Furthermore, the patterns of a navigation algorithm are able to be extracted and perturbed data are identified through multi-scale analysis.

The proposed RFBC is tested through simulations and experiments. An indoor environment in the simulation is considered as a known situation without noise, and knowledge about the environment is sufficient (as stated in Sec. 2.4.2). Compared with

the simulation, the indoor environment in the experiment is considered as an unknown situation (as stated in Sec. 2.4.1), and it contains much more noise and uncertainty.

8.3 Contributions

This thesis contributes to systematic analysis to the performance and behaviour of indoor navigation algorithm. A new implementation approach of perception-based computing is proposed. These contributions are outlined below.

(a) A structural analysis of the indoor navigation is stated and a review of current control algorithms is discussed. In this thesis, an indoor navigation task is decomposed into several important parts. Each component is discussed with respect to influence to the navigation behaviour. The analysis focuses on summarization of features for an indoor navigation task, and the factors that constitute to an efficient indoor navigation algorithm.

(b) A new perception-based computing is proposed to optimize drawbacks of current control algorithms. A review of widely accepted control algorithms is conducted, and the features and drawbacks are discussed. The importance of an information oriented control algorithm against uncertainty environment is emphasized. The reason of advantages of perception-based computing over numerical computing is explained. The perception-based computing demonstrates high efficiency with respect to uncertainty environment.

(c) An entropy-based information measure method is introduced to evaluate perceptual algorithms. The motivation of information measure is expressed and the detailed methods to apply entropy-based measure to performance evaluation are described. Furthermore, multi-scale analysis is utilized to extract behaviours from navigation performance, and perturbed data could be identified using learning entropy. The entropy-based measure method shows unique property on evaluating uncertainty of control outputs using distinct algorithm.

(d) An interaction model is designed to facilitate the implementation of CFBC and RFBC. In order to mathematically describe an indoor navigation, an interaction model of a mobile robot and its environment is created. In this case, sensory data and extracted information is properly grouped and unnecessary data is filtered out. Consequently, it speeds up the implementation of CFBC and RFBC.

(e) A new rough-fuzzy implementation approach of perception-based computing is presented. The algorithm is proposed in a hierarchical fashion starting from fuzzy and rough set theory, and then the feasibility of combing two theories for indoor navigation is discussed. The advantage of rough set theory is beneficial to the sensory input regulation. Furthermore, the CFBC has been demonstrated its ability of uncertainty reasoning through a fact of well adapted. The RFBC is designed to further improve CFBC performance through minimization of overshoot and optimization of transient-state period, which are extremely important for an efficient indoor navigation algorithm. The performance of proposed RFBC is verified through simulations and experiments.

8.4 Future Work

Direct extension of the work presented in this thesis is divided into two categories: (i) autonomous multi-scale analysis, and (ii) field-programmable gate array implementation.

A rough set theory is introduced to generate knowledge granules that mimic human sense to acquired environmental information. By far, granular computing is more a theoretical perspective than a coherent set of methods and principles. A multi-scale analysis approach is introduced as an implementation attempt of granular computing.

In this thesis, the multi-scale analysis is conducted by human experts with sufficient experience. In an ideal case, a mobile robot is required to recognize encountered situations continuously. In other words, knowledge granules are created and updated throughout the navigation. Consequently, a mobile robot should be able to manipulate knowledge granules autonomously. In order to improve navigation performance, a possible solution could be raised with respect to autonomous conduction of multi-scale analysis.

From an experimental perspective, a portable computer could be replaced by a device so that it is able to be integrated into a mobile robot. Such a device is designed to improve the mobility and computational efficiency during navigation. In order to fulfill the goal, an integrated circuit named *field-programmable gate array* (FPGA) could be used [GoNa13]. This device is designed to be configured by a customer or a designer after manufacturing for a specific purpose. In general, a FPGA has large resources of logic gates and RAM blocks to implement complex digital computations. Due to the

flexibility and increasing computing power, FPGAs have been used extensively in different fields. Therefore, it is possible to enhance the experimental performance via a FPGA implementation.

REFERENCES

[AmCh10] A. Amanatiadis, D. Chrysostomou, D. Koulouriotis, and A. Gasteratos, "A fuzzy multi-sensor architecture for indoor navigation," *in Proc. IEEE Int. Conf. Imaging Systems and Techniques*, (Thessaloniki, Greece; July 1-2, 2010), pp. 452-457, 2010.

[Arau06] Ernesto Araujo, "A fuzzy purpose-in-life perspective for artificial intelligence, robotics and computational intelligence," *in Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, (Taipei, China; October 8-11, 2006), pp. 132-136, 2006.

[ArCa12] Juan Antonio Arizaga, Jorge de la Calleja, Roberto Hernandez, and Antonio Benitez, "Automatic control for laboratory sterilization process based on Arduino hardware," *in Proc. IEEE Int. Conf. Electrical Communications and Computers*, (Cholula, Puebla; February 27-29, 2012), pp. 130-133, 2012.

[Ardi10] Igi Ardiyanto, "Task oriented behaviour-based state-adaptive PID control for low-cost mobile robot," *in Proc. Int. Conf. Computer Engineering and Applications*, (Bali Island, Indonesia; March 19-21, 2010), vol. 1, pp. 103-107, 2010.

[ArPo13] Andre Araujo, David Portugal, Micael S. Couceiro, and Rui P. Rocha, "Integrating Arduino-based educational mobile robots in ROS," *in Proc. IEEE Int. Conf. Autonomous Robot Systems*, (Lisbon, Portugal; April 24, 2013), pp. 1-6, 2013.

[Babu09] Robert Babuska, *Fuzzy and Neural Control*. DISC Course Lecture Notes. Delft, the Netherlands: Delft University of Technology, Nov 2009, 216 pp.

[Bela06] Mohamed A. Belal, *A Foundation for Perception Computing, Logic, and Automata*. Technical Report. Ithaca, New York, USA: Cornell University, Jul 2006, 39 pp.

[BuKB12] Ivo Bukovsky, Witold Kinsner, and Jiri Bila, "Multiscale analysis approach for novelty detection in adaptation plot," in *Proc. 3rd 2012 IEEE Sensor Signal Processing for Defence, SSPD 2012* (Imperial College, London, UK; September 25-27, 2012), pp. 1-6, 2012.

[BuKo13] Ivo Bukovsky, "Learning entropy: Multiscale measure for incremental learning," *Entropy*, vol. 15, no. 10, pp. 4159-4187, Oct 2013.

[Busa12] Asiya M. Al-Busaidi, "Development of an educational environment for online control of a biped robot using MATLAB and Arudino," *in Int. Workshop on Mechatronics*, (Paris, France; November 21-23, 2012), pp. 337-344, 2012.

[BuJa10] Widodo Budiharto, Achmad Jazidie, and Djoko Purwanto, "Indoor navigation using adaptive neuro fuzzy controller for servant robot," *in Proc. IEEE Int. Conf. Computer Engineering and Applications*, (Bali Island, Indonesia; March 19-21, 2010), vol. 1, pp. 582-586, 2010.

[ChHu06] F. Chaumette and Seth Hutchinson, "Visual servo control part 1: Basic approaches," *in IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82-90, Dec 2006.

[ChHu07] F. Chaumette and Seth Hutchinson, "Visual servo control part 2: Advanced approaches," *in IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109-118, Mar 2007.

[ChHu08] F. Chaumette and Seth Hutchinson, "Visual servoing and visual tracking," *in Springer Handbook of Robotics*, pp 563-583. New York, NY: Springer, 2008, 1611 pp.

[CoSc12] Diogo Santos Ortiz Correa, Diego Fernando Sciotti, Marcos Gomes Prado, Daniel Oliva Sales, Denis Fernando Wolf, and Fernando Santos Osorio, "Mobile robots navigation in indoor environment using Kinect sensor," *in Proc. Second Brazil Conf. Critical Embedded Systems*, (Campinas, Brazil; May 20-25, 2012), pp. 36-41, 2012.

[DaYo03] Nguyen Xuan Dao, Bum-Jae You, Sang-Rok Oh, and Myung Hwangbo, "Visual self-localization for indoor mobile robots using natural lines," *in Proc. IEEE Int. Conf. Intelligent Robots and Systems*, (Las Vegas, Nevada; October 27-31, 2003), vol. 2, pp. 1252-1257, 2003.

[DeKa02] Guiherme N. DeSouza and Avinash C. Kak, "Vision for mobile robot navigation: A survery," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237-267, Feb 2002.

[Dhar13] Mamoni Dhar, "Cardinality of fuzzy sets: An overview," *Int. J. Energy, Information and Communications*, vol. 4, no. 1, pp. 15-22, Feb 2013.

[DiSJ06] Shifei Ding, Zhongzhi Shi, and Fengxiang Jin, "Studies on fuzzy information measure," *in Proc. Int. Conf. Fuzzy Systems and Knowledge Discovery*, (Haikou, China; August 24-27, 2006), vol. 3, pp. 376-380, 2006.

[EsVa04] Jairo Espinosa, Joos Vandewalle, and Vincent Wertz, *Fuzzy Logic, Identification and Predictive Control*. New York, NY: Springer, 2005, 273 pp.

[FaKi05] A. Faghfouri, Witold Kinsner, and D. Swatck, "Comparison of entropy-based characterization of lightning strike maps using planar and spherical coordinates," *in Proc. Canadian Conf. Electrical and Computer Engineering*, (Saskatoon, Canada; May 1-4, 2005), pp. 1918-1922, 2005.

[Fara12] Ramsey Faragher, "Understanding the basis of the Kalman filter via a simple and intuitive derivation," *in Mag. IEEE Signal Processing*, vol. 29, no. 5, pp. 128-132, Sep 2012.

[FlEp05] Dario Floreano, Yann Epars, Jean-Christophe Zufferry, and Claudio Mattiussi, "Evolution of spiking neural circuits in autonomous mobile robots," *Int. J. Intelligence Systems*, vol. 21, no. 9, pp. 1005-1024, Sep 2006.

[Gers09] L. Gerstmayr, *Image-Based Visual Servoing*. Lecture Notes. Bielefeld, Germany: University of Bielefeld, 2009, 40 pp.

[GhHa04] Masoud Ghaffari and Ernest L. Hall, "Robotics and nature, from primitive creatures to human intelligence," *in Proc. the Int. Society for Optics and Photonics*, (Philadelphia, PA; October 25, 2004), vol. 5608, pp. 169-176, 2004.

[GoNa13] Jesus D. Terrazas Gonzales, Mohamed Temam Nasri, Tong Duan, and Witold Kinsner, "A datacomm laboratory with FPGAs in a microprocessor interfacing course" *in Proc. the Canadian Engineering Education Association Conference, CEEA 2013*, (Montreal, Canada; June 17-20, 2013), pp.51-54, 2013.

[Hart28] R. V. L. Hartley, "Transmission of information," *Bell Systems Technical J.*, pp. 535-563, 1928.

[Hayk05] Simon Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Selected Areas in Communications*, vol. 23, no. 2, pp. 201-220, Feb 2005.

[HuHC96] Seth Hutchinson, Gregory D. Hager, and Peter I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robotics and Automation*, vol. 12, no. 5, pp. 651-670, Oct 1996.

[JaMM96] Anil K. Jain, Jianchang Mao, and K. Mohiuddin, "Artificial neural networks: A tutorial," *in IEEE Computer*, vol. 29, no. 3, pp. 31-44, Mar 1996.

[JaSk08] Andrzej Jankowski and Andrzej Skowron, "Toward perception based computing: A rough-granular perspective," *in Web Intelligence Meets Brain Informatics*, pp. 122-142. New York, NY: Springer, 2007, 515 pp.

[JiBC06] Zhen Jia, Arjuna Balasuriya, and Subhash Challa, "Recent developments in vision based target tracking for autonomous vehicles navigation," *in Proc. IEEE Conf. Intelligent Transportation Systems*, (Toronto, Canada; September 17-20, 2006), pp. 765-770, 2006.

[JuLu13] Hau-Shiue Juang and Kai-Yew Lum, "Design and control of a two-wheel self-balancing robot using Arduino microcontroller board," *in Proc. IEEE int. Conf. Control and Automation*, (Hangzhou, China; June 12-14, 2013), pp. 634-639, 2013.

[KaEG12] Florian Kamergruber, Andreas Ebner, and Willibald A. Gunthner, "Navigation in virtual reality using Microsoft Kinect," *in Proc. Int. Conf. Construction Application of Virtual Reality*, (Taipei, China; November 1-2, 2012), pp. 350-359, 2012.

[KaKa12] Arshdeep Kaur and Amrit Kaur, "Comparison of mamdani-type and sugeno-type fuzzy inference systems for air conditioning system," *Int. J. Soft Computing and Engineering*, vol. 2, no. 2, pp. 323-325, May 2012.

[KaKh11] A. Karambakhsh, M. Yousefi Azar Khanian, M. R. Meybodi, and A. Fakharian, "Robot navigation algorithm to wall following using fuzzy Kalman filter," *in Proc. IEEE Int. Conf. Control and Automation*, (Santiago, Chile; December 19-21, 2011), pp. 440-443, 2011.

[Kawa91] Shigeki Ishikawa, "A method of indoor mobile robot navigation by using fuzzy control," *in IEEE Int. Workshop on Intelligent Robots and Systems*, (Osaka, Japan; November 3-5, 1991), vol. 2, pp. 1013-1018, 1991.

[KeMi12] Ho Wei Keat and Low Sew Ming, "An investigation of the use of Kinect sensor for indoor navigation," *in Proc. IEEE Region 10 Conf. TENCON*, (Cebu, Philippines; November 19-22, 2012), pp. 1-5, 2012.

[KiDa06] Witold Kinsner and R. Dansereau, "A relative fractal dimension spectrum as a complexity measure," *in Proc. IEEE Int. Conf. Cognitive Informatics*, (Beijing, China; July 17-19, 2006), pp. 200-208, 2006.

[KiLy09] Taeyeon Kim and Joon Lyou, "Indoor navigation of skid steering mobile robot using ceiling landmarks," *in IEEE Int. Symposium on Industrial Electronics*, (Seoul, Korea; July 5-8, 2009), pp. 1743-1748, 2009.

[Kins04] Witold Kinsner, "Is entropy suitable to characterize data and signals for cognitive informatics?" *in Proc. IEEE Int. Conf. Cognitive Informatics*, (Victoria, Canada; August 16-17, 2004), pp. 6-21, 2004.

[Kins05] Witold Kinsner, "A unified approach to fractal dimensions," *in Proc. IEEE Int. Conf. Cognitive Informatics*, (Irvine, USA; August 8-10, 2005), pp. 58-72, 2005.

[Kins07] Witold Kinsner, "Challenges in the design of adaptive, intelligent and cognitive systems," *in Proc. IEEE Int. Conf. Cognitive Informatics*, (Lake Tahoo, CA; August 6-8, 2007), pp. 13-25, 2007.

[Kins13] Witold Kinsner, *Fractal and Chaos Engineering*. Lecture Notes. Winnipeg, Canada: University of Manitoba, January 2013, 900pp.

[KiSo09] Kimcheng Kith, Olga Sourina, Vladimir Kulish, and Nguyen Minh Khoa, "An algorithm for fractal dimension calculation based on Renyi entropy for short time signal analysis," *in Proc. Int. Conf. Information, Communications and Signal Processing*, (Macau, China; December 8-10, 2009), pp. 1-5, 2009.

[Klee07] Lindsay Kleeman, *Understanding and Applying Kalman Filter*. Lecture Notes. Clayton, Australia: Monash University Clayton, July 2007, 37 pp.

[KuEk13] Laszlo Kundra, and Peter Ekler, "The summary of indoor navigation possibilities considering mobile environment," *in Proc. Eastern European Regional Conf. Engineering of Computer Based Systems*, (Budapest, Hungary; August 29-30), pp. 165-166, 2013.

[LeBZ06] Chang Su Lee, Thomas Braunl, and Anthony Zaknich, "A rough-fuzzy controller for autonomous mobile robot navigation," *in Proc. IEEE Int. Conf. Intelligence Systems*, (London, UK; September 4-6, 2006), pp. 679-682, 2006.

[LeCJ13] Carson Kai-Sang Leung, Alfredo Cuzzocrea, and Fan Jiang, "Discovering frequent patterns from uncertain data streams with time-fading and landmark models," *Trans.*

Large-Scale Data and Knowledge Centered Systems, vol. 7790, no. 12, pp. 174-196, 2013.

[LeJH11] Carson Kai-Sang leung, Fan Jiang, and Yaroslav Hayduk, "A landmark-model based system for mining frequent patterns from uncertain data streams," *in Proc. 15th Symposium on International Database Engineering and Applications*, (Lisbon, Portugal; September 21-23, 2011), pp. 249-250, 2011.

[LeMB08] Carson Kai-Sang Leung, Mark Anthony F. Mateo, and Dale A. Barjczuk, "A tree-based approach for frequent pattern mining from uncertain data," *in Advances in Knowledge Discovery and Data Mining*, pp. 653-661. New York, NY: Springer, 2008, 1099pp.

[LiCF08] Guan-Hao Li, Chih-Fu Chang, and Li-Chen Fu, "Navigation of a wheeled mobile robot in indoor environment by potential field based-fuzzy logic method," *in Proc. IEEE Int. Conf. Advanced Robotics and its Social Impacts*, (Taipei, China; August 23-25, 2008), pp. 1-6, 2008.

[LiWK06] D. K. Liu, X. Wu, A. K. Kulatunga, and G. Dissanayake, "Motion coordination of multiple autonomous vehicles in dynamic and strictly constrained environment," *in Proc. IEEE Conf. Cybernetics and Intelligent Systems*, (Bangkok, Thailand; June 7-9, 2006), pp. 1-6, 2006.

[LoHa07] Lyle N. Long, Scott D. Hanford, Oranuj Janrathitikarn, Greg L. Sinsley, and Jodi A. Miller, "A review of intelligent systems software for autonomous vehicles," *in Proc. IEEE Symposium on Computational Intelligence in Security and Defense Applications*, (Honolulu, HI; April 1-5, 2007), pp. 69-76, 2007.

[OlKa12] Ayrton Oliver and Steven Kang, "Using the Kinect as a navigation sensor for mobile robot," *in Proc. the 27th Conf. Image and Vision Computing New Zealand*, (Dunedin, New Zealand; November 26-28, 2012), pp. 509-514, 2012.

[PaGS05] S. Parasuraman, V. Ganapathy, and Bijan Shirinzadeh, "Behaviour based mobile robot navigation technique using AI system: Experimental investigation on active media pioneer robot," *J. Int. Islamic University Malaysia Engineering*, vol. 6, no. 2, pp. 13-25, 2005.

[PaSk94] Zdzislaw Pawlak and Andrzej Skowron, "Rough membership functions," *in Advances in the Dempster Shafer Theory of Evidence*, pp. 251-271. New York, NY: John Wiley & Sons, 1994, 608 pp.

[Pawl82] Zdzislaw Pawlak, "Rough sets," *Int. J. Computer & Information Sciences*, vol. 11, no. 5, pp. 341-356, Oct 1982.

[Pawl97] Zdzislaw Pawlak, "Rough set approach to knowledge-based decision support," *European J. Operational Research*, vol. 99, pp. 48-57, May 1997.

[PePS02] James F. Peters, Zdzislaw Pawlak, and A. Skowron, "A rough set approach to measuring information granules," *in Proc. Int. Conf. Computer Software and Applications*, (Oxford, England; August 26-29, 2002), pp. 1135-1139, 2002.

[Peri05] Vamsi Mohan Peri, *Fuzzy Logic Controller for an Autonomous Mobile Robot*. Master's Thesis. Cleveland, USA: Cleveland State University, 2005, 161 pp.

[PeSK08] Witold Pedrycz, Andrzej Skowron, and Vladik Kreinovich, *Handbook of Granular Computing*. New York, NY: Wiley, 2008, 1148 pp.

[Pete07] James F. Peters, "Near set: General theory about nearness of objects," *J. Applied Mathematical Sciences*, vol. 1, no. 53, pp. 2609-2629, 2007.

[Pete09] James F. Peters, "Fuzzy sets, near sets, and rough sets for your computational intelligence toolbox," *in Foundations of Computational Intelligence*, pp. 3-25. New York, NY: Springer, 2009, 313pp.

[Pete13] James F. Peters, "Near sets: An introduction," *J. Mathematics in Computer Science*, vol. 7, no. 1, pp. 3-9, 2013.

[RaMa11] V. Raudonis and R. Maskeliunas, "Trajectory based fuzzy controller for indoor navigation," *in Proc. IEEE Int. Symposium on Computational Intelligence and Informatics*, (Budapest, Hungary; November 21-22, 2011), pp. 69-72, 2011.

[RaTS09] S. M. Raguraman, D. Tamilselvi, and N. Shivakumar, "Mobile robot navigation using fuzzy logic controller," *in Proc. Int. Conf. Control, Automation, Communication and Energy Conservation*, (Perundurai, Tamilnadu; June 4-6, 2009), pp. 1-5, 2009.

[Rutk03] Danuta Rutkowska, "Perception-based reasoning: Evaluation systems," *in Task Quarterly*, vol. 7, no. 1, pp. 131-145, 2003.

[SaYe98] M. Sarkar and B. Yegnanarayana, "Rough-fuzzy membership functions," *in Proc. IEEE Int. Conf. Fuzzy Systems*, (Anchorage, USA; May 4-9, 1998), vol. 1, pp. 796-801, 1998.

[ScGW10] Daniel L. Schacter, Daniel T. Gilbert, and Daniel M. Wegner, *Psychology*. New York, NY: Worth Publishers, 2010, 800pp.

[ScRz12] Val E. Schmidt and Yuri Rzhanov, "Measurement of micro-bathymetry with a GOPRO underwater stereo camera pair," *in Proc. IEEE Conf. Oceans*, (Virginia, USA; October 14-19, 2012), pp. 1-6, 2012.

[Shan48] Claude E. Shannon, "A mathematical theory of communications," *Bell Systems Technical J.*, vol. 27, pp. 379-423, 623-656, Jul 1948.

[ShNa06] Himanshu Dutt Sharma and Umashankar N., "A model approach for perception based intelligent system design," *in Proc. IEEE Int. Conf. on Robotics and Biomimetics*, (Kunming, China; December 17-20, 2006), vol. 1, no. 114, pp. 660-665, 2006.

[SiBo04] Nozer D. Singpurwalla and Jane M. Booker, "Membership functions and probability measures of fuzzy sets," *J. American Statistical Association*, vol. 99, no. 467, pp. 867-877, Sep 2004.

[SkWa10] Andrzej Skowron and Piotr Wasilewski, "An introduction to perception-based computing," *in Proc. the Second Int. Conf. Future Generation Information Technology*, (Jeju Island, Korea; December 13-15, 2010), vol. 6485, pp. 12-25, 2010.

[SkWa12] Andrzej Skowron and Piotr Wasilewski, "Introduction to perception based computing," *in Emerging Paradigms in Machine Learning*, pp. 249-275, New York, NY: Springer, 2013, 498 pp.

[SuHP95] Hartmut Surmann, Jorg Huser, and Liliane Peters, "A fuzzy system for indoor mobile robot navigation," *in Proc. Int. Joint Conf. Fuzzy Systems and Fuzzy Engineering Symposium*, (Yokohama, Japan; March 20-24, 1995), vol. 1, pp. 83-88, 1995.

[Tan11] Tan Lujiao, *Takagi-Sugeno and Mamdani Fuzzy Control of a Resort Management system*. Master's Thesis. Karlskrona, Sweden: Blekinge Institute of Technology, 2011, 53 pp.

[TaSu85] Tomohiro Takagi and Michio Sugeno, "Fuzzy identification of system and its applications to modeling and control," *IEEE Trans. System, Man, and Cybernetics*, vol. 15, no. 1, pp. 116-132, Feb 1985.

[ThSu00] S. Thongchai, S. Suksakulchai, D. M. Wilkes, and N. Sarkar, "Sonar behaviour-based fuzzy control for a mobile robot," *in Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, (Nashville, TN; October 08-11, 2000), vol. 5, pp. 3532-3537, 2000.

[Titc00] Mark R. Titchener, "A measure of information," *in Proc. Data Compression Conf.*, (Snowbird, USA; March 28-30, 2000), pp. 353-362, 2000.

[WaHo09] Xiuqing Wang, Zeng-Guang Hou, Min Tan, Yongji Wang, and Liwei Hu, "The wall-following controller for the mobile robot using spiking neurons," *in Proc. IEEE Int. Conf. Artificial Intelligence and Computational Intelligence*, (Shanghai, China; November 7-8, 2009), vol. 1, pp. 194-199, 2009.

[Wang02] Yingxu Wang, "On cognitive informatics," *in Proc. 1st IEEE Int. Conf. Cognitive Informatics*, (Calgary, Canada; August 19-20, 2002), pp. 34-42, 2002.

[Wang11] Xueming Wang, *Wall Following Algorithm for a Mobile Robot Using Extended Kalman Filter*. Master's Thesis. Auburn, USA: Auburn University, 2011, 50 pp.

[WaZK10] Yingxu Wang, Du Zhang, and Witold Kinsner, *Advances in Cognitive Informatics and Cognitive Computing*. New York, NY: Springer, 2010, 297 pp.

[WeBi06] Greg Welch and Gary Bishop, *An Introduction to the Kalman Filter*. Technical Report. Chapel Hill, USA: University of North Carolina at Chapel Hill, July 2006, 16 pp.

[XiBe84] W. X. Xie and S. D. Bedrosian, "An information measure for fuzzy sets," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 14, no. 1, pp. 151-156, Feb 1984.

[Zade99a] L.A. Zadeh, "From computing with numbers to computing with words: Manipulation of measurements to manipulation of perceptions," *in Proc. the Second Int. Conf. on Intelligent Processing and Manufacturing of Materials*, (Honolulu, HI; July 10-15, 1999), vol. 1, pp. 3-4, 1999.

[Zade99b] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *J. Fuzzy Sets and Systems*, vol. 100, pp. 9-34, Jan 1999.

[Zade01] Lotfi A. Zadeh, "A new direction in AI: Toward a computational theory of perceptions," *in AI Mag.*, vol. 22, no. 1, pp. 73-84, Oct 2001.

[ZhMi04] Wenxiu Zhang and Jusheng Mi, "Incomplete information system and its optimal selections," *Int. J. Computers and Mathematics*, vol. 48, no. 5-6, pp. 691-698, Sep 2004.

APPENDIX A

Software

This chapter outlines the instruction of executing the simulation and experiment, respectively. The theoretical descriptions are stated in Ch. 4, Ch. 5, and Ch. 6. The source code is included in the "SourceCode" folder of the DVD accompanied this thesis. In this folder, "KNS" folder contains the source code of simulation, "PBCRFVNS" is the source code files for the experiment, and "PBCRFVNS1" is the executable files to run the experiment. Matlab 2012b or later is recommended for the simulation. Windows visual studio 2010 or later, and Arduino 1.0.5 or later is recommended for the experiment. Kinect for windows SDK 1.7 or later version is recommended.

A.1 Simulation

1. Copy the "KNS" folder from the DVD, and paste it into the Matlab source code folder.
2. Click "Run" to execute the simulation and all graphical results using distinct control algorithms should pop out. The energy-based (*e.g.*, mean and variance)

results are stored at the "RESULT" parameter, and information measure is in the "SHI" parameter.

3. All parameters are labelled with explanations, and change the settings to get different functions for the simulation.

A.2 Experiment

1. Connect the Kinect sensor and the Arduino Uno R3 board to the laptop.
2. Copy the "PBCRFVNS1" folder and paste it to the workspace folder of Windows visual studio.
3. Copy the "PBC_RF_VNS" folder and paste it into the workspace of Arduino 1.0.5.
4. Click "Upload" to check and upload the software into the Arduino Uno R3 board.
5. Open "PBCRFVNS1.sln" in the visual studio and click "Start Debug" to run the navigation software.
6. Test the software with different parameters that can be modified in the control panel.

APPENDIX B

Experiments Raw Data

The total amount of sampled data during the experiment is 4000, and each navigation algorithm (*i.e*, CFBC, RFBC, SNN, and BBC) has 1000 measured data. In order to better analyze the data, only 100 samples for each algorithm are listed below. The format of data is as order of sample, the measured distance, and the measured orientation difference. The full version of measured data is in the folder named "ExperimentalData" included in the DVD.

RFBC:

0	2.30646068067851	26.238770858897	50	2.97128396599549	0.123220967658797
1	2.32770431692916	25.7187821815493	51	2.97577842438984	0.118009422568188
2	2.39840469077318	30.6414176403867	52	2.97797778165119	0.115531669675193
3	2.52036352091928	37.4248135847613	53	2.98216181340599	0.110946102815274
4	2.71750903982372	26.022892678457	54	2.98677937980687	0.106075082830318
5	2.94003155086966	2.86656164068938	55	2.98862529333394	0.104182079736227
6	3.12945726378657	2.09330593702116	56	2.99518844008067	0.0976943337385458
7	3.22149704924146	-2.55154603122284	57	2.99886772606938	0.0942181597133919
8	3.26355968551314	-2.75551866640414	58	3.00069043504694	-2.07937571098242
9	3.26710980400109	-2.77523328257282	59	3.00107800221302	-2.07967985989563
10	3.24158953246455	-2.64240645076903	60	3.00120594039533	-2.07978049087879
11	3.20272906440954	-0.47660130303704	61	3.00010850722719	-2.07892099376998
12	3.16391367161854	-0.34848552545643	62	2.99577473072342	0.0971328085034029

13	3.12611480184722	-0.253258683323743	63	2.99222994203137	0.10057256121794
14	3.09384253334065	-0.190704917717468	64	2.98269972029843	2.11036854818723
15	3.07044969068139	-0.154264069968378	65	2.97229837967573	2.12202707579821
16	3.05155390447204	-0.12946689692358	66	2.96688349579102	2.12852159990975
17	3.0303509698871	-0.105909968169795	67	2.963695818956	2.13248738147855
18	3.01259595286272	-0.0892103150442567	68	2.96041843942918	2.13667808233944
19	3.00375208865596	-0.0818070815131806	69	2.95454615975337	2.14448277082193
20	2.99424236192213	2.0986065964349	70	2.95142988761128	2.14878336460536
21	2.98290382018588	2.11015010992168	71	2.94865263290722	2.15271151734548
22	2.98326109694509	2.10976866537425	72	2.95705114914516	0.141106257951539
23	2.98654169018542	2.10632106251398	73	2.963374261031	0.132893423256146
24	2.98806606615912	2.10475235022211	74	2.96742329678105	0.12786061614008
25	2.99266436850168	2.10014521125578	75	2.97493789441182	0.118968819341271
26	2.99795755202013	0.0950675833375283	76	2.98848074546624	0.104329214736669
27	3.00573129178004	-2.08341422126156	77	2.99670505457184	0.096247728276896
28	3.01507292947706	-2.09138873320955	78	3.00828415092396	-2.08552900489503
29	3.01893366013643	-2.09487920352202	79	3.01531730448769	-2.09160621404885
30	3.01875304670735	-2.09471328894908	80	3.01902295186015	-2.09496132444919
31	3.01307230754702	-2.08962560330965	81	3.02213010515125	-2.09785885549597
32	3.00627618433521	-0.0838616284447484	82	3.01844426920082	-2.09443024192627
33	2.99929341503245	2.0938232293334	83	3.00986468317215	-0.0868622997942845
34	2.98706605748078	2.10577908496854	84	3.00342609505052	-0.0815450538668479
35	2.97681165128786	2.11683955502686	85	2.99876336079265	2.09431521168088
36	2.97164921221831	2.12278990313783	86	2.99270663815283	2.10010371724986
37	2.96719013996638	2.1281457427258	87	2.98668060051117	2.10617724532946
38	2.96515714655034	2.13065600772023	88	2.97875173685834	2.11467087504729
39	2.96069976985122	2.13631377740677	89	2.97309174945086	2.12110054600974
40	2.95317162033078	2.14636588714612	90	2.96975872595224	2.12503569461479
41	2.95056889659084	2.14999144348251	91	2.9670802344395	2.12828034230488
42	2.95456381045189	2.14445873009524	92	2.9663184546543	2.12921675244882
43	2.95941909302435	0.137979189934679	93	2.96461878607854	2.13132805340217
44	2.95985627325609	0.13740864905446	94	2.96269863490429	2.13375017233126
45	2.9615274397915	0.135247005555673	95	2.96314946512525	2.13317793850801
46	2.96538335994787	0.130374536996964	96	2.96235474173815	2.13418814356265
47	2.96634733998697	0.129181134581138	97	2.96305875576953	2.13329289931436

48 2.96658833499674 0.128884309946086
49 2.96821278336265 0.126899352722165
98 2.96568414134756 0.130001120638039
99 2.96798673495035 0.127173924070037

CFBC:

0 2.33442675278512 21.7733072234393
1 2.3586647318787 11.2982349351161
2 2.41817662912446 14.7464270518572
3 2.49569307659745 17.2833654915584
4 2.60195288298148 18.2783131552102
5 2.70228101192828 20.1850480228897
6 2.80408761167223 22.3025951299225
7 2.91811274146588 23.6824372607406
8 3.02521247254897 1
9 3.14005184649602 1
10 3.20117523377593 1
11 3.19158493953152 -6.36546763753583
12 3.12190260907712 -1.25609911050451
13 3.03161011227235 -5.38311881729801
14 2.96843628581866 -1
15 2.92360614482037 -1
16 2.89481823149869 -1
17 2.87797472926321 -1
18 2.86716679405121 7.79117719681708
19 2.87439532868913 10.3205367425957
20 2.91540147327095 1.58347229004743
21 2.96413651967865 4.8844380216336
22 3.00872916513927 1
23 3.05687005970979 1
24 3.10498619760367 1
25 3.13152298026104 -5.79408034245771
26 3.13987731095201 -7.18369654888446
27 3.12297690791661 -10.0529103551487
28 3.07261053740951 -1.4155685787331
29 3.00877979107733 -4.27526230062407
30 2.97059081524624 -1
50 2.96834256699593 9.07370434108562
51 2.9625406456648 9.78501242187039
52 2.98369296665369 11.7845294458638
53 3.0159539386337 -10.3750014964475
54 3.02865768330128 -9.69748772922618
55 3.0173327210198 -11.0439309965505
56 2.98674357722876 11.4812197194132
57 2.97079095367581 10.8232694768731
58 2.99154566316219 12.9019542441106
59 3.03200051598423 -7.78728835161788
60 3.06089513884886 -6.9012318182079
61 3.05877017442224 -8.73774595653987
62 3.02548368303124 0.777135009504967
63 2.98373766623879 10.4326245439512
64 2.9761221694217 10.5503290238088
65 2.99610695548027 12.5825583070866
66 3.0311218214532 -9.54865458200395
67 3.05201604302203 -9.28529024824527
68 3.03027009015112 0.817271032729217
69 2.99920191892129 10.4446904172404
70 2.9973354031149 11.4459211572237
71 3.03125237341601 -10.2977512888575
72 3.04433093312332 -9.76374457955157
73 3.00800834665614 -0.932092460863592
74 2.96888626225027 9.60703463278869
75 2.95300027206095 9.37599307676851
76 2.97824502134056 12.2165611526219
77 3.01353770744273 -9.40962124151898
78 3.0314920697163 -9.02191121918848
79 3.02413321010341 -9.93009516647073
80 2.99197352829298 11.756343175666

31 2.94334617767538 -1
32 2.91676584471042 -1
33 2.89519885960504 7.62474957085138
34 2.88772225695247 8.91104355991921
35 2.91376373129993 11.6743121880746
36 2.95075508157699 2.49660964130116
37 2.98909537210707 4.99123111981772
38 3.02146709664102 -5.87732029309826
39 3.04415677446603 -6.28385999460466
40 3.04914396068303 -7.82891013485848
41 3.04099606683458 -9.55962969558643
42 3.01782334359333 0.38946931066738
43 2.99134017904116 10.5751376959904
44 2.98387865198924 10.4101755977366
45 2.99778417922348 11.8006539542819
46 3.03002487798473 -9.98790803875464
47 3.04932520204755 -9.13436669598738
48 3.03559995154011 -10.6935426935778
49 3.00176126118692 -1.53622412874253
81 2.96628492299874 10.7252402289521
82 2.97968982210065 12.5355993975329
83 3.00542949144414 -9.48443688575523
84 3.02150838368519 -9.0525050887209
85 2.99037946354114 12.7585026111796
86 2.95910032749114 11.135514578613
87 2.9621217130074 11.9517368327556
88 2.9936039630217 2.68637776876241
89 3.04043967734842 -6.65664184002639
90 3.07704572223975 -5.84712936327552
91 3.08772194315993 -7.11547518720132
92 3.08129878647756 -9.08544716784686
93 3.06042363482477 -11.0431612167739
94 3.02895031429863 -0.979244727434217
95 3.00975661235824 -2.14234538591781
96 2.99479105638099 9.68497759569142
97 2.99837589070067 10.8001489454633
98 3.01700831606616 0.109455783966528
99 3.03526015164137 -10.4761509027237

SNN:

0 2.34011210784736 25.0976393603581
1 2.36840917179003 23.8601793101568
2 2.45903154138117 20.0157790337768
3 2.64802440994017 12.5122760229197
4 2.84031377439051 5.47251548056139
5 2.9811102699601 0.631663754703089
6 3.05184579832729 -1.71350278491693
7 2.99748264637811 0.0839581433502801
8 2.86864610671551 4.47871857637145
9 2.79403230338156 7.118657213789
10 2.77451010664587 7.82177528151589
11 2.79189107284564 7.19551856361916
12 2.82548041426158 5.99698861289752
13 2.86692791672746 4.53869289720948
50 3.03801549201274 -1.25924379557546
51 3.06101780402901 -2.01364393508949
52 3.07466149565913 -2.45848913191866
53 3.06910497824986 -2.27755523717566
54 3.0327110276175 -1.08447620582174
55 2.97338827278497 0.891038352885714
56 2.9266586828212 2.47535387614255
57 2.89841012903959 3.44568630381238
58 2.88486688396581 3.91436353091909
59 2.88111190577869 4.0447126600283
60 2.88176288319908 4.02210218085505
61 2.89456043679403 3.57867738814917
62 2.91494382849334 2.87658188949314
63 2.94026317480843 2.0114665602636

14	2.92032610932059	2.69203619896931	64	2.98066181312735	0.646708477610281
15	2.97561150836838	0.81629244343314	65	3.02381364254635	-0.790651374633029
16	3.0280575844884	-0.930908842670561	66	3.05759279551856	-1.90166865557669
17	3.05899495324088	-1.94752486267369	67	3.081008357399	-2.66476880569219
18	3.07369604247692	-2.42707452072643	68	3.08574385873399	-2.81840827154892
19	3.0744349093195	-2.45111716666484	69	3.08230370166147	-2.70681800041032
20	3.06277689399654	-2.0711069025635	70	3.07302086606559	-2.40509942292169
21	3.04669378305482	-1.54452130501028	71	3.05426951860606	-1.79290165216815
22	3.02162646444586	-0.718291357983061	72	3.03312653175625	-1.09817683409382
23	2.99296001293552	0.234953361623642	73	3.01706161518243	-0.567101387977056
24	2.96503481547094	1.17238951177746	74	3.00211491807423	-0.0704613340348725
25	2.94570808401711	1.82641929027424	75	2.98447043275883	0.519010201359142
26	2.92719458297737	2.4570389939294	76	2.96990147363388	1.00837933853078
27	2.91597888428244	2.84106508169709	77	2.9545137061106	1.52789250380968
28	2.91536292971197	2.86219932675674	78	2.95001471644294	1.68030332975824
29	2.92488933361774	2.53584743427872	79	2.94486537919937	1.85503611954741
30	2.94457442371687	1.86491841224179	80	2.94073693851694	1.99535165391658
31	2.97076801667391	0.979204649121881	81	2.94000593449486	2.02021760887154
32	3.00875191947991	-0.291295274434669	82	2.93985288865951	2.02542444005038
33	3.04889532491118	-1.61676427298793	83	2.93979597575749	2.02736076630298
34	3.08016750206984	-2.63746392462976	84	2.94180859747786	1.95890931679236
35	3.09700430191613	-3.18282651780746	85	2.94324435987773	1.9101066553806
36	3.09723762787511	-3.19036398311761	86	2.95100244027737	1.64682230106227
37	3.06785399595469	-2.23677604680515	87	2.95849713138065	1.39314384311099
38	3.02989723060199	-0.991646548769349	88	2.96562588188415	1.15245559734094
39	2.99314614747002	0.228734459032109	89	2.97307845148981	0.901459121648589
40	2.96230407511316	1.26453688879216	90	2.9807727771212	0.642985671910843
41	2.94171777266538	1.96199733812834	91	2.98910452716763	0.363855490756642
42	2.92709031706406	2.46060210849456	92	3.00255670758095	-0.0851762000019018
43	2.91997424254687	2.70409019761776	93	3.01277251841835	-0.42483707420776
44	2.91676230178126	2.81419158610006	94	3.02383780577984	-0.791450494501866
45	2.92175508868487	2.64309857362246	95	3.03251464372614	-1.07800009420656
46	2.94278795674859	1.92561751522422	96	3.03929244965283	-1.30127103746561
47	2.96785979751135	1.07715231130996	97	3.04399436634421	-1.45587053348465
48	2.99012856492315	0.32960213336251	98	3.04346663824074	-1.43853051526147

49 3.01152405278128 -0.383389048486507
99 3.04003914123212 -1.32583808306619

BBC:

0 2.34813727387892 13.5872365509742
1 2.36122180202162 14.4540026549421
2 2.38102685335474 16.1275619690006
3 2.40855280897362 18.1295062090103
4 2.45315425323174 20.9896999423793
5 2.52128011393947 24.9245842595697
6 2.62381369948484 28.9930077282221
7 2.79930348742336 35.1786326159553
8 3.11836562335481 14.0589711426633
9 3.52871420028595 17.3101347865348
10 3.83206275527791 12.2386930872439
11 3.99683262423881 5.63015503083439
12 4.08419488044232 -1.32544812915297
13 4.11104166948692 -7.28563773551972
14 4.09586728274805 -12.0610179065417
15 4.0441925335485 -16.9323873410445
16 3.94682312222584 -21.8071241690403
17 3.79737137073757 -27.6969787722111
18 3.58532241698825 -34.2370623186431
19 3.35275919232126 -39.7842814173852
20 3.10679963944973 -43.8035405885198
21 2.86688513181595 -15.814774815035
22 2.6801785461311 -15.3608712885524
23 2.575973008502 -12.7958898431075
24 2.51276489870857 -10.0679004960543
25 2.45514570295118 -6.49965526712794
26 2.41187137464522 -1.50668953329088
27 2.3987358716991 3.22554757746446
28 2.40377675064872 6.92362904107856
29 2.42672679743681 10.2596123675998
30 2.4602912332638 13.804051746757
31 2.5040231103654 17.1635520525867
50 2.92648841898115 15
51 2.96514045298765 15
52 3.02118842608699 -15
53 3.05173957164393 -15
54 3.0359320131536 -15
55 2.97292142731476 15
56 2.91612592890472 15
57 2.93723421125307 15
58 2.99982400801823 15
59 3.06130339984284 -15
60 3.0928023644441 -15
61 3.0723092663729 -15
62 2.99894648957908 15
63 2.93486659343877 15
64 2.95097735575786 15
65 3.00725654089041 -15
66 3.05608358642307 -15
67 3.03735177885777 -15
68 2.93501378484192 15
69 2.82387810799072 6.60218114848069
70 2.81581464097337 10.1033358234644
71 2.86074922559329 15.0978731697291
72 2.91107014209785 15
73 2.9609843436577 15
74 3.00808882389344 -15
75 3.02363266744342 -15
76 2.99814740260815 15
77 2.97120640952414 15
78 2.98903874067644 15
79 3.02722997587457 -15
80 3.04196760969428 -15
81 2.99738670487112 15

32	2.54523271421567	20.2458128963436	82	2.92534601380926	15
33	2.59138712409089	22.7209356598859	83	2.93137076188967	15
34	2.64817959269077	25.6913383177545	84	2.97225562601458	15
35	2.72480367028011	29.0233700170027	85	3.03783250926872	-15
36	2.80916159835813	32.3457611996821	86	3.09338075123393	-15
37	2.91262553838648	15	87	3.09286712926796	-15
38	2.99281640390901	15	88	3.06578845545352	-15
39	3.05785750396198	-15	89	3.00739468758851	-15
40	3.10658750114309	2.11905303230575	90	2.91267405761096	15
41	3.12272394299286	-3.74220687297808	91	2.87333242121891	6.6372814654001
42	3.12167248511485	-8.49324901790312	92	2.89206922306265	10.0287892188336
43	3.11223722354299	-11.092520541195	93	2.92598054569177	15
44	3.09504862056695	-15	94	2.97854061462413	15
45	3.04250607671803	-15	95	3.05724525852844	-15
46	2.94024918505294	15	96	3.09528618873251	-15
47	2.87353918003356	5.67920137733014	97	3.07904862974236	-15
48	2.8760331538484	8.35636633069539	98	3.02095139702722	-15
49	2.89618470428775	11.3482579674851	99	2.93849034726558	15

APPENDIX C

Hardware

C.1 Servo Motor

In order to precisely control the heading direction of a mobile robot, it is necessary to have a steering unit that executes exactly as the outputs of the RFBC, *i.e*. the corrected heading angle ω (see Sec. 4.3). A servo motor shown in Fig. 8.1 is a rotary actuator that performs precise position control. The servo motor is comprised of a DC motor, potentiometer, and a control circuit.

Fig. 8.1: Traxxas 2056 servo motor.

The output control wheel connects to the motor through gears. By changing resistance of the potentiometer along with rotation of the motor, the control circuit is able to precisely determine the amount of movement and its direction. The power supply of the motor is disconnected as soon as it is at the desired position that is specified by the input electrical pulse. The speed of the motor is proportional to the difference between its actual position and desired position. That is to say, the motor turns slowly if it is close to the desired position, otherwise it turns quickly. It is called the proportional control.

Generally, the amount of the shaft movement of the motor is determined by an input electric signal, called pulse width modulation (PWM). It is a modulation technique that conforms the duration of the pulse according to modulator signal information. The moving position of the shaft is determined by the duration of the pulse. Usually a servo motor receives input PWM every 20 milliseconds in order to keep the motor steady. For instance, as illustrated in Fig. 8.2, the shaft moves 0 degrees if the input pulse has duration of 1 milliseconds. It moves 90 degrees if the duration of an input pulse is exactly 1.5 milliseconds, and any input pulse that duration is 2 milliseconds drives the motor to 180 degrees.

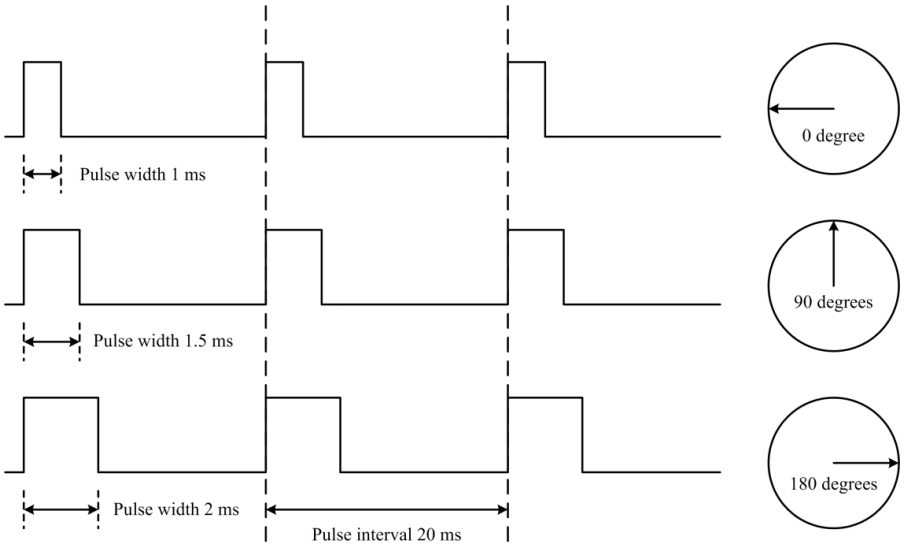


Fig. 8.2: The pulse width modulation.

Another device that may be suitable for the steering purpose is called stepper. Compared with a servo motor, the stepper motor provides a constant holding torque without the need of power. The torque of a stepper at low speeds is greater than a servo motor. Perhaps the most obviously advantage of a stepper is its relatively low cost and availability. For the purpose of indoor navigation, the steering response should be precise and fast. Both stepper and servo are able to move to desired position precisely. However, the response time of a stepper is substantially longer than a servo motor. Stepper motor peaks around speeds of 2,000 rounds per minute, while servo motor is available many times faster. Besides, a servo motor is more efficient in terms of power consumption. Compared with a stepper motor, a servo motor is better suited to high speed, high torque applications that involve dynamic load changes. By contrast, a stepper motor is less expensive and is optimal for applications that need low acceleration, high holding torque, and the flexibility of open or closed loop operation.

C.2 Arduino Uno R3

In the workflow of the proposed vision-based indoor navigation system, the control unit subsystem guides the movement of a mobile robot by correcting the heading angle. A mobile platform is designed to be only responsible for movement by receiving electrical signals. Outputs from the control unit are numerical numbers and thus they cannot be used immediately to a mobile platform. Therefore, an interpretation device should exist to translate outputs of the control unit into proper inputs of the mobile platform.

Fig. 8.3: Arduino Uno R3.

The type of Arduino used for the proposed indoor navigation system is Arduino Uno R3 as shown in Fig. 8.3. It is a microcontroller board based on the ATmega328. It has 14 digital input/output pins including 6 PWM pins. In this case, it is able to translate numerical outputs (control unit) into electrical signals for general motors and PWM signals for servo motors. Furthermore, it has a 32KB memory and 2KB SRAM. It also

provides 6 analog input pins, and two types of output voltage, which are 3.3V and 5V, respectively.

C.3 Kinect Sensor

C.3.1 Comparison with GOPRO Hero3

The performance of image processing subsystem has significant influence on outputs of the control unit subsystem. Consequently, an efficient image processing subsystem is as important as the control unit. Two aspects are essential when evaluating the image processing subsystem: (i) frame quality and (ii) processing efficiency. Frame quality varies among distinct selection of sensor hardware. By contrast, processing efficiency is more complicated and it depends on specific application purpose.

The minimum effective detection range for a Kinect sensor is 0.4 meter and maximum range is 3.5 meters. The resolution is 10 millimeters at 2 meters range [OlKa12]. The field of view is 58 degrees. A Kinect sensor provides multiple frame formats for both depth image and conventional color image ranging from 640×480 to 1280×960 pixels at maximum 30 frames per second (PFS).

In addition to the introduction of Kinect sensor, another vision sensor named Hero3 is also considered in terms of frame quality. It is popular with outdoor sports and manufactured by the GOPRO Company. Compared with a Kinect sensor, a Hero3 is able to capture higher quality frame with respect to resolution and frequency. It supports a maximum video resolution of 1920×1080 pixels at 30 FPS. The maximum field of view

is 170 degrees at resolutions of 1280×960 pixels, which provides rich environment information. Besides, it performs very well in low light environment. Furthermore, two or more Hero3 cameras have the capability of producing stereo frames with the help of a stereo camera kit and synchronization cable. Synchronized images of a scene are taken by stereo cameras from differing vantage points. If an object is identified on both images, the three-dimensional (3D) information can be calculated by knowing the translation and rotation of each camera relative to the second. In this case, both Hero3 and Kinect can provide depth information of objects in the environment [ScRz12].

In this thesis, a mobile robot should be able to calculate the distance between itself and the wall quickly and reliably. It requires precise and efficient image processing techniques, which implies massive geometrical mathematic calculation. Time issue is critical since the calculation is conducted while the robot is moving. Stereo images taken from Hero3 cameras can be analyzed through software such as OpenCV. It is true that the data precision is guaranteed. However, the processing time cannot compete with techniques that combine software and hardware. The primary advantage of a Kinect sensor is the ability of building depth maps within quite short period (4 milliseconds for depth image with resolution of 640×480 pixels). Since it utilizes both hardware (IR sensor) and software, the results are much faster and more reliable. Though the frame quality is lower than a Hero3 camera, a Kinect sensor is selected in terms of image processing efficiency.

In addition to above proposition, the standard SDK of Kinect sensors is provided, which facilitates the use of a Kinect sensor. By contrast, the Hero3 camera lacks the

flexibility of customization. Since the vision-based indoor navigation system integrates multiple types of hardware, it is better to choose hardware that is easy to install and integrate. A Hero3 camera is much lighter (74g) than a Kinect sensor (1360g), which indeed constitute better carriage performance. However, the battery life of a hero3 camera is very short (maximum 2 minutes with resolution of 1920×1080 pixels at 30 FPS). A Kinect sensor initially requires AC power. It is able to work using DC power after power supply modification, and the battery life is estimated over 4 hours.

C.3.2 Power Supply Modification

Initially an AC adaptor is used to power the sensor. However, the existence of a wired cable limits the moving range and performance. Especially for the navigation purpose, the moving range of a mobile robot could easily exceed 20 meters, as well as complicated layout of the indoor environment, which means a wired power cable is not feasible for this case. In order to overcome this problem, two distinct approaches are considered.

A straight forward way is to supply the Kinect sensor with movable AC power so that the original setup can be immediately used without any modification. A power device called *uninterruptible power supply* (UPS) is able to provide alternative AC power when the mains power fails. One main advantage of a UPS is the capability of near-instantaneous power continuation. It is usually used to protect hardware such as computers, data centers, and telecommunications equipment. As soon as the main power is down, the battery backup produces AC power through an inverter. However, there are two disadvantages that determine a UPS may not be a good choice for the proposed

application of this thesis. Firstly, the on-battery runtime of most UPS devices lasts within 20 minutes. It is sufficient to rescue data but not appropriate for research purpose. Another drawback is the weight of a UPS, which can be over 10 kilograms. For instance, one of the best choices for the proposed application may be a 255 watt UPS manufactured by the CyberPower Company. It provides backup runtime of 2 minutes at full load. However, the weight is 1.59 kilograms, which is too heavy to be placed on a mobile robot. Furthermore, the 8 hours recharge time is not acceptable for academic study.

Another solution would be a DC power supply directly to the Kinect sensor without the AC adaptor. A Kinect sensor works with 12 DC power. In this case, a battery pack is considered as an alternative DC power solution since it is much lighter than a UPS, and a battery pack runtime is much longer than a UPS. However, to achieve this goal, some modifications must be conducted. It is necessary to find out a way to directly power a Kinect sensor as well as to group batteries properly. The AC adapter cable is cut and the output end is connected to a battery holder. 8 batteries with voltage 1.5V and a BK-6049-ND battery holder shown in Fig. 8.4 are used to support functional performance of a Kinect sensor. A Kinect sensor has internal voltage regulator to maintain stable input voltage. The feasibility of battery power supply is verified through experiment. It works well and is able to last over 4 hours.

Fig. 8.4: BK-6049-ND Battery holders.