Adaptive Navigation and Motion Planning for Autonomous Mobile Robots

Dissertation
der Fakultät für Informations- und Kognitionswissenschaften
der Eberhard-Karls-Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr.rer.nat.)

vorgelegt von
Dipl. Ing. Ashraf Aboshosha
Aus Ägypten

Tübingen
2004

Tag der mündlichen Qualifikation	06.10.2004
Dekan:	Prof. Dr. Michael Diehl
1. Berichterstatter:	Prof. Dr. Andreas Zell
2. Berichterstatter:	Prof. Dr. Wolfgang Straßer

Acknowledgments

I wish to express my highest appreciation and gratitude to my advisors Prof. Dr. Andreas Zell and Prof. Dr. Wolfgang Straßer for the encouragement, suggestions and offering me the facilities necessary for this work.

Appreciation is extended to my current and former colleagues of Computer Science Dept., Faculty of Informatics, University of Tübingen for their hospitality, friendship and cooperation. It is a great honour to join this international group. I learned valuable teamwork within a multi cultural group, how to exchange ideas with respect among a qualified academic staff and how to explore the world of science and technologies for new innovations.

I would like to thank S. Wiest, A. Mojaev, H. Tamimi, P. Heinemann, C. Walter, C. Motoc and K. Beyreuther.

Also, I would like to acknowledge the financial support by the German Academic Exchange Service (DAAD) of my Ph.D. scholarship at the University of Tübingen.

I'm indebted to Amal, Mohamed and Osama for their patience in difficult circumstances and under stress for 4 years to end this work, I'm deeply grateful to them.

List of Abbreviations

2D	two dimensional
3D	three dimensional
AC	alternating current
ANFIS	adaptive-network-based fuzzy inference systems
ARX	auto regressive exogenous
APD	avalanche photo-diode
B-spline	bases spline
CCA	cross correlation analysis
CCD	charge couple device
CMOS	complementary metal oxide silicon
COG	center of gravity
COV	covariance
DC	direct current
DCT	discrete cosine transform
DFT	discrete Fourier transform
DST	discrete sine transform
DWT	Daubechies Wavelets
ED	Euclidean distance
FFT	fast Fourier transform
FIS	fuzzy inference system
FLC	fuzzy logic control
GA	genetic algorithms
GPS	global positioning system
GUI	graphical user interface
HSI	hue saturation intensity
HSV	hue saturation value
HWT	Haar wavelet transform
IGRF	international geomagnetic reference field
IR	infrared
LAN	local area network
LASER	light amplification by simulated emission of radiation
LMS	least mean square

MIL	matrix inversion lemma
MIMO	multi input multi output
ML	maximum likelihood
MRA	multi resolution analysis
MVC	model-view-controller
NN	neural networks
OO	object oriented
RADAR	radio ranging
RGB	red-green-blue color model
RL	reinforcement learning
RLS	recursive least squares
PC	personal computer
PAC	pole assignment control
PCB	printed circuit board
PPP	probabilistic path planning
SISO	single input single output
SLAM	simultaneous (concurrent) localization and mapping
SLN	straight line navigation
S-line	straight line
SONAR	sound navigation and ranging
TOF	time of flight
TS	Takagi-Sugeno
Var	variance
VLSI	very large scale integration
VMP	vector mapping paradigm
WMM	world magnetic map

List of Symbols

Symbols of Sonar and Laser Integration

r	distance
S_{trans}	sonar signal power
f	frequency
$I(\Theta)$	sonar characteristic radiation function
Θ	azimuthal angle of sonar
J_1	Bessel function of the first kind
K	wave number
λ	sonar wave length
$m_x,\ m_y$	mean value of x, y
$E()$	expectation
ρ	cross correlation
τ	delay
S_s	speed of sound
l_s	speed of light
$F(\omega)$	spectral analysis
d	data
a_i	wavelet averages
c_i	coefficients
h_i	scaling function coefficients
g_i	wavelet function coefficients
$\Phi(x)$	analyzing wavelet or mother wavelet
l	location
$W(x)$	scaling function for the mother function
δ	delta function

Symbols of Geomagnetic Compasses

I	inclination
D	declination
H	horizontal intensity
F	total intensity
Z	vertical intensity
X	north
Y	east

Symbols of Motion Planning for Non-Holonomic Robots

l	path length
Θ	deviation angle
L_i^n	lagrange polynomial
B_i^n	B-spline polynomial

Symbols of Image Processing

R	red component of RGB model
G	green component of RGB model
B	blue component of RGB model
H	hue component of HSV
S	saturation component of HSV model
V	value color component of HSV model
ρ	cross correlation
Λ	guide distance
N	number of pixels

Symbols of Adaptive Navigation

Θ	system parameters
Υ	aggregated system output
$\widehat{\Theta}$	estimated system parameters
υ	system output
Φ	input/output readings
ϵ	white noise
C_ϵ	covariance matrix
Ψ	Kalman based update matrix
λ	system input
$A(q^{-1})$	system model poles
$B(q^{-1})$	system model zeros
$T(q^{-1})$	assigned controller poles
$G(q^{-1})$	feedback controller component
$F(q^{-1})$	control signal conditioning
$D(q^{-1})$	state to output mapping matrix
q^{-1}	backward shift operator
H_c	compensator
n_a	number of poles
n_b	number of zeros
α	network learning speed factor
N	number of parameters, weights or coefficients
L	log likelihood function
V	variance
i, j, k	counters
$S(t)$	state either $\upsilon(t)$ or $\lambda(t)$
o	system output
$E()$	expectation or the average
$p()$	probability

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Exploring autonomy in robotics is a meaningful task. The intuitive definition of autonomy is the capability of a robot to make a decision based on its own knowledge, acquired by its distributed sensors, without any human interference. Throughout this framework we discuss some algorithms and techniques underlying the subjects of adaptive navigation and motion planning for autonomous mobile robots.

Mobile Robots will play an important role in many future applications, such as personal and service robots, handicapped aid, entertainment, space exploration, medical applications, nuclear industry, surveillance or autonomous transportation. It is the mobility that distinguishes a mobile platform from robot manipulators and gives the possibility to actively and adaptively interact with the environment and humans. In real world environments actual mobile robot platforms will need increased adaptability and autonomy with better techniques for navigation safety, map building, obstacle avoidance and path planning. This study will first focus on sensor integration and adapted interaction that is considered one of the major basic concepts for mobile platforms.

The contributions of this study arise from a formulation of new methods and techniques for sensor integration, mapping, motion planning and adaptive navigation instead of previously dominant approaches. By manipulating the manner in which feature information of sensor data is incorporated into processing, it can be shown that significant improvements in the performance of the presented algorithms can be attained. Moreover, the simplicity and the efficiency of the applied techniques succeeded to reinforce the robustness of the overall system in static and dynamic environments. The key idea of that is to achieve the following tasks improvement of system autonomy, reinforcement of the overall stability, enhancement of precision, increment of flexibility, reduction of energy consumption and

more adaptation.

Throughout the practical work, the integration of both laser and sonar succeeded to inherit advantages of both sensors and to eliminate their drawbacks by which system safety is assured. Furthermore, the development of the vector mapping paradigm (VMP), based on laser ranging, is considered one of the most important contributions of this framework. It preserves the consistency of the conventional mapping algorithms e.g. occupancy grid, topological graph and integration of both of them at the same time reduces the size of the map drastically. Related to the same algorithm, the model reference is employed to correct the odometric errors relying on a network of central nodes and their corresponding vectors. Compared with the traditional techniques used to correct map odometric errors based on neural networks, this algorithm is simpler, faster and generates high precision maps. Moreover, matching of laser signatures based on spatial, or spectral analysis and combinations of both of them under wavelets has led to speeding up matching process thereby the localization capabilities of mobile robots have been improved. Concerning mapping of a static environment, we employed not only visible odometric data acquired from laser and sonar sensors but also electromagnetic fields detected by a digital compass. Employing the invisible geomagnetic signature as a signature by detecting anomalies of compass deviation is a novel trend that replaces the conventional concept of considering the errors of geomagnetic compasses as a Gaussian white noise. Also in static environments, extending the Bug algorithm to our proposed SLN algorithm succeeded to create a new competitor to the Voronoi diagram, the most famous algorithm in motion planning.

In dynamic environments, the integration between the universe of discourse in fuzzy logic and the state space in stochastic Kalman filters provide mobile robots with reliable adaptive navigation that maintains their stability, robustness and leads to fine motion.

It is worth mentioning that not all algorithms and planned targets were successful. For example, due to a deficit in precise localization data, the adaptive navigation control has been applied to speed control, and not to position control. For the same reason, we failed to apply the SLN algorithm in real time and the algorithm is implemented as a simulator. Attributed to the presence of local minima, which emanates from similarity and affinity in features, the subject of imprecise localization is still a destructing problem in robotics. Furthermore, the use of a digital compass as an accurate heading device was faulty due to the presence of colored and white noise.

The remainder of the thesis is organized as follows:

Chapter 2 focuses on the concept of sensor integration with its advantages and disadvantages. Moreover, it clarifies the development guidelines of the whole project that comprise notes on the hardware and software techniques.

Chapter 3 draws out the design of the sonar laser integration scheme. It introduces the principles of sonar physics, a brief description of transducer types, the time of flight ranging concept, sonar 3D distribution, strengths, weaknesses, sonar mapping capability and fuzzy based fusion of sonar data.

Concerning laser ranging, this chapter explains laser characteristics, laser ranging modes, benefits, drawbacks, the vector mapping paradigm, using model reference for odometric error correction and laser signature matching. Finally, the integration scheme of laser and sonar is presented.

Chapter 4 shows the use of the compass as a heading and localization instrument. The basics of the natural geomagnetism and the use of digital compasses as a heading device are explained. Furthermore, the detection and analysis of geomagnetic signatures are studied.

Chapter 5 introduces motion planning for non-holonomic robots based on the SLN algorithm. The 5 phases of the SLN algorithm are initialization, segmentation, linearization and relaxation. Furthermore, a comparison between SLN and both Bug and Voronoi diagram is presented.

Chapter 6 illustrates the development of an adaptive navigation system for mobile robots based on the integration of both stochastic Kalman filters and fuzzy logic. The main purpose of this is to guide a mobile robot adaptively in unknown terrain using visual and laser data. So, the canonical structure of adaptive control systems modeling, identification, visual sensor data extraction and controller design is described in detail. Finally, a comparison between stochastic Kalman filters and fuzzy logic approaches is presented.

Chapter 2

Integration of Distributed Sensors

Over the last two decades there has been intensive research to support the sensory integration in robotics [189, 153, 17]. These efforts were devoted mainly to two major fields; the first one is the development of high precision sensors[1], the second is the development of an intelligent scheme to integrate homo- and heterogeneous sensors [105]. Throughout this framework we focus on illustrating the strategies of sensor integration and their benefits in mapping and adaptive navigation. Also we explain the characteristics, operation theory and data manipulation of the sensors under study.

For simple and small size robots, individual sensors are enough to accomplish their tasks. But recently robot tasks have become more sophisticated and simple sensor patterns are not adequate to achieve them [97]. Various problems occur with using individual/homogeneous sensors such as; **1)** Individual sensors cover only a restricted region with limited spatial consistency. Consequently, the physical measurement maps may be inconsistent. **2)** Individual sensors have narrow bandwidth and limited setup time. **3)** Due to several limitations and restrictions, readings of individual sensors may be imprecise. **4)** In spite of using complicated algorithms to assure the measurements of individual/homogeneous sensors, results of such systems are still uncertain. Nonetheless, integration of heterogeneous sensors is not an optimal solution. It exhibits different strengths and weaknesses.

2.1 Benefits and Drawbacks of Sensor Integration

On one hand, depending on sensor integration, the integrated system inherits several potential advantages, which are:

[1] in general environmental data acquisition systems

- Integration of heterogeneous sensors enables the system to still provide information even in case of individual failure, by which the robustness of the overall system is improved.
- Relying on sensor integration, the measurements are confirmed and drawbacks of individual sensors could be overcome. Therefore, sensor integration is essential for safe navigation in known/unknown terrains.
- The use of sensor integration leads to better spatial consistency, whereas individual sensors cover limited ranges and their readings may be inconsistent. Consequently, integrating heterogeneous sensor readings leads to an improvement of the overall consistency.
- Applying sensor integration reduces the ambiguity and the vagueness of the surrounding properties and helps in reaching a higher level of cognition.
- When multiple independent measurements made of the same property are integrated, the resolution of this value is better than of a single sensor's measurement.
- The confidence of multiple sensor systems is higher than of a single ones. The parallelism and the redundancy are employed to confirm the final results.

On the other hand, integration of heterogeneous sensors is fraught with difficulties:

- Dealing with heterogeneous physical properties of sensors increases the complexity of the applied algorithms.
- The massive use of sensor integration may lead to system instability, especially if the transmission and computation power is not adequate [197, 23].
- The sparse distribution of sensors increases the computational burden payed to interpret their measurements.
- Distributed sensors have different time bases and the synchronization of their control modules is critical.
- Sensor readings are accompanied by noise. The noise factor may multiply when increasing the number of integrated sensors.
- Processing of multiple sensor data demands higher computation power or costs more time.

2.2 Sensor Classification

Sensors are instruments used as input devices for robots which enable it to determine aspects regarding the robot's environment, as well as the robot's own status. They can also be thought of as devices which convert signals detected in the environment into an electrical signal that is sent to a robot's control system. Sensors are used for robot data acquisition and robot control. Sensors themselves can be classified into two categories: active or passive. Active sensors, also called transceivers or radiative sensors, are sensors which send and receive measurement modulated signals such as sonar, laser, radar and IR signals. Passive sensors measure and detect physical quantities.

Also, sensors can be classified according to various criteria such as;

Sensor position: internal and external

Measurement bases: time of flight, phase shift, touch, visual and magnetic

Subject of measurement: pose, velocity, acceleration, temperature, etc.

Carrier signal: IR, sonar, laser, radio, audio and video

Biological inspiration: biologically inspired sensors and non-biologically inspired (synthetic) sensors

Some of these sensor species are:

- Positioning and Ranging Sensors: are used to position mobile robots regarding their surroundings such as ranging sensors (sonar, laser, radar and IR), digital compass, geometrical vision, global positioning systems (GPS) etc.
- Motion Detection Sensors: detect motion parameters e.g., translation and rotation (encoders), rotational and longitudinal velocity (speedometers), acceleration (accelerometers), vibration (gyroscopes and inverted pendulums), etc.
- Vision Sensors: are used in surveillance of robot surroundings using general and special purpose cameras such as; CCD, CMOS, 3D, 3D laser, panoramic, IR cameras etc.
- Biologically Inspired Sensors: simulate biologically sensing organs such as; smelling (electronic nose), touching (artificial skin - tactile sensors), hearing (sound analyzer - speech synthesizer) and seeing (robot vision).

- Environmental Sensors: are used to explore robot surroundings and detect physical phenomena such as; temperature, air pressure, radiation, moisture, humidity, wind speed and direction, electromagnetic waves, metals, magnetic fields, etc.

This study focuses on the performance and integration scheme for sensors mounted on the RWI-B21 robot platform (Colin) including Polaroid 6500 sonar modules, Sick-LMS-200 laser range finder, vision system and fluxgate digital compass, see figure (2.1).

2.3 Sensor Fusion

Autonomous robots rely on numerous sensors to obtain a consistent and coherent view of the world. This inherently introduces difficulties as different sensors may react differently to the same stimuli, or may provide incorrect or inconsistent data. These sensor discrepancies have to be handled in a framework that allows the robot to visualize a unified view of its surrounding environment. Concretely speaking, Multi sensor integration refers to the synergistic use of the information provided by multiple sensory devices to assist in the accomplishment of a task by the system. Also, it may refer to any stage in the integration process where there is actual combination (fusion) of different sources of sensory information into one representation format.

In order to obtain the most accurate estimate of the dynamic system states, a convenient sensor fusion method is employed to integrate the data provided by homo- and heterogeneous sensors. Sensor fusion algorithms are particularly useful in high complexity applications (e.g. robotics, aerospace systems, nuclear plants, military applications and medical instrumentations) with a limited set of sensors, where acceptable performance and reliability are desired. The sensor fusion system can provide: filtered high-rate sensor data for improved performance, estimates of indirectly measured robot parameters (i.e. input output interaction parameters), detection of significant changes in robot dynamics (e.g. velocity, heading, rotational velocity and obstacles histogram changes), and the ability to replace failed sensor outputs with estimates (graceful degradation). The most powerful algorithms for sensor data fusion are: **Kalman filter**, **probabilistic approach** and **fuzzy inference systems**. It would be too simplistic to say that one of these algorithms is superior to the others because each one them exhibits distinct properties, strengths and weaknesses. Hence, the optimal solution is achieved by applying the best suited algorithm to the appropriate application.

Kalman filter: The Kalman filter is attractive due to its simplicity and its low computational demand [98, 78, 211, 89, 92]. It is particularly suitable for dynamic mobile systems since typical plants require safe, smooth and robust navigation among obstacles, with minimal maneuvering. To be able to fuse sensor data of a dynamic plant such as a mobile robot under deliberative navigation control of a Kalman filter, first we design a model and an adaptive controller with multiple sensors for unknown operation conditions. Then, the Kalman filter estimates the dynamic states of the robot from the equilibrium-point for that navigation condition and accommodates the controller parameters. The detailed discussion of Kalman filters based analysis is in section (6.2.4), page (90).

Probabilistic approach: Early research work characterized the sensor fusion problem as one of incremental combination of geometric information. Most of these techniques were ad-hoc and did not take the uncertainty in the system into consideration. Therefore, the probability theory, with its inherent notation of uncertainty and confidence, has found widespread popularity and has a good reputation in the multi sensor fusion community. Concerning this matter, several algorithms have been proposed by various researchers and these algorithms can be classified into major broad categories; Bayesian networks, Markov chain, Monte Carlo, evidence theory and particle filters.

The first work on incorporating uncertainties in an explicit manner in sensor fusion was performed by Smith and Cheeseman [183]. They proposed the use of Bayesian estimation theory and derived a combination function that resembles an equivalent form of Kalman filter. Since then, the same approach has been adopted efficiently by many scientists; Burgard, Elfes, Mojaev, Moravec, Thrun and Zell to underlie mapping and localization capabilities of mobile robots [53, 54, 129, 144, 55, 198, 139, 140].

Fuzzy logic: The operation of an autonomous mobile robot in an unstructured real world with unknown terrains requires consideration of multiple issues. Principally, the sensor fusion system must be able to operate under conditions of imprecision and uncertainty. To cope with these difficulties, the system must be able to respond reactively to unforeseen events as soon as they are perceived. This can be achieved by decomposition of the overall sensor fusion system into a number of simple units, called rules. If we also want to be able to operate under conditions of imprecision and uncertainty the intelligence of autonomous robots needs to use fuzzy logic.

The universe of discourse, membership functions, fuzzification and defuzzi-

fication techniques gives fuzzy logic system surprising facilities to ease applying them to multi sensor fusion efficiently. Many robotics scientists employed fuzzy logic to mapping, localization and multi sensor fusion such as; Surmann, Fukuda, Borenstein, Saffiotti and Zimmer [220, 1, 75, 203, 50].

We combine both fuzzy logic and stochastic Kalman filters to fuse the sensory information underlying the adaptive guidance system. The integration of both approaches succeeds to provide autonomous mobile robots with more safety, smoothness, adaptation and robustness. The details of this subject are found in chapter (6), page (85).

2.4 Integration Scheme Design

2.4.1 Overview

Developing an integration scheme to control multiple sensors and actuators is not an easy task. Different parameters must be taken into consideration to provide a robust, stable, reliable performance. Some of these parameters are: **1)** Synchronization, **2)** Organizing events and callback routines, **3)** Emergency precautions, **4)** Task priorities, **5)** Scheduling of duties, **6)** Distribution of local and remote tasks **7)** Noise filtering and **8)** Reduction of energy consumption.

2.4.2 RWI-B21 Robot Platform

The RWI-B21 robot platform has been used to achieve the presented research work. This platform, shown in figure (2.1), incorporates mainly 2 computers, sensors and actuators. Its sensors include: 24 Polaroid sonar sensors, one 2D Sick-LMS-200 laser scanner, 2 Sony CCD cameras, 52 IR sensor, 52 bumper switches, a KVH-C-100 fluxgate digital compass, rotational encoder, 4 programmable button switches, battery charge tester, microphone, speedometer and translation encoder. These sensors are interfaced to 2 computers via special purpose digital conditioning units. RWI-B21 actuators include 4 direct current (DC) motors, robot arm and pan-tilt unit, the translation of the robot is derived by 3 motors while the fourth one is used to rotate the turret and robot wheels. The software structure is an object oriented OO based C++ programming. Robot sensors and actuators, mounted on the system, are programmed using special software servers and libraries (Beesoft), while robot data exchange obeys local area network (LAN) protocols.

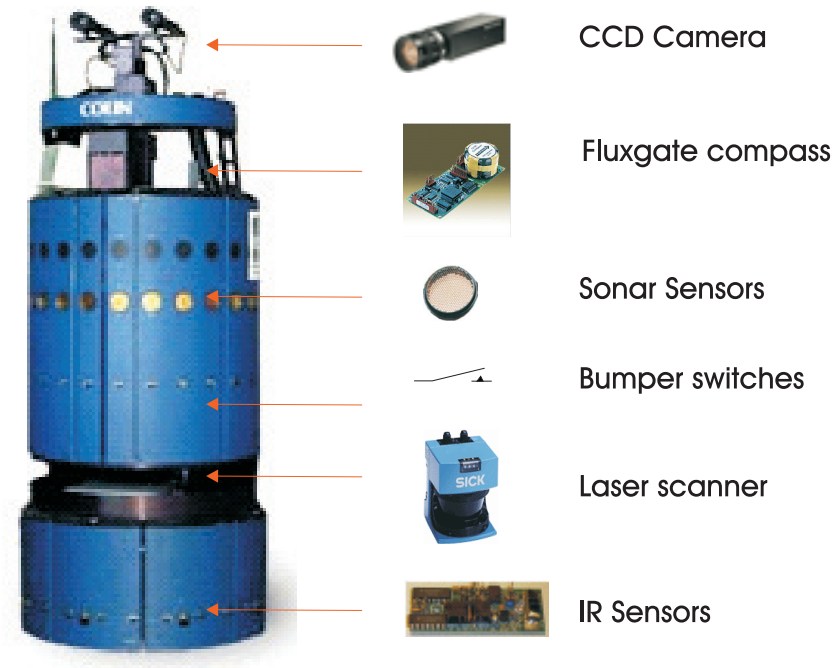


Figure 2.1: RWI-B21 robot platform

2.4.3 MVC based Software Development

The original Model-View-Controller (MVC) design pattern was developed using the Smalltalk programming environment for the creation of user interfaces. It is a popular object oriented (OO) pattern and is often referred to in the literature. The goal of the MVC design pattern is to separate the application object (model) from the way it is represented to the user (view) from the way in which the user controls it (controller) [177, 106, 184].

Throughout this research work we extend the simple idea of the MVC to a more general and robust conception to design the software for large scale systems e.g. autonomous mobile robots, see figure (2.2). Therefore, the original definition of model, view and controller will be accommodated to suit sensor integration and robot data manipulation. According to the new definition of the MVC rule, its 3 component will be:

Model: represents the analysis of the system under study. In this case the model is expressed by the ARX paradigm, state space model and universe of discourse.

View: is the graphical user interface (GUI), responsible for man robot interface.

Controller: is implemented by designing an adaptive controller, communication facilities and management techniques used to steer the robot autonomously.

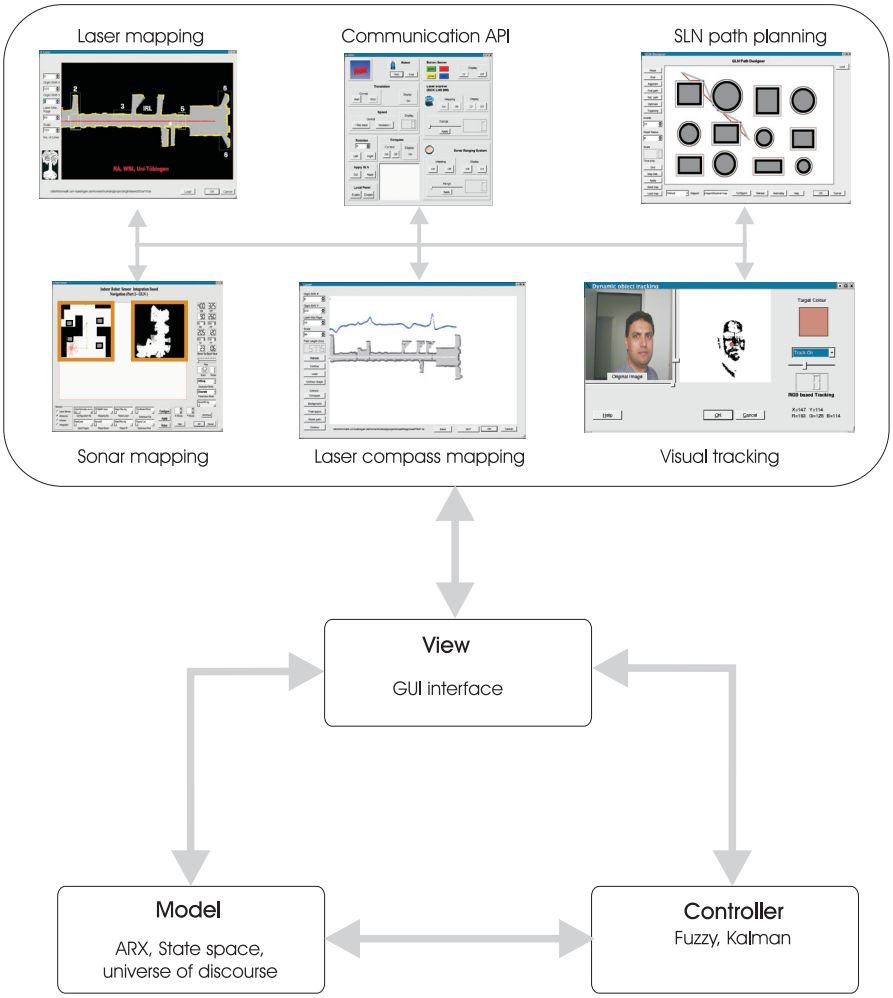


Figure 2.2: MVC software structure

Chapter 3

Sonar Laser Integration

Time of flight (TOF) sensors emit a signal or pulse which is transmitted through a medium. The transit time through the medium is measured electronically and the distance traveled can be deduced. Sonar ranging modules and laser range finders rely on this principle to measure the range between a robot and its surrounding obstacles. This chapter, therefore, draws out the design of a sonar laser integration scheme. It introduces the principles of sonar physics, a brief description of transducer types, the time of flight ranging concept, sonar 3D distribution, strengths, weaknesses and sonar mapping capabilities. Concerning laser ranging, this chapter focuses on laser characteristics, laser ranging modes, benefits, drawbacks, vector mapping paradigm, using model reference for odometric error correction and laser signature matching. Finally, the integration scheme of both laser and sonar is presented.

3.1 Sonar Ranging Modules

Sonar is efficiently employed in different applications such as medical imaging, nondestructive tests and ranging systems. The main purpose of using sonar sensors in robotics is to enhance ranging and environment modeling capabilities of robots. The acronym SONAR stands for SOund NAvigation and Ranging. Sound generated above the human hearing range (typically over 20 KHz) is called ultrasound or sonar. Although sonar behaves in a similar manner to audible sound, it has a much shorter wavelength. This means it can be reflected off very small surfaces. The acoustic spectrum shown in figure (3.1) breaks down sound into 3 ranges of frequencies. The ultrasonic range is then broken down further into 3 sub sections. There are 3 types of sonar transducers; electrostatic, piezoelectric and magnetostrictive.

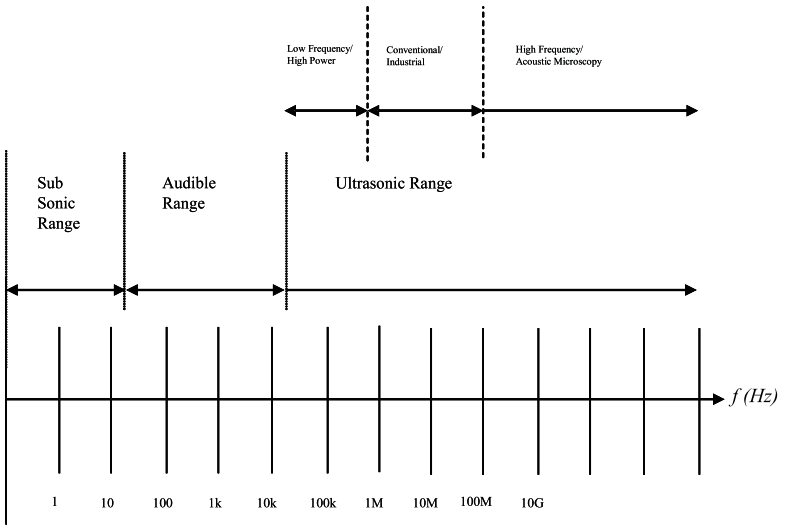


Figure 3.1: The acoustic spectrum

3.1.1 Relevant Research Work

In Mobile robot mapping and Navigation, there are many examples of research using sonar ranging systems with good results, e.g. Elfes 1989 [144, 53, 54], Leonard & Durrant Whyte 1992 [114], Fox et. al. 1997 [55], Borenstein & Koren 1995 [31]. An improvement scheme of mapping using integration of multiple sonar sensors over time based on the Bayesian rule is also done by Thrun 1996 [198] and (Mojaev, Zell 1998) [140]. Another improvement technique using triangulation algorithm has been introduced by Wijk, 1998 [56] to get better results using sonar sensors.

3.1.2 Sonar Transducer Operation Theory

In robotics electrostatic sonar transducers are becoming widely used because of their suitability for elastic low-density media (air). The advantage of this type of transducer is that (if designed correctly) it can emit and detect a wide range of frequencies. Most robotics applications employ the Polaroid sensor which uses a large electrostatic transducer for its sensing. The major disadvantage of the electrostatic transducer is the power needed to generate a signal. Electrostatic transducers require voltages across the plate in the order of hundreds of volts. Moreover, the instantaneous current needed to produce sound is also large. Figure (3.2) illustrates the structure and the measurement principle of a common electrostatic sonar transducer, namely a Polaroid 6500 sonar sensor. This sensor is mounted on the B21 robot platform used throughout this work, also it is already

mounted on a wide variety of platforms. As with many other sonar devices, this sensor works as both transmitter and receiver. When transmitting, a sound wave (mechanical wave) is emitted with a frequency f=49.4 kHz. The wave is generated from a thin foil, which transforms electrical energy into sound waves and conversely, transforms sound waves into electric energy when receiving an echo [135].

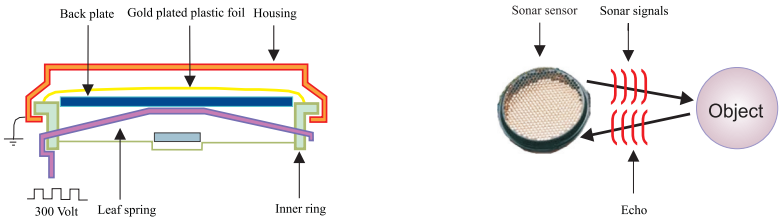


Figure 3.2: Sonar sensor structure and TOF principle

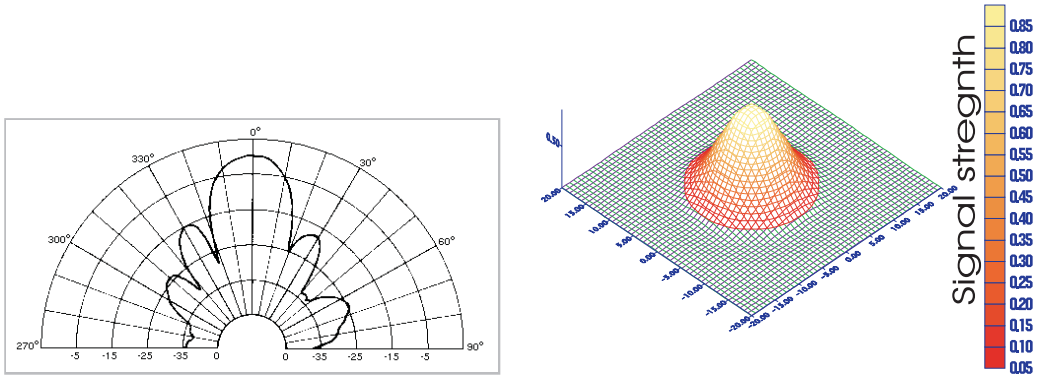
The foil is plastic with a conductive gold coating on the front side, and is stretched over a metallic (aluminum) back plate together. The back plate and foil are functioning as an electric capacitor. When charged with an electric square wave signal (300 V peak to peak), an electrostatic force is exerted on the foil, forcing it to vibrate in the same manner, thus a square sound wave with a frequency of f = 49.4 kHz will be generated. The pulse is 56 cycles long and is transmitted during 56/49.4 = 1.13 ms.. The emitted sonar propagates through air with the speed of sound, which varies with local fluctuations in temperature, humidity, air pressure, and the aerodynamics. In an indoor environment these fluctuations are negligible, and hence the wave propagation speed is close to a constant of 340 m/s. The transmitted signal power S_{trans}, from a sound wave point source decreases primarily as $1/r^2$, where r is the distance of wave propagation. There is also an exponential loss in signal strength, due to attenuation or absorption of sound in the transmission medium (air), i.e.

$$S_{trans} = \frac{e^{-\alpha r}}{r^2} \tag{3.1}$$

where α is an absorption coefficient. According to the sonar ranging system most objects scatter a sound wave randomly, such that the echo power decreases again with the same factor through the back reflection. To compensate for the decrement in signal power, the sonar receiver amplifies the echo differently depending on the elapsed time since the sound wave was emitted. The amplification function is implemented as discrete gain steps that approximate the ideal curve, namely the inverse of the attenuation function. For echo detection, a simple threshold is applied to the amplified signal. It should be emphasized that the power of

the transmitted sound wave is not evenly distributed in space. In figure (3.3.a) a normalized energy beam pattern from a sensor ranging module is presented. The left hand side graph is the simulation of the characteristic radiation function and the right hand side graph is typically described in Polaroid sonar manuals. The pattern in the plot is actually an approximation of the true beam. A pattern obtained from modifying the transducer as a plane circular piston and the characteristic radiation function $I(\Theta)$ can be described by [56]:

$$I(\theta) = \left(\frac{2J_1 \left(K r_p \sin(\theta) \right)}{k r_p \sin(\theta)} \right)^2 \tag{3.2}$$



(a) 2D distribution of sonar power

(b) The characteristic radiation function of sonar in 3D

Figure 3.3: Sonar 2D and 3D distribution

Where J_1 is a Bessel function of the first kind, $r_p = 19$ mm is the piston radius, and Θ is the azimuthal angle. The wave number $k = 2\pi f / S_s$ is obtained from the sound wave frequency f and the speed of sound S_s. Figure (3.3.a) and (3.3.b) shows a pattern obtained from the transducer as 2D and 3D conic distribution. In the beam pattern plot figure (3.3.a), it is seen that there is a main lobe about 39° wide. In addition to the main lobe there are side lobes, which are much smaller. In practice, echoes from a sonar sensor have a high probability of originating from the main lobe. Usually only the closest side lobes are strong enough to trigger detections in the receiver, but then the reflecting object must be of large size, for instance a wall. Therefore most attempts to model sonar sound waves interacting with an object only consider the effect of the main lobe in the beam pattern, see figure (3.3.a). To minimize the number of false echoes , the sonar ranging sensors are equipped with a tuned LC-circuit that filters out all frequencies but those closest to the transmission frequency. By computing the elapsed time Δt

between when the sonar is emitted and received, it is straightforward to calculate the distance r from the sensor device to the object that the wave interacted with:

$$r = \frac{S_s \Delta t}{2} \tag{3.3}$$

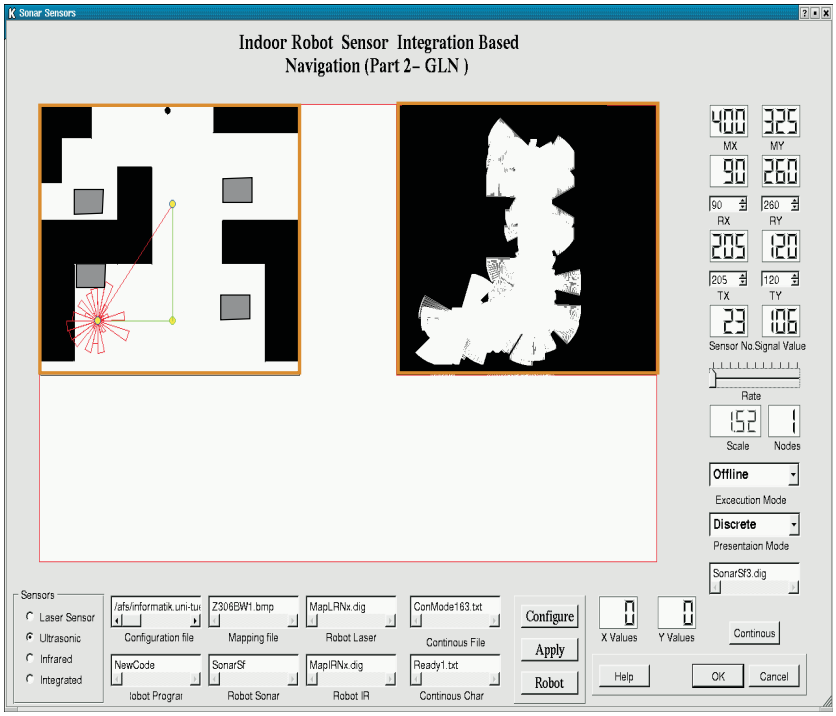


Figure 3.4: Sonar mapping program

The division by two is because the wave has traveled twice the distance between the sensor and the obstacle. When understanding the simple principle of measuring distances to objects with sound waves, the sonar seems to be an attractive sensor. Figure (3.4) displays a representation of a working environment using sonar sensors mounted on a B21 robot platform. The reflection of the sonar depends on the nature of the object surface that the sound wave interacts with. If the surface is rough, i.e. the pattern thereof has a size larger than the sound wavelength $\lambda = S_s/f = 6.95$ mm, then the wave can be reflected back to the sensor even at a large angle of incidence. However, if the surface is smooth, it will act as a mirror following the reflection law. In that case the main lobe of the sonar must be directed not more than 39/2=19.5^o off from the surface normal in order to produce an echo.

The preceding discussion focuses on Polaroid 6500 sonar sensors mounted on the RWI-B21 robot used in our experimental work. More information about piezo-electric sonars can be reviewed in [3].

3.1.3 Weaknesses of Sonar Sensors

Mapping with the sonar ranging system is not sufficiently precise. This is due to various disadvantages that have given the sonar sensors a bad reputation when applied in indoor environments:

- With the sonar as a sound wave traveling in the transmission media (air), it is absorbed and hence the speed goes down. This leads to an increment of the estimated distance more than the actual value.

- Ideally, the sonar would reflect directly from the surface of the obstacle back to the receiver. In the real world the sonar reflects multi time or gets specular reflections before returning back to the receiver, so the estimated distance is normally larger than the real one.

- Normally, the robot carries multi sensors, so several sonars send out identical sound waves, hence a cross talk problem emerges. Giving each sonar a different sound wave signature can solve this problem. This is normally based on the hardware design of the sonar ranging modules.

- Sonar sensors have a limited measurement range ($\approx$ 0.30:6.5m) and an angular direction ($\approx 9 : 39°$). Outside the defined working range the acquired results will tend to be uncertain.

- Data loss can frequently happen if the reflected sonar is either too weak to be detected by the receiver or it is reflected out of detection view range.

- The fluctuation of temperature, humidity, and pressure has an influence on the speed of the sound wave, hence the estimated distance to any obstacle tend to be imprecise. In the indoor ambient these variations can be neglected.

- Compared with the laser range finder, the sonar ranging module seems too slow. In a sonar system most of the time is wasted, waiting for the pulse to return back. This comparison is based on the speed difference between sound and light.

3.1.4 Advantages of Sonar Sensors

Despite the mentioned disadvantages, sonar sensors are widely used in mobile robots because of several positive features; **1)** Sonar sensors are relatively cheap ($\approx$ \$10 for each sensor) compared with the laser range finder. **2)** Sonar sensors are light and small, so they can be mounted in a flexible manner on the robot's body. **3)** In some applications the large opening angle is considered as a constructive factor e.g. in integration of sonar and laser ranging. This feature enables robots to detect objects which are unreachable by laser scanner [6]. **4)** Sonar sensors consume low power ($\approx$10 mw). So, they are not considered as a load on robot batteries during navigation.

3.2 Laser Range Finders

The term laser stands for Light Amplification by Stimulated Emission of Radiation. Laser light is an unusual form of light, which does not occur in nature. It exhibits both high coherence and monochromaticity. The coherence of light means that the electromagnetic oscillations of the photons are in phase. In practical terms, this means that laser light behaves like one big wave and exhibits wave-like properties such as interference and diffraction.

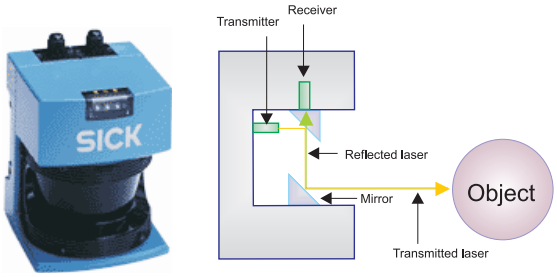


Figure 3.5: Sick LMS 200 scanner and the TOF ranging principle

The monochromaticity of laser light means that the photons have nearly the same wavelength (or, equivalently, energy). Several methods of laser range finding have been developed, including; **1)** Time of flight (TOF) measurement, **2)** Phase shift measurement, **3)** Triangulation, **4)** Absolute interferometry. Each method is suitable for different measuring ranges and conditions. In the thesis the TOF ranging is described while the other ranging methods are explained in [3].

3.2.1 TOF Laser Ranging Principles

Even the fastest photon requires a certain period of time to cover the distance from the sensor to the target and back. This time is directly proportional to the distance traveled, taking into account the velocity of light in the medium involved, which may be easily derived from the velocity of light in a vacuum. The cost and complexity of this method depends upon the precision and resolution required. Data acquisition and analysis electronics must cope with ns and sub-ns

time scales: decimeter ranges may be easily resolved by nanosecond pulses but precision in the millimeter and sub-millimeter range requires pulse lengths of a few tens of picoseconds and the associated electronics. Clearly, a poorly resolved pulse will lead to uncertainty in the accuracy of the measurement. The standard deviation in measured distance is proportional to the optical pulse rise time and is inversely proportional to the signal-to-noise ratio. For example a low-cost avalanche photodiode (APD) enables a fourfold increase in range compared with a PIN photodiode, and a high-end APD provides a ten-fold increase for the same laser power. Suitable emitters are available only in the infrared (e.g. at 850 and 1650 nm). The rotary motion of the mirror does not allow the laser beam to be condensed on a definite spot for a long time. This means that the beam is invisible to the eye but also ensures that, using the pulse lengths necessary for good performance, the system is eye-safe despite the high peak pulse power. [164, 160, 207] (in German).

To gather the odometric data of an environment, the laser range finder returns a set of points corresponding to the intersection points of the laser beam with the obstacles. The laser beam rotates in a horizontal plane and emanates from a scanner mounted on the robot. The scan is a 2D slice of the environment. The Sick LMS 200 laser range finder is mounted on various types of platforms. A comparison between SICK and AccuRange laser range finders has been introduced by A. Scott et.al. (2000) [176]. The measurement using laser range finder depends on TOF measurement system principles. In a TOF system a short laser pulse is sent out and the time Δt until it returns is measured. The ranging principle is thus the same as for the standard sonar sensor.

$$r = \frac{l_s \Delta t}{2} \tag{3.4}$$

Where l_s is the speed of light (3x10^8 m/s) and Δt is the round trip time. In order to realize such a system, a high precision means for measuring time is needed. Thinking in terms of a robot application a range resolution in the order of centimeters is desirable. One advantage with the short pulses is that higher levels of powers can be used, giving better range coverage, but still keeping a high safety level and low power consumption. Most of the commercial laser range finders measure the distance in a single direction. By mounting it on a rotating body, a scanning effect can be achieved. Instead of rotating the whole sensor, there are now commercially available laser range finders based on a rotating mirror, which can cover a large field of view up to 180°. Many researchers have used this kind of scanner on different platforms. The Sick LMS 200 laser range finder and the TOF measurement principle are presented in figure (3.5). Compared to the sonar sensor, the laser range finder is very expensive and price vs. performance must be

weighted. It is possible to add one more degree of freedom and let the sensor scan vertically as well, yielding a 3D scanning device.

3.2.2 Strengths of 2D Laser Range Finders

The 2D laser range finder is efficiently used in several robotics applications due to the following reasons:

- They are fast ($3x10^8$ m/s speed of light and up to 500 kb/s transmission speed), i.e. the measurement can in most cases be considered as instantaneous. This means that, one does not have to think about compensating for the motion of the platform between sending and receiving. What stops laser range finders from working even faster is the mechanics. When using laser, the problem is instead to be able to rotate the scanning device fast enough and at the same time very accurately.
- The range accuracy is fairly good as reports to have a standard deviation of the new generation of laser range finders from Sick of 10 mm and the angular resolution is down to 0.25°.
- The angular resolution is far better with the laser range finder than with sonar sensors. The resolution for the Sick LMS 200 laser range finder has an angular resolution of $180{:}0.5^\circ/1^\circ$, which is an order of magnitude better than for the sonar.
- The data from the laser range finder can be interpreted directly from the range to an obstacle in a certain direction.

Due to these points, the laser range finder is intensively used alone or integrated with other sensors to improve the mapping, localization and navigation capabilities of mobile robots. Successful examples of this integration are laser-sonar [6], laser-compass [7, 152] and laser-vision [2].

3.2.3 Problems of 2D Laser Range Finders

The use of 2D laser range finders has some limitations, they are:

- The scanner provides range information limited to a plane, which means that only a 2D intersection of the 3D world can be sensed. To be able to get information about other parts of the environment, the sensor has to be moved. Another possible solution is to mount the sensor so that it intersects the world under an angle and instead move the platform. This angle becomes a design parameter for the system, which will determine the range of the scanner.

- The laser range finder consumes more power (17.5 Watts) compared with other sensors such as sonar.
- The laser range finder is still very expensive. The price will drop and the scanner will become more widely available if an application is found that will motivate mass production of it.
- Some materials appear as transparent for the laser, such as glass and others that can absorb the laser producing no reflection. Therefore, we combine the laser range finder with a sonar ranging module in order to reinforce the robustness [6].

In figure (3.6) we define some special marks (denoted from 1 to 6). The marks 1, 5 and 6 are used to sign the position of the glass doors. The laser range finder failed to detect them due to their transparency. The area marked by 2 is the downward stairs and should not be entered with the B21 platform. The horizontal plane laser range finder is not able to detect downward stairs. The area marked by 3 is a table and the laser range finder was able to identify just the legs of the table but the table itself wasn't recognized. The object marked by 4 is a dynamic object (door) and the presented location is not permanent. Based on the preceding discussion an intelligent integration between the laser range finder and the sonar sensors has been developed to avoid such disadvantages. To overcome disadvantages of both laser range finders and sonar ranging modules an intelligent integration scheme can be implemented to improve the overall system robustness [6].

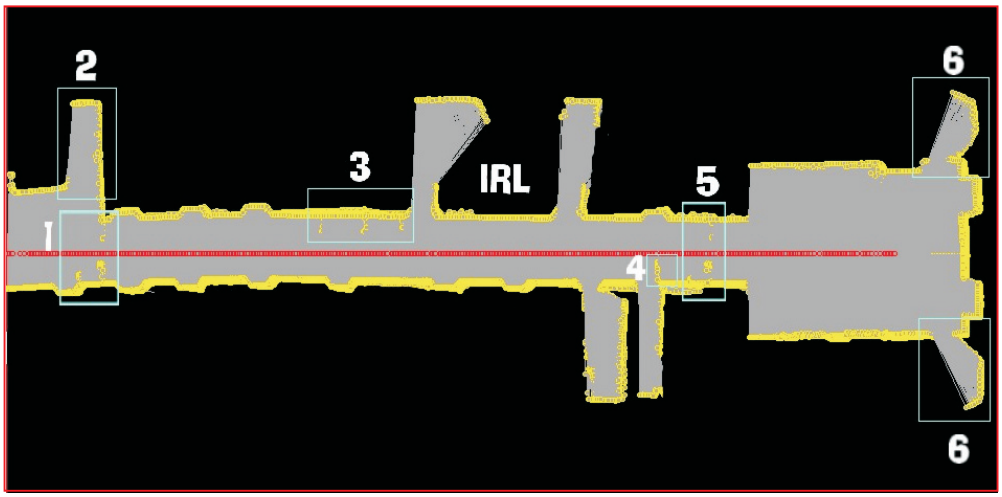


Figure 3.6: Mapping using 2D laser range finder

3.2.4 Vector Mapping Paradigm (VMP)

Several mapping paradigms have been investigated to model the robot's working environment. The most widely used methods are the occupancy grid, the topological graph and maps integrating both. Each of these methods has its characteristics, advantages and disadvantages. Throughout this work the vector mapping paradigm (VMP) has been used. This technique is suitable for robots equipped with a laser scanner. The vector mapping is originally based on the occupancy grid with a reduced map size. The main idea of the vector mapping is to determine an empty space as a region of cells between the laser range finder (robot's position) and the detected obstacles. In other words, these empty cells will be represented by end points of the laser rays (vectors) and the position of the laser range finder. The measurements of the vector mapping are given in a polar coordinate system whose origin is the position of the scanner (robot), while the end of the vectors are the obstacle boundaries. This map reduces both computer power and memory requirements. In this method, it is required to record just the start (once per scan) and the end of the vectors so the region in between is considered free space that will preserve the consistency of the maps. The laser range finder sends out 180 laser rays per scan with 1° angular resolution. The resolution of the vector mapping can be controlled using either hardware or software techniques. In the case of VMP, the robot can map its environment from central measurement points covering all the empty area (visible region). The vector mapping technique can be easily converted to the traditional map paradigms. For example, the occupancy grid can be generated by marking all cells included in the vector region by null (empty) and the remainder cells by one (occupied). Also the topological graph and integrated form can be estimated through the occupancy grid. Based on vector mapping the contour-graph can be generated. This graph includes the contours of the existing obstacles. This graph is used later in the path planning and navigation based on the SLN-algorithm [6]. Enlarging the obstacle's boundaries is used to generate the contour graph. The minimum extension of obstacle boundaries is the sum of the threshold distance and the robot radius. This extension can be enlarged to get a contact in the middle of the free space (Voronoi diagram) if the obstacles are near and can be decreased to the minimum value if the obstacles are far away. Figure (3.7) presents four different models based on vector mapping: the first one has been built using 500 measurement points and the last one using only 10 points. The difference in precision between the two models seems little although the map size difference is clearly large: the laser map (3.7-d) size is about 2% of the first map size (3.7-a). This sort of mapping has a flexible resolution format depending on the required precision and the computational power of the system. Adjusting the number of the central measurement points can control the resolution of the map. The vector mapping has the advantages of the traditional

mapping techniques: precision, consistency and efficiency with a low storage size [6].

3.2.5 Model Reference

The acquisition of the odometric data is associated with different types of noise e.g. non-linear noise (backlash hysteresis, toggle, threshold, saturation, damping and friction), time based noise (accumulation effect) and white noise. Another form of odometric errors is generated by the mechanical system from rotation or due to slippage and drifting, see figure (3.8).

In outdoor applications the global positioning system (GPS) is widely used as a localization system with a certain (large) tolerance. In indoor applications similar global positioning systems are not widely used or available, so various algorithms have been developed to correct the maps and to obtain precise models of the environment. The model reference method is considered as a plausible solution to correct these erroneous maps. This method can easily be implemented and generates high precision maps. The overall map is an assembly of segments arranged corresponding to a model of nodes and paths (reference model), see figure (3.9). The model reference method presents an accurate solution for the correction of odometric data errors based on direct learning. That means a model of nodes and paths is used to correct the maps. Another technique has been used by Thrun et. al. 1997, which depends on the training (indirect learning) of neural networks.

3.2.6 Related Mapping Paradigms

Metric maps, topological graphs and The integrated form are the most famous mapping paradigms. The following is a brief description of these methods.

Metric Maps: Major interest in metric maps was generated with the publishing of Moravec and Elfes's 1985 paper, High Resolution Maps from Wide Angle Sonar. Their approach involved taking range measurements from a fixed array of sonar sensors arranged in a circular fashion whose position and angle in relation to each other is known. This range information is then translated onto a two-dimensional map by making the assumption that one of the points along the curved end of the arc is occupied by an obstacle, and incrementing the probability that all points along the curve are occupied. The model they developed for this map was a tessellated spatial random field called an Occupancy Grid. Conversely, if it can be assumed that a certain range reading means that an obstacle is in the beam's path at that distance, all points in the body of the beam can be assumed to be free of

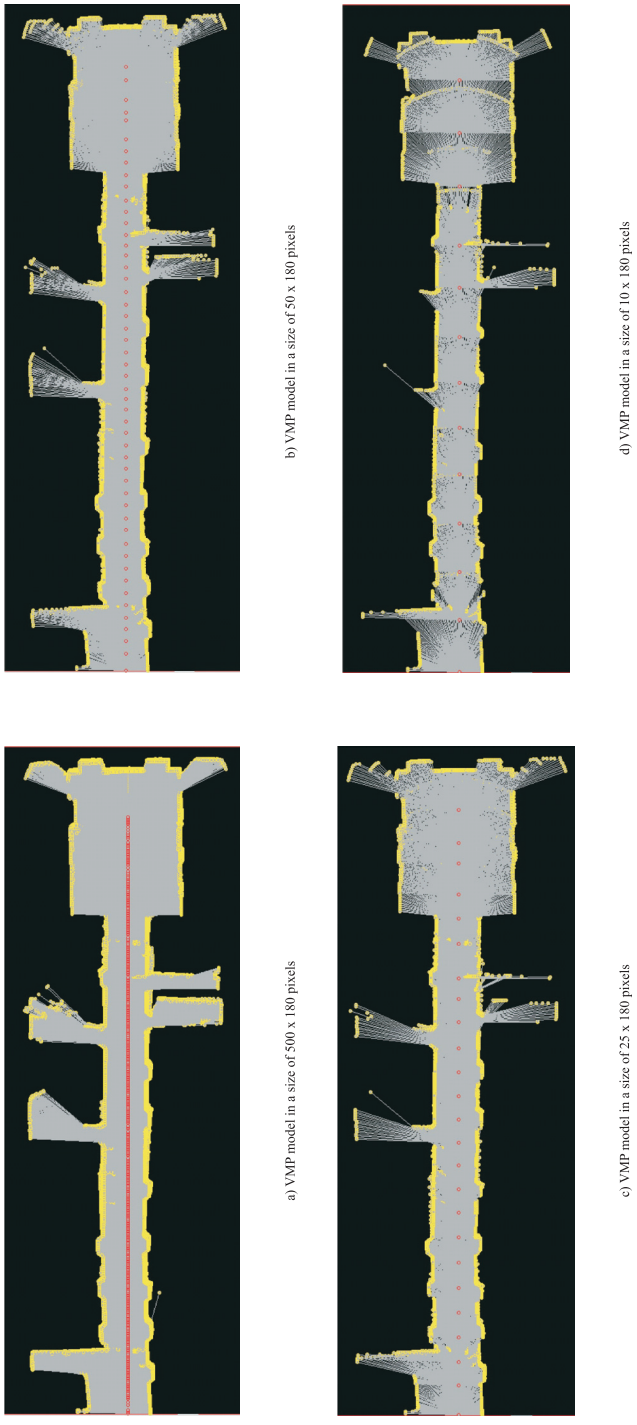


Figure 3.7: Different VMP resolution models

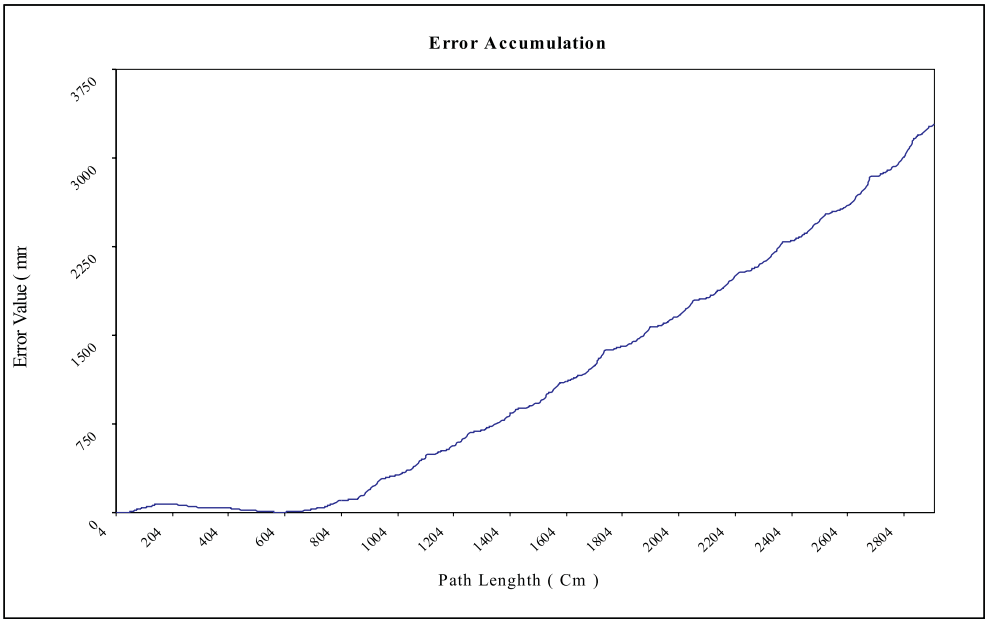


Figure 3.8: Odometric error

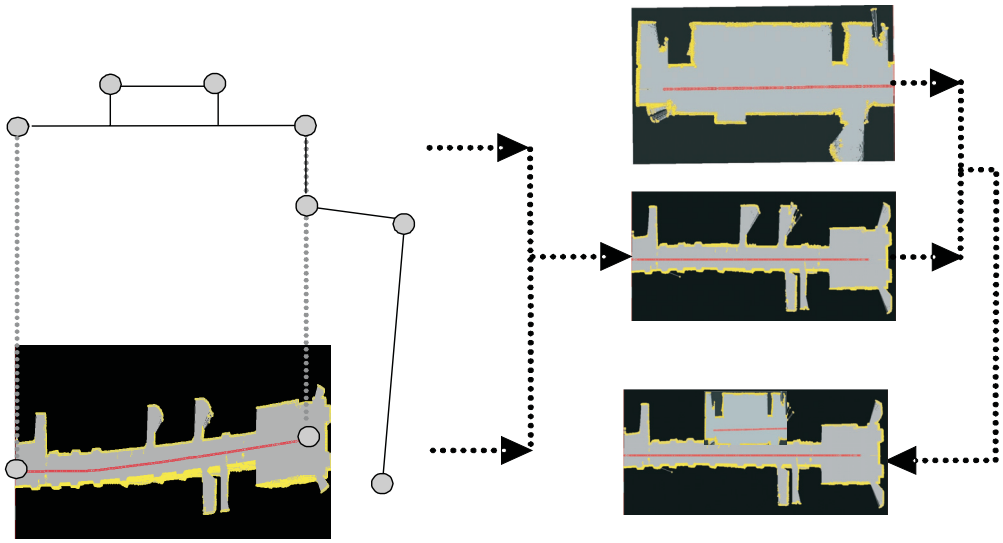


Figure 3.9: Model reference

obstacles, and their probabilities of occupancy decremented appropriately. While a single sonar reading yields very little information, each one changes the map slightly, and with a reading every 100 ms, a strong confidence grid is quickly generated.

Topological Graphs: Metric and topological maps each have their strengths and weaknesses. Metric maps are relatively easy to generate, topological maps are not. Metric path planning is slow, topological planning is fast. Thrun proposes a 5-step method of generating a topological map from a metric map. Firstly, it must be decided which cells are occupied and which not. This is done with thresholding, where all cells whose occupancy values are below a certain value are said to be empty, and all those above it are said to be occupied. Secondly a Voronoi diagram is generated. A basis point is the closest occupied cell to the unoccupied cell. In a simple corridor-like environment, this would lead to a line being drawn down the center of the corridor. The next two steps are to identify critical points and critical lines.

Integrated Form: Sebastian Thrun et. al. advise combining the two in order to maximize the positive attributes of each [199]. These include the grid-based map's ease of construction and usefulness in self-localization, and the speed at which it is possible to plan a path using a topological map allied with its compactness. This involves building a metric map, then converting it to a topological map.

3.3 Matching of Laser Signatures

Pose tracking is a key subject in robotics. Mobile robot pose tracking is the process of deducing a robots pose (location, heading) relative to its environment from its sensor data. Therefore, it has been referred to as the most fundamental methodology to provide a mobile robot with autonomous capabilities. Using odometry encoders is the most widely used method for determining the position of a mobile robot. Unfortunately, the acquisition of odometric data is associated with different types of noise e.g. non-linear noise (backlash, hysteresis, toggle, threshold, saturation, damping and friction), time based noise (accumulation effect) and white noise.

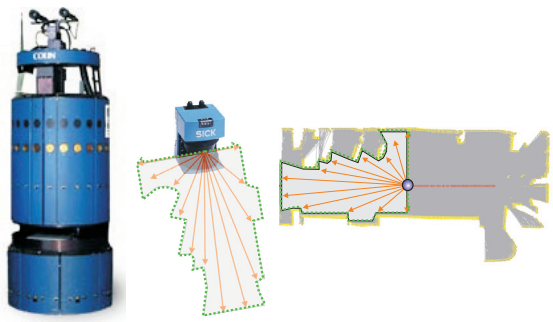


Figure 3.10: Laser signature matching

Another form of odometric errors is generated by the mechanical system from rotation or due to slippage and drifting. Consequently, the odometry needs to be reset from time to time, as the VMP algorithm [6]. To overcome this problem, robust and accurate localization of the robot is needed. A variety of algorithms have been used to solve the problem of robot localization e.g. expectation maximizing, maximum likelihood, probabilistic approach (Markov, MCL) and concurrent localization and mapping (SLAM, Kalman) [139, 147, 67, 44, 115]. Also, there are many researchers, who adopted vision systems for the recognition of natural or artificial landmarks. Those techniques often require high-performance onboard computers and expensive vision systems as well as computational overhead. Moreover, those systems incorporate a lot of instability, complexity and noise. Throughout the coming discussion we will study techniques to estimate the pose based on the spatial domain, spectral analysis, wavelet and the least mean squares (LMS) of laser signatures, see figure (3.10).

3.3.1 Relevant Research Work

Lu and Milios proposed a solution for laser scan matching of points and tangents using least squares [120]. Mota and Ribeiro employed the maximum likelihood algorithm to match 2D laser scans in 3D reconstructed models. Bengtsson and Baerveldt presented a scan matching algorithm, IDC-S, Iterative Dual Correspondence Sector, which deals with changes in the environment by dividing the scans in several sectors, which are matched separately [19, 20]. Gutmann and Schlegel combined the approach with the point to line matching of Cox [115, 81, 79]. Both are iterative methods, i.e. they need a relatively large amount of processing time. Therefore, they are used as offline algorithms after all distance data has already beenacquired. In the method of Weiss et. al., histograms are used as a base of scan points matching [210]. Röfer (Bremen Autonomous Wheelchair) extended the method of Weiss et. al. to get faster matching [167].

Concerning the activities of the laboratory for autonomous robots at the University of Tübingen, several pattern matching algorithms have been investigated for robot self-localization. Feyrer and Zell developed an integration scheme of visual and laser signatures in real-time for persons pursuing [61, 63, 62, 60, 59]. Mojaev and Zell incorporated scan points into occupancy maps by which these local grids were matched to generate a global map [140, 139]. Aboshosha, Tamimi and Zell employed spatial, spectral and wavelet techniques to analyse and match both laser and geomagnetic signatures [7, 1, 2, 6]. Another technique has been introduced by Biber and Straßer to match 2D laser scans relying on the normal distribution transform [24].

3.3.2 Spatial Analysis (ED - CCA)

Spatial domain analysis is an efficient method for data series matching. This method can be implemented by deducing the Euclidean distance of two series: x_i and y_i (Eq: 3.5) or the cross correlation of both of them (Eq: 3.6).

$$ED(x,y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{3.5}$$

The correlation between two signals (cross correlation) is a standard approach to feature detection as well as a component of more sophisticated techniques. It is well known that cross correlation can be efficiently implemented in the transform domain, the cross correlation preferred for feature matching applications does not have a simple frequency domain expression.

Cross correlation is a standard method of estimating the degree to which two series are correlated. Consider two series x_i and y_i where $i = 1, 2...N$. The cross correlation ρ at delay τ (signature phase shift) is defined as:

$$\rho(\tau) = \frac{\sum\limits_{i=1}^{N}(x_i - m_x)(y_{i-\tau} - m_y)}{\sqrt{\sum\limits_{i=1}^{N}(x_i - m_x)^2}\sqrt{\sum\limits_{i=1}^{N}(y_{i-\tau} - m_y)^2}} \tag{3.6}$$

Where m_x and m_y are the means of the corresponding series. If the above is computed for all delays $\tau = 1, 2, \ldots N$ then it results in a cross correlation series of twice the length as the original series. Rewrite the equation (3.6) as a function of variance and covariance as follows:

$$\rho = \frac{\mathrm{Cov}(x\ ,\ y)}{\sqrt{Var(x)}\sqrt{Var(y)}} \tag{3.7}$$

Generally, the expectation of the product of two series x and y is given by

$$E(xy) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xy f_{xy}(x,y)dxdy \tag{3.8}$$

There is a special simplification of equation (3.8) that occurs when x and y are independent. In this case, $f()$ may be factored. Equation (3.8) then reduces to

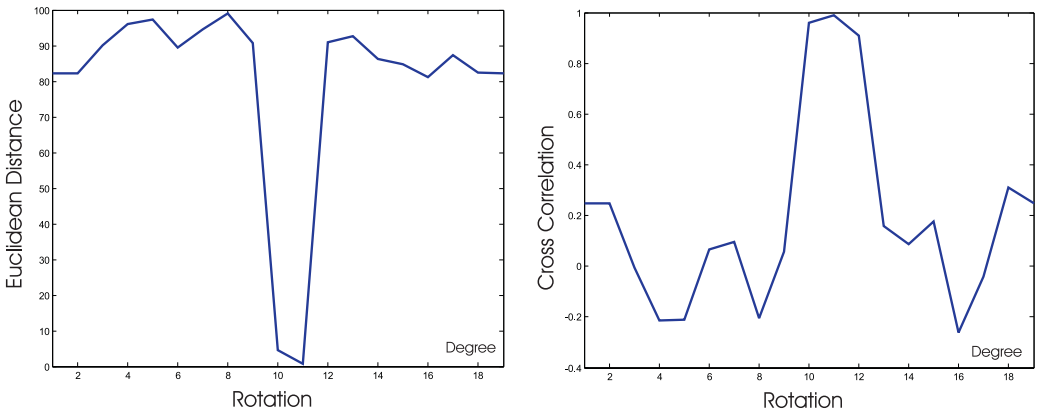
$$E(xy) = \int_{-\infty}^{\infty} x f_x(x)dx \int_{-\infty}^{\infty} y f_y(y)dy = E(x)E(y) \tag{3.9}$$

If x and y possess the property of equation (3.9), that is, the expectation of the product is the product of the individual expectations, they are said to be uncorrelated. Obviously, if x and y are independent, they are also uncorrelated. However, the converse is not true, except in a few special cases.

As a matter of terminology, if $E(xy) = 0$ then x and y are said to be orthogonal. the covariance of x and y is also of special interest, and it is defined as:

$$Cov(x, y) = E[(x - m_x)(y - m_y)] \tag{3.10}$$

The denominator in the expression (3.6) serves to normalize the correlation coefficients. Therefore, the correlation coefficient is a normalized measure of the degree of correlation between two series, and the normalization is such that ρ always lies within the range $-1 \leq \rho \leq 1$ This will be demonstrated by looking at three special cases; **1)** $y = x$: maximum positive correlation, $\rho = 1$, **2)** $y = -x$: maximum negative correlation, $\rho = -1$, **3)** orthogonality: x and y are uncorrelated, $\rho = 0$.



(a) Euclidean distance (b) Cross correlation

Figure 3.11: Signature matching in spatial domain w.r.t. robot rotation

Figure (3.11) demonstrates laser signature matching using spatial techniques. The robot rotates and the reference heading is deduced by matching the measured signature with a stored one. In figure (3.11.a), the best match is the global minimum of the Euclidean distance, while the best match as shown in figure (3.11.b) is the global maximum of the cross correlation.

There is the issue of what to do when the index into the series is less than 1 or greater than or equal to the number of points. $(i - d < 1 \, or \, i - d >= N)$ The

most common approaches are to either ignore these points or to assume the series x and y are zero for $i < 1$ and $i >= N$. In many signal processing applications the series is assumed to be circular in which case the out of range indexes are wrapped back within range, ie: $x(0) = x(N)$, $x(N + 5) = x(5)$ etc. The range of delays d and thus the length of the cross correlation series can be less than N, for example the aim may be to test correlation at short delays only. If the laser signature is continuous there is no need to recycle the signature or to get the delayed portion to zero. Hence, we can estimate the correlation of both signatures directly.

3.3.3 Spectral Analysis (DCT)

The spectral analysis is a powerful technique which studies the signatures based on their frequency components. It is used to convert a time domain signal to a signal in the frequency domain. So, it transforms a function $f()$ that depends on time into a new function $F()$ which depends on frequency. If the spectral analysis of a signal in the time domain is taken we obtain a frequency-amplitude representation of that signal. The spectral analysis tells us how much of each frequency exists in the signal, but it does not tell us when in time these frequency components exist. The spectral analysis is not a suitable technique for non-stationary signals i.e., signals with time varying spectra.

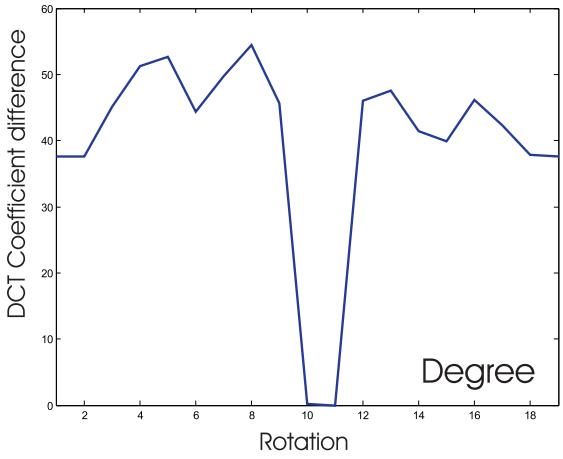


Figure 3.12: Heading deduction based on DCT coefficients matching

Compared with the discrete Fourier transform (DFT), the discrete cosine transform (DCT) is preferred to analyze the laser signature for the following reasons:

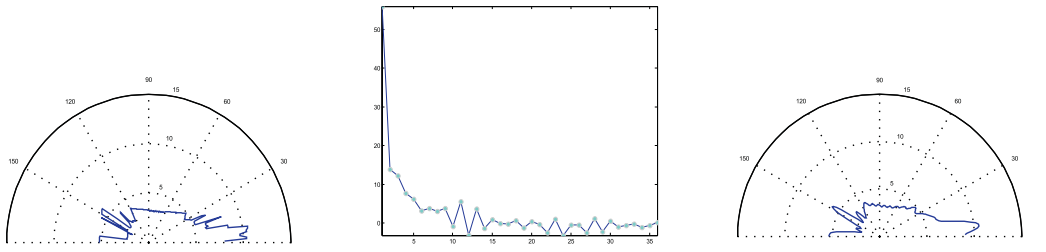
- In case of the DFT, the basis sequences are the complex periodic sinusoidal sequences of the exponential function, and in general the transforms yield a complex form even if the signal is real.

- DCT is an orthogonal transform representation for real sequences. The DFT involves the implicit assumption of periodicity, the DCT involves an implicit assumption of both periodicity and even symmetry.

- Computationally, the DCT is more efficient than the DFT because it is a completely real transform and does not require complex variables or arithmetic.

The distribution of the DCT coefficients within the signature is non-linear. Coefficients at the lower frequencies have large magnitude, while they have less magnitude at higher frequencies. The values obtained using DCT are scaled so as to be represented by 8 bits, this means that it can be implemented in hardware. A large number of bits are assigned to values corresponding to lower frequencies, while fewer bits are assigned to higher frequencies.

Compared with the spatial domain analysis, prevalent in signature matching and robot localization, the frequency domain provides more stability for dealing with pattern recognition subjects, especially in presence of noise, phase shift and scaling errors. Consequently, matching of signatures in the frequency domain facilitates pose estimation even if signatures are partially distorted or in the presence of limited changes due to rotation or displacement. The DCT pertains to a group of algorithms (e.g. DFT, FFT, DST, wavelet, Gabor, etc.) used for the frequency analysis of patterns. Furthermore, the DCT has a compression capability which speeds up matching of thousands of patterns, see figure (3.13).



(a) Original laser signature (b) DCT low coefficients (c) Restored laser signature

Figure 3.13: DCT based spectral analysis

The DCT transforms a signal from a spatial representation into a frequency representation. Lower frequencies contribute more to a signal than higher frequencies, so if we transform a signature into its frequency components and throw away data about higher frequencies we can reduce the amount of data needed to describe those signatures without sacrificing too much signature quality. The DCT transform can be accomplished as follows:

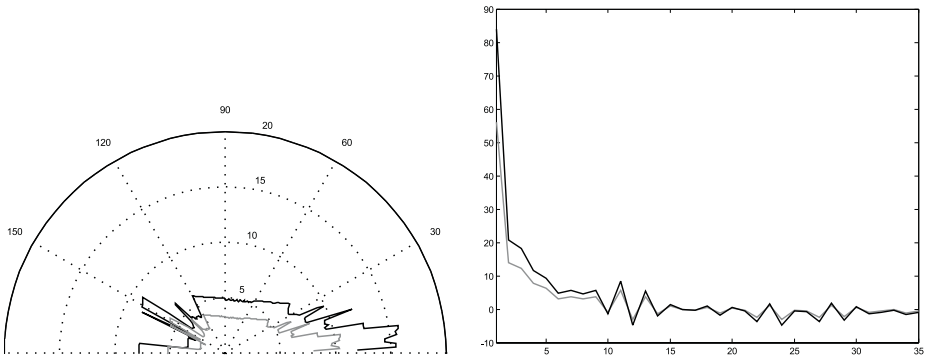
$$y_i = \Lambda_i \sum_{j=1}^{N} x_j \cos \frac{\pi(2j-1)(i-1)}{2N},\ i = 1, \ldots, F \tag{3.11}$$

where x_i is the data series, at time t_i, $x_i \in R,\ i = 1, \ldots, N$, F is the number of DCT coefficients, y_i are the DCT coefficients and N is the signature length. For all frequencies ($F = N$), i varies from 1 to N, there is no compression. To compress the signature (by omitting high frequencies), i varies from 1 to fl ($F = fl$), where fl is the frequency limiter or called the number of DCT coefficients. The inverse DCT can restore the original signature using a limited number of DCT coefficients, see figure (3.13).

$$x_i = \sum_{j=1}^{F} \Lambda_j y_j \cos \frac{\pi(2i-1)(j-1)}{2N},\ i = 1, \ldots, N\, and\, j = 1, \ldots, F \tag{3.12}$$

The spectral analysis has some advantages when applied to pose deduction; first, it can match the signature even in the presence of noise or if the signature is partially distorted. In addition, it can detect signature scaling and rotation.

Scaling Property The spectral analysis has the capability to detect scaling, see figure (3.14).

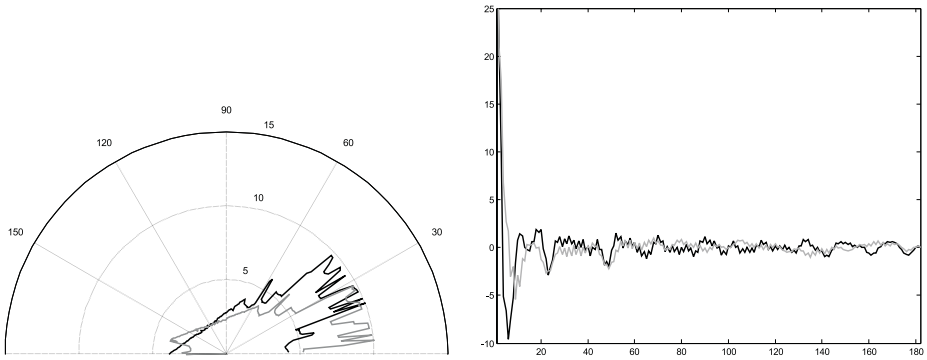


(a) Laser signature scaling (b) DCT coefficients

Figure 3.14: DCT based spectral analysis in presence of scaling

So, If $f(t) \Leftrightarrow F(\omega)$, then $f(at) \Leftrightarrow \frac{1}{|a|} F\left(\frac{\omega}{a}\right)$. The value $|a| > 1$ compresses the time axis and expand the frequency axis, while the value $|a| < 1$ expands the time axis and compress the frequency axis, ω is the frequency, t time and $a,\ t_o$ constants. Figure (3.14) shows the the scaling effect on spatial and spectral representations.

Phase Shift Property Second, the spectral analysis can counteract the influence of limited rotation or displacement. Figure (3.15) shows the effect of rotation on the spatial and spectral representation of a signature.



(a) Laser signature rotation (b) DCT coefficients

Figure 3.15: DCT based spectral analysis in presence of phase shift

The success of the DCT depends mainly on the nature of the signature. If the elementary frequency components of a signature are homogeneously distributed over the spectrum range, the DCT fails to compress the signature. Also, if the shape of a signature is extremely sharp, Lagrange concept, (e.g. square wave, sawtooth, etc.) the DCT precision will be drastically decreased. Figure (3.12) shows how to track a pose under rotation where the best suited pose is the global minimum of differences between the stored and the reference signatures.

3.3.4 Wavelet Analysis

The wavelet analysis procedure is to adopt a wavelet prototype function, called an analyzing wavelet or mother wavelet. Temporal analysis is performed with a contracted, high-frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low-frequency version of the same wavelet. Because the original signal or function can be represented in terms of a wavelet expansion (using coefficients in a linear combination of the wavelet functions), data operations can be performed using just the corresponding wavelet coefficients. And if one further chooses the best wavelets adapted to your data, or truncate the coefficients below a threshold, the data is sparsely represented. This sparse coding makes wavelets an excellent tool in the field of data compression. Other applied fields that are making use of wavelets include astronomy, nuclear engineering, medical applications, signal and image processing, cryptography and sub-band coding,

acoustics-optics applications, neurophysiology, turbulence, earthquake-prediction, radar, human vision, and pure mathematics applications.

The Discrete Wavelet Transform

Dilations and translations of the **mother function** or **analyzing wavelet** $\Phi(x)$, define an orthogonal basis, our wavelet basis:

$$\Phi_{(s,l)}(x) = 2^{-\frac{s}{2}}\Phi(2^{-s}x - l) \tag{3.13}$$

The variables s and l are integers that scale and dilate the mother function $\Phi(x)$ to generate wavelets. The scale index s indicates the wavelet's width, and the location index l gives its position. Notice that the mother functions are rescaled, or **dilated** by powers of two, and translated by integers. What makes wavelet bases especially interesting is the self-similarity caused by the scales and dilations. Once we know about the mother functions, we know everything about the basis. To span our data domain at different resolutions, the analyzing wavelet is used in a scaling equation:

$$\mathrm{W(x)} = \sum_{\mathrm{k}=-1}^{\mathrm{N}-2} (-1)^{\mathrm{k}} c_{k+1}\Phi(2x + k) \tag{3.14}$$

where $W(x)$ is the scaling function for the mother function $\Phi(x)$; and c_k are the wavelet coefficients. The wavelet coefficients must satisfy linear and quadratic constraints of the form

$$\sum_{\mathrm{k}=0}^{\mathrm{N}-1} \mathrm{c_k} = 2, \sum_{k=0}^{N-1} c_k c_{k+2l} = 2\delta_{l,0} \tag{3.15}$$

where δ is the delta function and l is the location index.

Wavelets versus DCT

To discriminate the DCT from wavelets, the following points have to be taken into considerations:

- The standard DCT decomposes the signal into individual frequency components.
- The DCT basis functions are infinite in extent.
- DCT can never tell when or where a frequency occurs.
- Any abrupt changes in time in the input signal $f(t)$ are spread out over the whole frequency axis in the transform output $F(\omega)$ and vice versa.

- The wavelet uses a short window at high frequencies and a long window at low frequencies. It can localize abrupt changes in both time and frequency domains.
- DCT is signature length independent while the wavelet is a decomposition of a 2^n length signature. Hence, the DCT can be used to compare different length signatures.

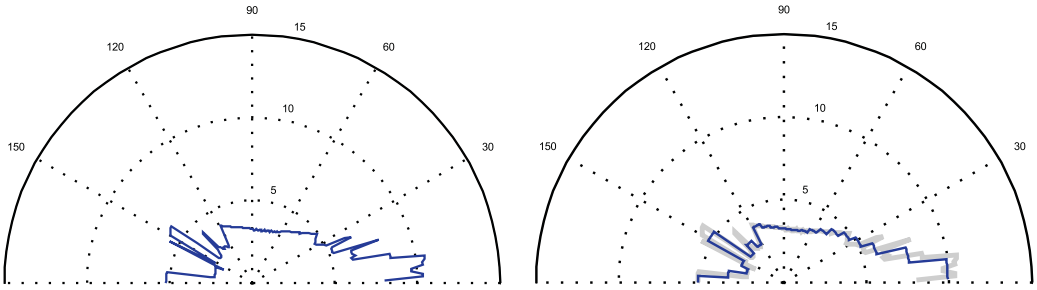
From the preceding discussion we conclude that wavelets is superior to the DCT in compression because they maintain both frequencies and locations while the DCT is superior in matching due to its approximation capabilities.

Haar Wavelet Transform HWT

Among many wavelet algorithms, including Daubechies wavelets (DWT), Mexican Hat wavelets and Morlet wavelets, the Haar Wavelets are especially popular due to their simplicity and limited support. The HWT enable applying our approach in online mode on onboard computers of mobile robots. We want to have a decomposition that is fast to compute and requires little storage for each sequence. The Haar wavelet is chosen for the following reasons: **1)** It allows good approximation with a subset of coefficients, **2)** It can be computed quickly and easily, requiring linear time in the length of the sequence and simple coding, **3)** It preserves Euclidean distance.

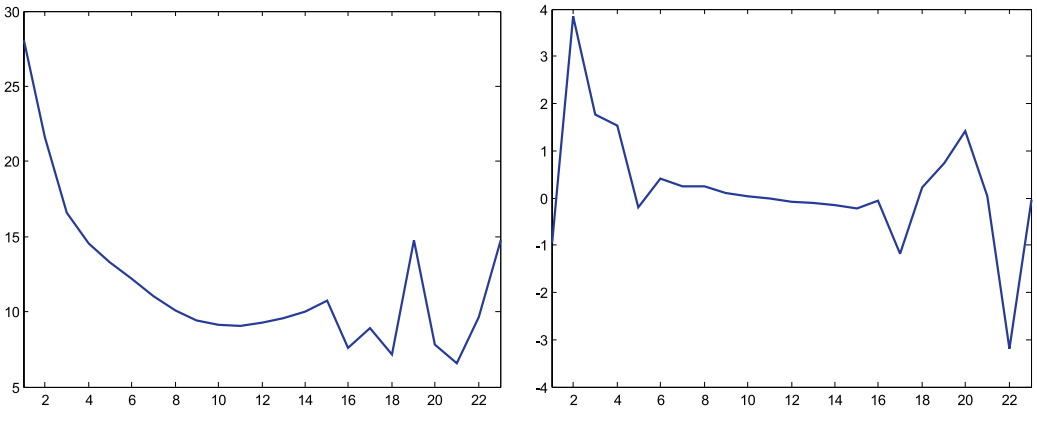
The most interesting dissimilarity between the DCT and the HWT is that individual wavelet functions are localized in space, while DCT functions are not. This localization feature, along with wavelets' localization of frequency, makes many functions and operators using wavelets sparse when transformed into the wavelet domain. This sparseness, in turn, results in a number of useful applications such as data compression, detecting features in patterns, and removing noise from data series. One way to see the spatial-spectral resolution differences between the DCT and the HWT transforms is to look at the basis function coverage of the spatial-spectral domain. The Haar wavelet uses a rectangular window to sample the data series. The first pass over the time series uses a window width of two. The window width is doubled at each step until the window encompasses the entire data series.

In order to study the laser signal using the HWT, we expose the signature to a recursive (multi-resolution) transform. Each time, we extract a set of coefficients, which deduce the data variation found in the signal at a given sub-band. Figure (3.16.a) and figure (3.16.b) show a laser sample and the corresponding complete signature taken from 8 sub-bands. The signal is entailed with the approximate coefficients found with the highest sub-band. In order to perform compression



(a) Orignal laser signature (b) Original and restored signatures

Figure 3.16: Haar wavelet transform (HWT) of laser signature



(a) Wavelet approximations (b) Wavelet coefficients

Figure 3.17: laser signature wavelet components

in the wavelet domain, a given number of the lower sub-bands is being neglected when defining our signature. The HWT calculates a set of wavelet coefficients and a set of averages. If the data series $d_0, d_1, \ldots d_{n-1}$ contains n elements, there will be $n/2$ averages and $n/2$ coefficients. Averages are stored in the lower half of the n element array and the coefficients are stored in the upper half. The averages become the input for the next step in the wavelet calculation. The recursive iterations continue until a single average and a single coefficient are reached. This replaces the original data set of n elements with an average, followed by a series of coefficients. The equations used to calculate HWT averages are $a_i = (d_{2i} + d_{2i+1})/2$ and coefficients $c_i = (d_{2i} - d_{2i+1})/2$.

In wavelet terminology, the average is calculated by the ***scaling function***, while the coefficient is calculated by the ***wavelet function***. The inverse can be simply reconstructed using $d_{2i} = a_i + c_i$ and $d_{2i+1} = a_i - c_i$, where $i = 0, 1, 2, \ldots (n/2 - 1)$. The first scaling vector (average) can be calculated in the linear algebra view of the HWT by the product of the signature data $[d_0, d_1, \ldots d_{n-1}]$ and the vector, of the same size, $[0.5, 0.5, 0, 0, \ldots, 0]$. The first coefficient is calculated by the product of the signature and the vector $[0.5, -0.5, 0, 0, \ldots, 0]$, this is the wavelet vector. The next average and coefficient are calculated by rotating the scaling and wavelet vectors by two and calculating the products. In the wavelet literature scaling and wavelet values are sometimes represented by h_i and g_i respectively. In the case of the HWT, scaling function coefficients are $h_0 = 0.5$ and $h_1 = 0.5$, while wavelet function coefficients are $g_0 = 0.5$ and $g_1 = -0.5$. The scaling and wavelet matrix of HWT can be constructed as follows:

$$\begin{bmatrix} a_0 \\ a_1 \\ . \\ a_{n/2-1} \\ c_0 \\ c_1 \\ . \\ c_{n/2-1} \end{bmatrix} \Leftarrow \begin{bmatrix} a_0 \\ c_0 \\ a_1 \\ c_1 \\ . \\ . \\ a_{n/2-1} \\ c_{n/2-1} \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & 0 & 0 & . & . & 0 & 0 \\ g_0 & g_1 & 0 & 0 & . & . & 0 & 0 \\ 0 & 0 & h_0 & h_1 & . & . & 0 & 0 \\ 0 & 0 & g_0 & g_1 & . & . & 0 & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & . & h_0 & h_1 \\ 0 & 0 & 0 & 0 & . & . & g_0 & g_1 \end{bmatrix} \bullet \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ . \\ . \\ . \\ d_{n-1} \end{bmatrix} \tag{3.16}$$

The arrow represents a split operation that reorders the result so that the average values are in the first half of the vector and the coefficients are in the second half. To complete the HWT the above transform is repeated to reach the final average and the final coefficient.

Signature Restoration

Like the forward HWT, a step in the inverse HWT can be described in linear algebra terms. The matrix operation to reverse the first step of the HWT for a signature is as follows:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ . \\ . \\ . \\ d_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & . & . & 0 & 0 \\ 1 & -1 & 0 & 0 & . & . & 0 & 0 \\ 0 & 0 & 1 & 1 & . & . & 0 & 0 \\ 0 & 0 & 1 & -1 & . & . & 0 & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & . & 1 & 1 \\ 0 & 0 & 0 & 0 & . & . & 1 & -1 \end{bmatrix} \bullet \begin{bmatrix} a_0 \\ c_0 \\ a_1 \\ c_1 \\ . \\ . \\ a_{n/2-1} \\ c_{n/2-1} \end{bmatrix} \Leftarrow \begin{bmatrix} a_0 \\ a_1 \\ . \\ a_{n/2-1} \\ c_0 \\ c_1 \\ . \\ c_{n/2-1} \end{bmatrix} \quad (3.17)$$

In this case the arrow represents a merge operation that interleaves the averages and the coefficients.

HWT vs. DWT

The area of wavelet literature covers a wide variety of wavelet algorithms, which are drawn from an infinite set of wavelet algorithms. The first and most important question is "which algorithm should I use?". Unfortunately, this question has no definite answer, we can simply say that the choice of the wavelet algorithm depends on the application, the computation power, the required precision etc.. Generally, the wavelet transform generates a "down sampled" smoothed structure of the signature (calculated by the wavelet scaling function) and a "down sampled" structure of the signature that includes variations among signal elements. In some references, the ***scaling function*** (smoothing) is sometimes referred to as a ***low pass filter*** and the ***wavelet function*** is sometimes referred to as a ***high pass filter***.

In a comparison of the HWT with the DWT, we notice that the HWT has no overlap between successive pairs of scaling and wavelet functions, as there is with the DWT. The HWT high pass filter generates a result that comprises the difference between even and odd elements, while the difference between odd and even elements are not comprised in the coefficient band, calculated by a single step of HWT high pass filter (although this change will be picked up by later steps). In contrast, there is overlap between successive DWT high pass filters, so change between any two elements will be comprised in the result.

The main differences between HWT and DWT are **1)** The DWT wavelet transform is more precise, since a change in the input data set is included in the high pass filter results at each transform step, **2)** The cost of using the DWT algorithm is higher computation overhead, twice the number of operations, compared with the HWT, **3)** The DWT is more complicated, the algorithm must properly consider the edge condition when $i = 0$.

3.3.5 Notes on the Applied Matching Techniques

To clarify the capabilities of data series matching algorithms; ED, CCA, DCT and HWT, some important points must be considered such as the compression capability, preserving frequency and euclidean distance, probability of local minima, and transition between levels. Table (3.1) introduces a brief illusteration of these algorithms.

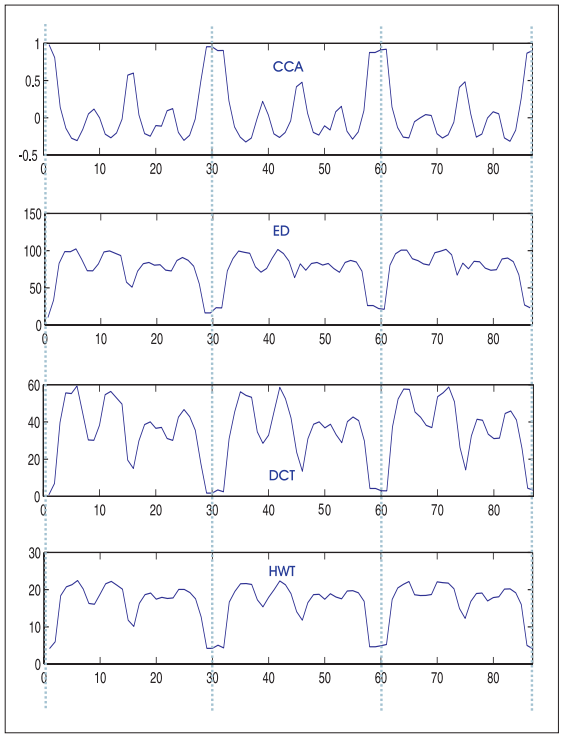


Figure 3.18: Comparison signature matching techniques

To understand the presented approaches deeply, we have performed some experiments using a Sick LMS 200 laser scanner mounted on our B21-RWI robot platform (Colin) in the laboratory for autonomous mobile robots. The laser signature of an arbitrary landmark has been registered and laser scan samples from four nearby locations have been stored. The stored samples comprise both translation and rotation. Then we apply HWT and DCT compression on the scans using equal compression factor. Finally we use the selected landmark and search for it among all samples. To judge the results fairly we use mean square error in order to match the spatial, DCT, and HWT signatures with the corresponding ones. Figures (3.18) shows the experimental results. It is worth mentioning that we use a compression factor of a high rate for DCT and HWT without sacrificing the nature of the signature.

These experiments show that neither DCT nor HWT are superior to each other as their signatures are still capable of embracing the information even with high compression rates. Figure (3.18) clarifies that both DCT and spatial techniques comprise a unique global reference pose while they incorporate multiple local minima. Therefore, both DCT and HWT are considered partially invariant to translation and rotation.

	Spatial(ED-CCA)	**Spectral(DCT)**	**Wavelet(HWT)**
Compression	no	yes	yes
Frequency	no	yes	yes
Spatial	yes	no	yes
Transition	sharp	smooth	smooth
Local minima	+	++	++
Data length	dependent	independent	dependent
Model	no	no	no
Learning	no	batch	batch
Convergence	abrupt	gradual	gradual

Table 3.1: Comparison of signature date matching algorithms

3.4 Laser Sonar Integration

The integration of both laser and sonar sensors has a meaningful role in mapping and navigation. It leads to generating high consistent maps that cover the area detected by both of them. The integration of open angle sonar and the 2D planner laser is capable of ensuring robot's safety as it navigates in unknown terrains with randomly distributed obstacles. This spatial fusion is essential in some environments whereas the laser penetrates through glass doors, it gets no reflection back from mat black or specular objects, or it is unable to detect the obstacles because of its 2D planner scanning. In that case, the open angle sonar is considered as an advantage.

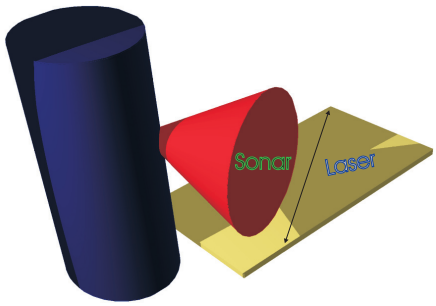


Figure 3.19: Spatial laser sonar itegration

The disadvantages of sonar can be eliminated by using a simple fuzzy membership function that returns the minimum range value from multi ranges acquired by sonar and laser sensors. This membership function excludes the error emanating from multi reflections of sonar waves, 2D limitations of laser scanning (tables) or from penetration of laser ray through transparent materials. The formulation of the AND fuzzy rule is as follows:

$$d_r = \mu_{s \cap l} = \min(s_1, s_2, \ldots, s_{n_s}, l_1, l_2, \ldots, l_{n_l}) \tag{3.18}$$

where d is the distance to the nearest obstacle in the front free space, s_i is the range acquired from front panel sonar sensor no. i, and l_i is the length of laser ray no. i of a registered scan. This function returns the minimal displacement to the nearest obstacle in the front free space and that achieves more safety. More details about this subject are found in chapter (6), page 85.

Chapter 4

Compasses

In this chapter we concentrate on employing a digital compass in mapping and localization of indoor terrains. In mapping, we use the geomagnetic field distribution along with the traditional visible odometric maps acquired by laser range finders and sonar ranging modules to model the environment. The most important idea of this approach is that we allow the robot to use non-biomemtic sensors, or at least sensors that humans don't normally possess. The second benefit of using compasses is the possibility of detecting geomagnetic signature peaks that remark the poses of the central nodes of the VMP discussed in chapter (3), page (23). In contrary to the dominant idea of considering the disturbances of compass measurements as white noise [152, 139], we could clearly detect and analyze a geomagnetic signature of the VMP vector [192, 7].[1]

We used 2 compasses: a KVH-C100 fluxgate compass and a vector compass mounted on RWI-B21 and Pioneer Active Media robot platforms, see figure (4.3).

4.1 Natural Magnetic Field

The earth acts like a great spherical magnet, in that it is surrounded by a magnetic field. In general, The earth's magnetic field resembles the field generated by a dipole magnet (i.e., a straight magnet with a north and south pole) located at the center of the earth. The axis of the dipole is offset from the axis of the earth's rotation by $\approx 11^{\circ}$. This means that the north and south geographic poles and the north and south magnetic poles are not located in the same place. At any point, the earth's magnetic field is characterized by a direction and intensity which can be measured.

[1]I would like to acknowledge the technical support of P. Heinemann in the programming of the Pioneer robot platform, Active Media, to record the presented compass measurements.

4.1.1 Definition of Magnetic Elements

To measure the earth's magnetism in any place, we must measure the direction and intensity of the field. The earth's magnetic field is described by seven parameters. These are: **1)** declination (D), **2)** inclination (I), **3)** horizontal intensity (H), **4)** vertical intensity (Z), **5)** total intensity (F), **6)** north (X), **7)** east (Y). Where, X and Y are the components of the horizontal intensity.

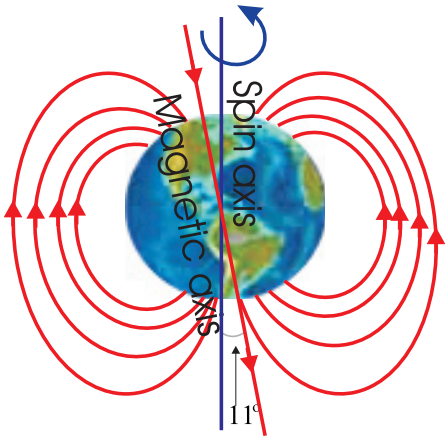


Figure 4.1: Natural magnetic field

The parameters describing the direction of the magnetic field are declination (D) and inclination (I). D and I are measured in units of degrees, positive east for D and positive down for I. In older literature, the term *magnetic elements* often referred to D, I, and H, see figure (4.2).

Declination is the angle between north and the horizontal projection of (F). In other words, magnetic declination is the angle between magnetic north and true north. This value is measured positive through east and varies from 0° to 360°. D is considered positive when the angle measured is east of true north and negative when west.

Inclination Magnetic inclination is the angle between the horizontal plane and the total field vector, measured positive into earth. In other words, the angle between the surface of the earth and (F). Positive inclinations indicate (F) is pointed downward, negative inclinations indicate (F) is pointed upward. Inclination varies from −90° to 90°.

Magnetic Equator The location around the surface of the earth where the earth's magnetic field has an inclination of zero (the magnetic field vector F is horizontal). This location does not correspond to the earth's rotational equator.

Magnetic Poles The locations on the surface of the earth where the earth's magnetic field has an inclination of either plus or minus 90° (the magnetic field vector F is vertical). These locations do not correspond to the earth's north and south poles.

4.1.2 Magnetic Units

The intensity of the total field (F) is described by the horizontal component (H), vertical component (Z), and the north (X) and east (Y) components of the horizontal intensity. These components may be measured in units of Oersted (1 Oersted= Gauss), but are generally reported in nanoTesla (1 nT * 100,000=1 Gauss). The earth's magnetic field intensity is roughly between 25,000 - 65,000 nT (0.25 - 0.65 Gauss). The geomagnetic field measured at any point on the earth's surface is a combination of several magnetic fields generated by various sources. These fields are superimposed on and interact with each other. This portion of the geomagnetic field is often referred to as the main field. The main field varies slowly in time and can be described by mathematical models such as the International Geomagnetic Reference Field (IGRF) and World Magnetic Model (WMM). The main field creates a cavity in interplanetary space called the magnetosphere, where the earth's magnetic field dominates in the magnetic field of the solar wind. The magnetosphere is shaped somewhat like a comet in response to the dynamic pressure of the solar wind. It is compressed on the side facing the sun to about 10 earth radii and is extended tail-like on the side away from the sun to more than 100 earth radii. The magnetosphere deflects the flow of most solar wind particles around the earth, while the geomagnetic field lines guide charged particle motion within the magnetosphere. The differential flow of ions and electrons inside the magnetosphere and in the ionosphere form current systems, which cause variations in the intensity of the earth's magnetic field. These external currents in the ionized upper atmosphere and magnetosphere vary on a much shorter time scale than the internal main field and may create magnetic fields as large as 10% of the main field. It is the main field component that is modeled by the IGRF and WMM. Other important sources are the fields arising from electrical currents flowing in the ionized upper atmosphere, and the fields induced by currents flowing within the earth's crust.

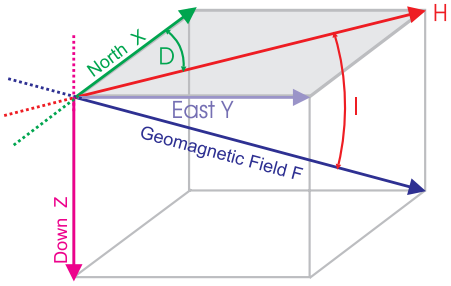


Figure 4.2: Magnetic field elements

4.1.3 Field Variations

The magnetic field is different in different places. In fact, the magnetic field changes with both location and time. It is so irregular that it must be measured in various places to get a satisfactory picture of its distribution. This is done at the approximately 200 operating magnetic observatories worldwide and at several

more temporary sites. However, there are some regular features of the magnetic field. At the magnetic poles, a dip needle stands vertical (dip=90°), the horizontal intensity is zero, and a compass does not show direction (D is undefined). At the north magnetic pole, the north end of the dip needle is down; at the south magnetic pole, the north end is up. At the magnetic equator the dip or inclination is zero. Unlike the earth's geographic equator, the magnetic equator is not fixed, but slowly changes. The disturbances of the magnetic compasses could be natural or artificial.

Natural Sources: emanate from environmental and cosmic phenomena. The earth's magnetic field is actually a composite of several magnetic fields generated by a variety of sources. These fields are superimposed on each other and through inductive processes interact with each other. The most important of these geomagnetic sources are:

- The earth's conducting, fluid outer core $\approx 90\%$
- Magnetized rocks in the earth's crust
- Fields generated outside earth by electric currents flowing in the ionosphere and magnetosphere
- Electric currents flowing in the earth's crust (usually induced by varying external magnetic fields)
- Ocean current effects

These contributions all vary with time on scales ranging from milliseconds (micro pulsations) to millions of years (magnetic reversals). More than 90% of the geomagnetic field is generated by the earth's outer core. It is this portion of the geomagnetic field that is represented by the Magnetic Field Models.

Artificial Sources: are man-made sources affecting the magnetic field locally such as:

- Large ferro-magnetic structures
- Internal electrical components and robot motors
- Mechanical vibration, see figure (4.5), which can be reduced using mechanical dampers and low pass filters.
- Power lines
- Surrounding electrical instruments

4.1.4 Correction of Compass Bearing

One can compute the true bearing from a magnetic bearing by adding the magnetic declination to the magnetic bearing. This works so long as you follow the idea that degrees west are negative. (i.e. a magnetic declination of 10° west is −10° and bearing of 45° west is −45°).

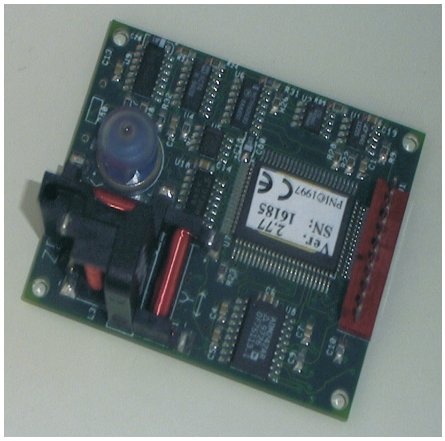
4.2 Robot Compass

Vehicle heading is the most significant of the navigation parameters (x, y, and θ) in terms of its influence on accumulated dead-reckoning errors. For this reason, sensors which provide a measure of absolute heading are extremely important in solving the navigation needs of autonomous robots, the magnetic compass is such a sensor. One disadvantage of any magnetic compass is that the earth's magnetic field is often distorted near power lines and steel structures. This makes the straightforward use of geomagnetic sensors difficult for indoor use. The 2 compasses used in this study are shown in figure (4.3). Based on a variety of physical effects related to the earth's magnetic field, different geomagnetic sensors are available. These include: **1)** Fluxgate compasses, **2)** Magnetoresistive compasses, **3)** Hall-effect compasses, **4)** Mechanical magnetic compasses, **5)** Magnetoelastic compasses, **6)** Magnetoinductive compasses.

The fluxgate compass is best suited for the use of mobile robot applications. When maintained in a level attitude, the fluxgate compass will measure the horizontal component of the earth's magnetic field with decided advantages of low power consumption, no moving parts, tolerance to shock and vibration, rapid start-up, and relatively low cost. If the vehicle is expected to operate over uneven terrain, the sensor coil should be gimbal-mounted and mechanically dampened to prevent serious errors introduced by the vertical component of the geomagnetic field [152].

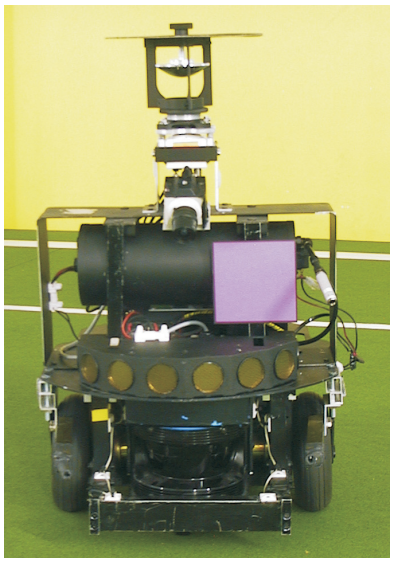
4.2.1 Related Work

In 1832 Gauss and Weber began investigating the theory of terrestrial magnetism after Alexander von Humboldt attempted to obtain Gauss's assistance in making a grid of magnetic observation points around the earth. Gauss was excited by this prospect and by 1840 he had written three important papers on the subject: *Intensitas vis magneticae terrestris ad mensuram absolutam revocata* (1832), *Allgemeine Theorie des Erdmagnetismus* (1839) and *Allgemeine Lehrsätze in Beziehung auf die im verkehrten Verhältnisse des Quadrats der Entfernung wirkenden Anziehungs-*



(a) Vector Magnetoinductive Compass

(b) KVH C-100 fluxgate compass



(c) Pioneer, Active Media

(d) B21, RWI

Figure 4.3: Digital compasses and the corresponding robot platforms used throughout our research work

und Abstossungskräfte (1840). These papers all dealt with the current theories on terrestrial magnetism, including Poisson's ideas, absolute measure for magnetic force and an empirical definition of terrestrial magnetism.

Related to the research work on using compasses in robotics, many researchers adopted the use of compass as a heading device for autonomous mobile robots e.g.:

A. Schmolke et. al. studied the performance of the path integration without external reference compared with the performance using a polarization compass. The experiments were carried out using a Khepera miniature robot. A polarization compass has been tested for indoor circumstances. Thereby, the robot determines the E-vector orientation, i.e. the direction of the electric field oscillation of the polarized light with three sensors which is the minimal number of sensors needed for this task. An artificial polarization source has been used to substitute the blue sky [172].

S. Suksakulchai et. al. proposed a method for mobile robot localization using a digital magnetic compass. They took advantage of the magnetic field disturbances as distinctive place recognition signatures. They employed a sequential least-squares approximation for matching the electromagnetic signature of a corridor [192].

L. Ojeda and J. Borenstein published a paper on *"Experimental results with the KVH C-100 fluxgate compass in mobile robots"*. This paper presents a discussion on the use of electronic compasses in mobile robots. In that way, different error sources are considered, and solutions are proposed to correct these errors. Their experimental results show the effectiveness of some of the error reduction measures. These results also show what performance can be expected from a well-calibrated compass system. The overall most important result is that errors due to external magnetic interferences are the most severe and hard-to-correct ones [152].

L. S. Lopes et. al. published a paper entitled *"Intelligent Control and Decision-Making demonstrated on a Simple Compass-Guided Robot"*. This paper presents the architecture and algorithms developed for Dom Dinis, a simple compass-guided robot built by the authors. This includes environment exploration, task planning and task execution. Environment exploration, based on repeating a reactive goal search, enables a progressive construction of a grid-based map. Based on the (possibly incomplete) map, the robot is able to plan its tasks. The execution capabilities of the robot include exception handling. Essential to all these

capabilities is the knowledge of the robot's position in the world. The position is computed based on tracking traversed distances and followed orientations. Orientation is given by a compass. Dom Dinis did not use wheel encoders at all [119].

Several animals use the bio-compass in their navigation and orientation such as; loggerhead sea turtles, Spiny lobsters, pigeons, blind mice and humpback whales, see figure (4.4) [118, 166, 212].

4.2.2 Fluxgate Compass

The fluxgate magnetometer was originally designed and developed during World War II. It was used as a submarine detection device for low-flying aircrafts. Today it is used for conducting magnetic surveys from aircraft and for making borehole measurements. The fluxgate magnetometer is based on what is referred to as the magnetic saturation circuit. One model of them has two parallel bars of a ferromagnetic material which are placed closely together. The susceptibility of the two bars is large enough so that even the earth's relatively weak magnetic field can produce magnetic saturation in the bars. Magnetic saturation refers to the induced magnetic field produced in the bars. In general, as the magnitude of the inducing field increases, the magnitude of the induced field increases in the same proportion relating the external to the induced magnetic fields. For large external field strengths, however, this simple relationship between the inducing and the induced field no longer holds. Saturation occurs when increases in the strength of the inducing field no longer produce larger induced fields. Each bar is wound with a primary coil, but the direction in which the coil is wrapped around the bars is reversed. An alternating current (AC) is passed through the primary coils causing a large, inducing magnetic field that produces induced magnetic fields in the two cores that have the same strengths but opposite orientations. A secondary coil surrounds the two ferromagnetic cores and the primary coil. The magnetic fields induced in the cores by the primary coil produce a voltage potential in the secondary coil. In the absence of an external field (i.e., if the earth had no magnetic field), the voltage detected in the secondary coil would be zero because the magnetic fields generated in the two cores have the same strength but are in opposite directions (their effects on the secondary coil exactly cancel). If the cores are aligned parallel to a component of a weak, external magnetic field, one core will produce a magnetic field in the same direction as the external field and reinforce it. The other will be in opposition to the field and produce an induced field that is smaller. This difference is sufficient to induce a measurable voltage in the secondary coil that is proportional to the strength of the magnetic field in the direction of the cores. Thus, the fluxgate magnetometer is capable of measuring the strength of any component of the earth's magnetic field by simply re-orienting

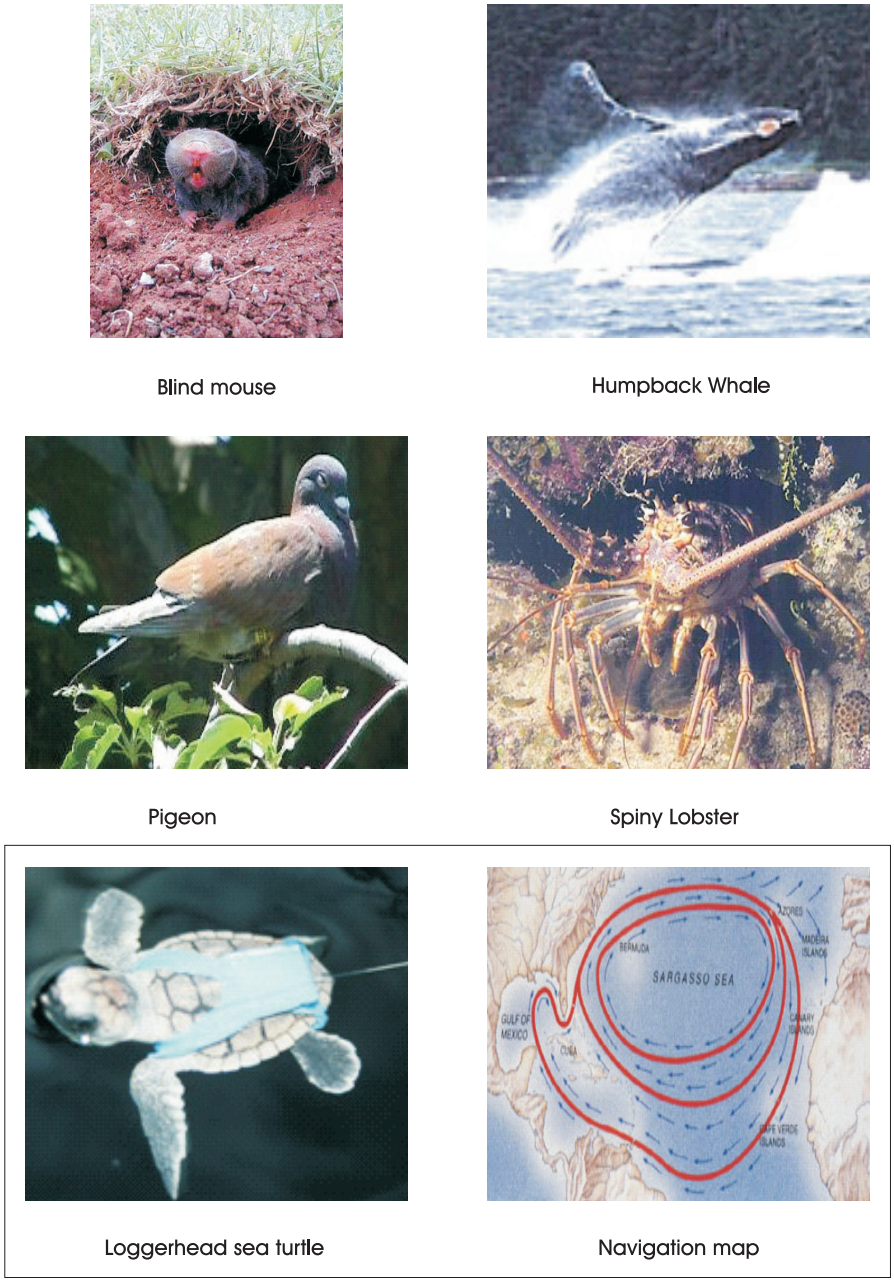
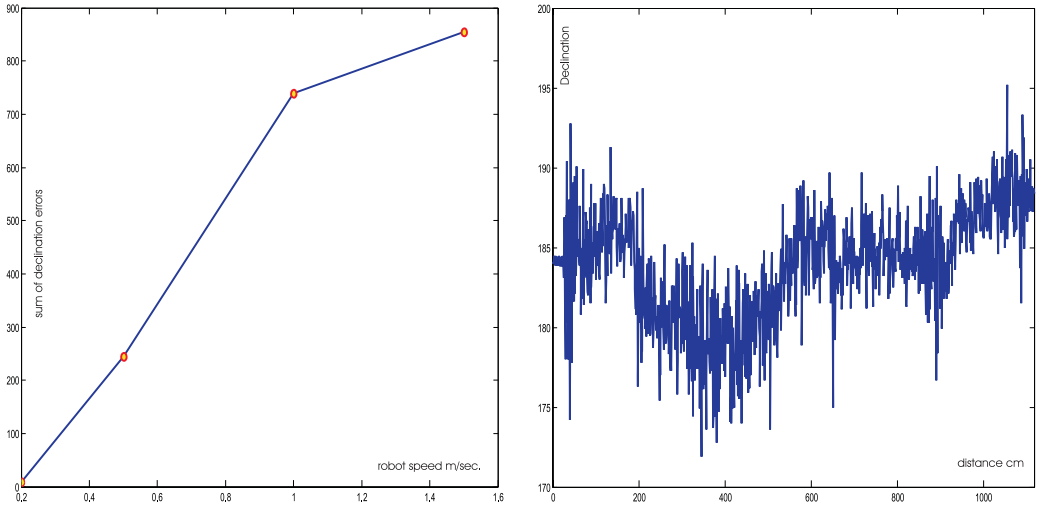


Figure 4.4: Animals using Bio-Compass

the instrument so that the cores are parallel to the desired component. Fluxgate magnetometers are capable of measuring the strength of the magnetic field to about 0.5 to 1.0 nT. Unlike the commonly used gravimeters, fluxgate magnetometers show no appreciable instrument drift with time.



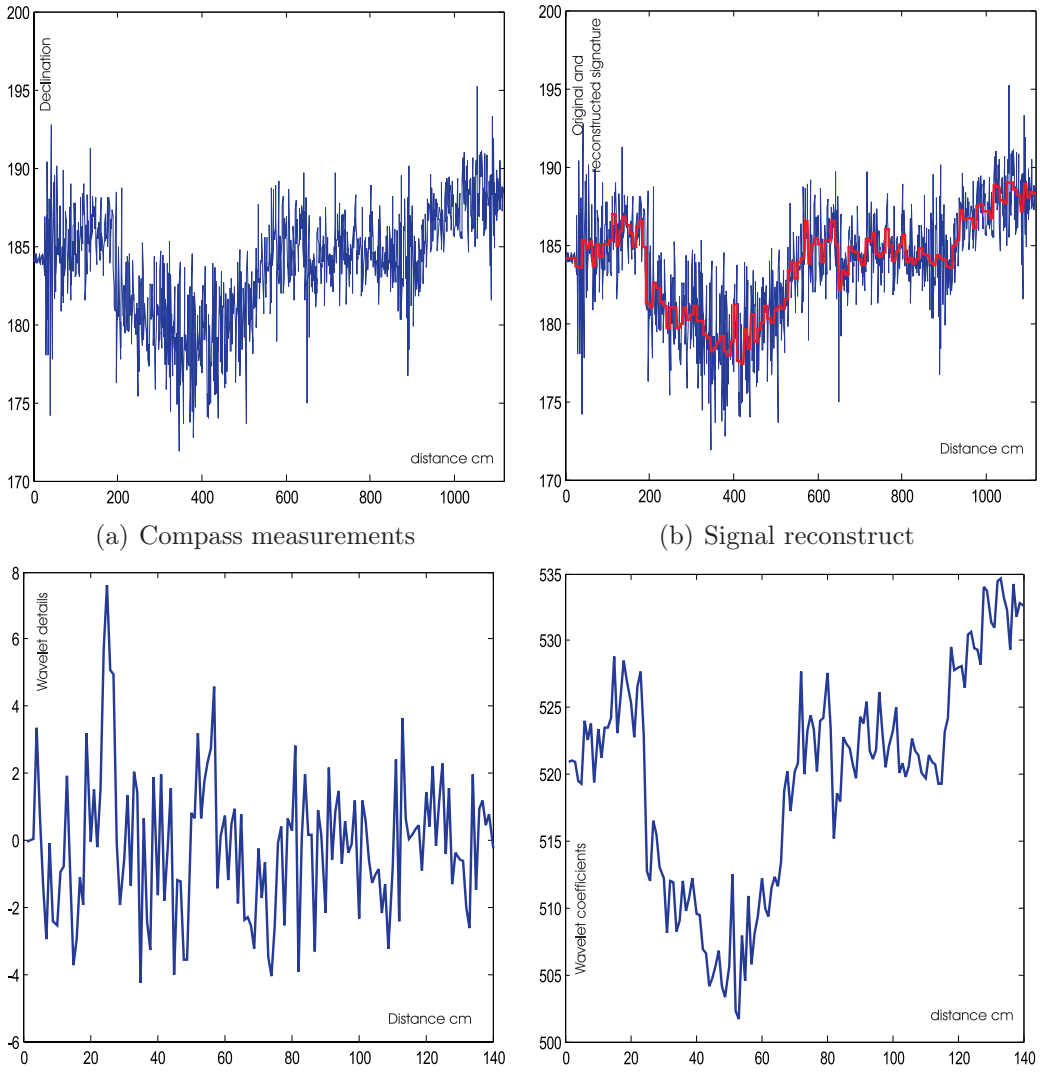
(a) Compass disturbance w.r.t. speed (b) Compass measurements

Figure 4.5: Compass disturbances and measurements under mechanical vibration

To reduce the high frequency (HF) noise we employed the wavelet decomposition to extract the original compass measurements, see figure (4.6).

4.2.3 Magnetic Shielding

Magnetic fields can be reduced using high-permeability shielding alloys called μ-metal. These materials are able to divert the magnetic flux to themselves, so the magnetic fields around them can be reduced significantly. The deviation produced by the host platform (motors, wires, etc.) can be reduced with the use of magnetic shielding made of mu-metal. The magnetic shielding have to provide a complete path for the magnetic field lines, in order to prevent the magnetic field from causing interference outside of the shielding. Closed shapes like cylinders with caps around the motors, boxes with covers on power supplies, tubes around wires, etc. are the most effective. In addition, some physical separation between the mu-metal housing and the compass is still necessary to avoid magnetic interference [27, 3, 152]. To keep the compass away from internal fields we mounted it above the robot.



(a) Compass measurements

(b) Signal reconstruct

(c) Wavelet details at decomposition level 3

(d) Wavelet coefficients at decomposition level 3

Figure 4.6: High frequency noise reduction using wavelet decomposition

4.3 Geomagnetic Localization

Odometry is the most widely used method for determining the position of a mobile robot. Odometry is the only dead-reckoning system available on several mobile robots. Unfortunately, the acquisition of the odometric data is associated with different types of noise e.g. non-linear noise (backlash, hysteresis, toggle, threshold, saturation, damping and friction), time based noise (accumulation effect) and white noise. Another form of odometric errors is generated by the mechanical system from rotation or due to slippage and drifting [6]. Therefore, the odometry needs to be reset from time to time. To overcome this problem, robust and accurate localization of the robot is needed. There are many researchers, who have adopted using vision for natural landmark recognition. Those techniques often require high-performance onboard computers and expensive cameras.

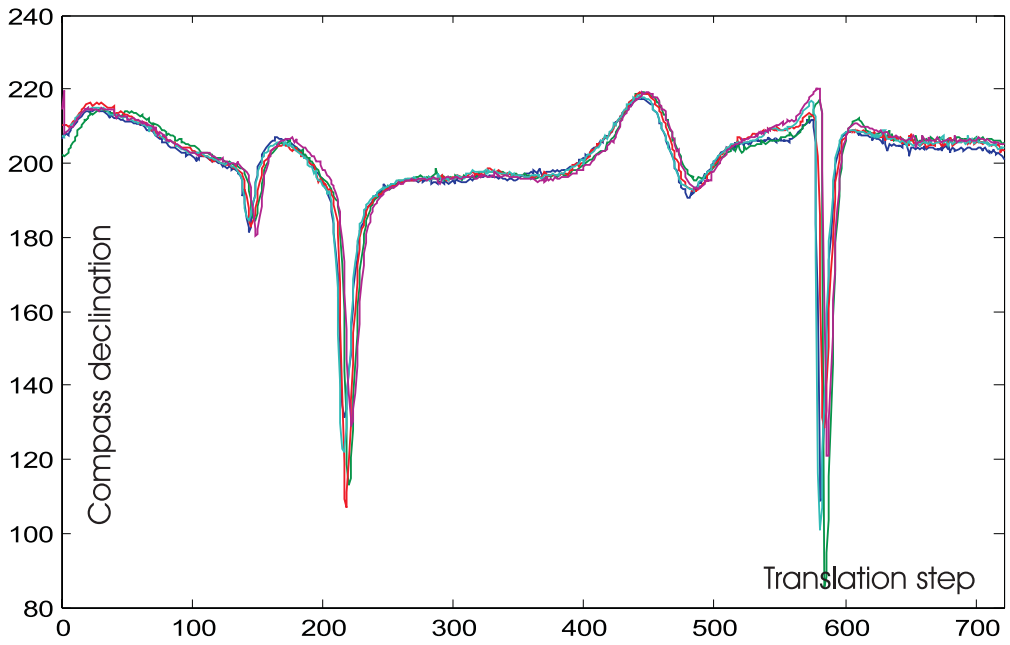
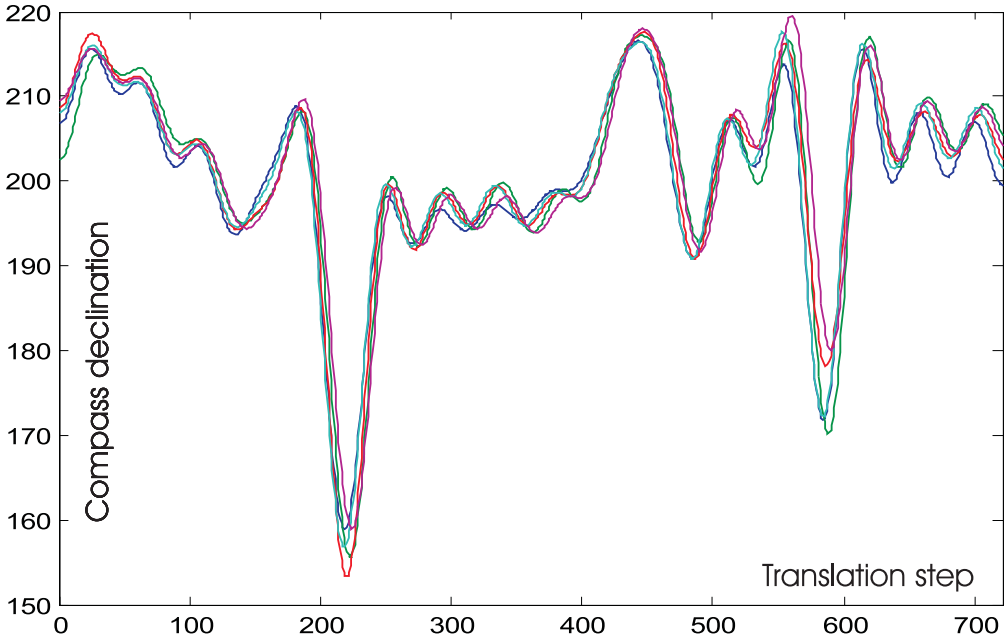
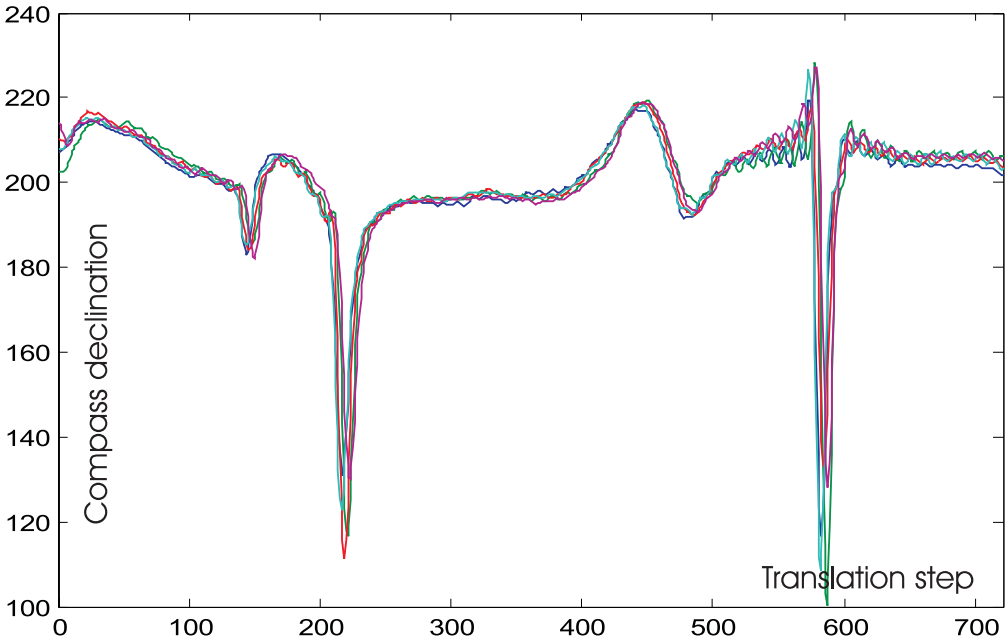


Figure 4.7: The original signature constructed in repeated 5 trials

Electronic compasses are often used to detect headings of mobile robots in outdoor terrains. However, electronic compasses have one drawback when used inside a building: they can easily be disturbed by electromagnetic sources (e.g. power lines, electromagnetic fields of robot internal components, fields of external instruments or large Ferro-magnetic structures) [152, 7]. This makes it impossible to use electronic compasses as reliable heading devices for indoor applications. In-



(a) Signature reconstruction by using 5% of the DCT coefficients



(b) Signature reconstruct by using 20% of the DCT coefficients

Figure 4.8: Signature reconstruction by using the DCT

spiration for this investigation arose to investigate error readings of the compass caused by electromagnetic sources or large ferromagnetic structures.

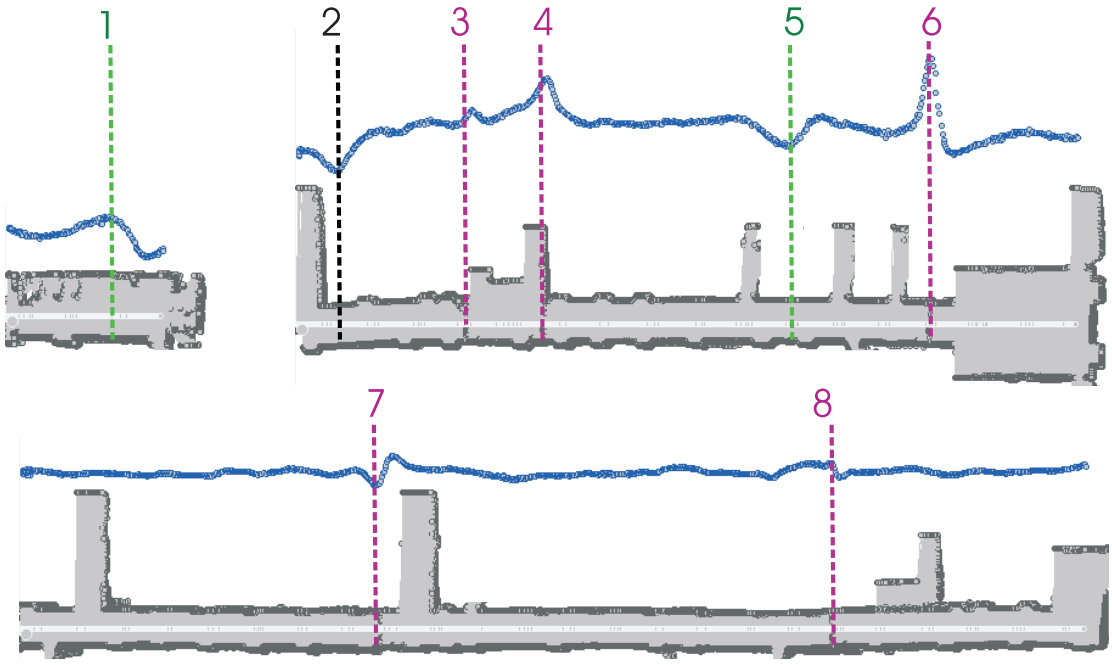


Figure 4.9: Combination of Laser and magnetic mapping of the corridors at our Institute WSI, Sand 13 , Tübingen

In figure (4.9) we define some special marks (denoted from 1 to 8). The disturbances of the geomagnetic measurements at positions 3,4,6,7 and 8 are due to presence of ferro-magnetic structures (glass door frames), disturbances at positions 1, 5 are due to the electromagnetic field of wireless Ethernet-LAN transceivers while the disturbance at position 2 has no obvious cause.

While the geomagnetic signature alone is not as good as signatures obtained by a laser scanner and in our experiments could not be used for navigation alone by itself it may be used as an additional sensor to remove ambiguities and to help in global localization together with another sensor like sonar or laser scanner.

With its capabilities to detect steel frames WLAN transceivers which may be hidden behind wall or ceiling, a digital compass may be used to resolve ambiguities in probabilistic global navigation and map building or the mapping approach like the VMP.

The advantage of the magnetic field disturbances is taken by using them as recognition signatures to localize distinctive central nodes. The spectrum analysis and reconstruction of the geomagnetic signature, presented in this study, rely mainly upon the frequency domain convolution of the discrete cosine transform (DCT). The DCT belongs to algorithms (e.g. DFT, FFT, DST, wavelet, Gabor, etc.) used for the frequency analysis of signals and patterns. Moreover, the DCT is used not only for coding but also for signal and pattern compression. Figure (4.7) shows typical geomagnetic signatures in repeated trials, 5 times, in a ≈ 60 m corridor in our work space and the corresponding reconstructed signatures are shown in figures (4.8)

Chapter 5

Motion Planning for Non-Holonomic Robots

5.1 Introduction

The robot path planning problem, which asks for the computation of collision free paths in environments containing obstacles, has received a great deal of attention in the last decades [218, 41, 86, 6]. In the basic problem, there is one robot present in a static and known environment, and the task is to compute a collision-free path describing a motion that brings the robot from its current position to some desired goal position. Variations and extensions of this basic problem statement are numerous.

Motion planning algorithms support a wide variety of scientific applications in different fields, like

Astronomy : Several algorithms have been developed originally to study cosmic phenomena. Moreover, these algorithms are important to plan manned and unmanned space missions as well as to design the trajectories of space rovers [181, 180, 112].

Automation of traffic systems: The main goal of that is to manage the global traffic system and to plan the motion of individual vehicles autonomously.

Simulators and animators: demand efficient and robust path planning techniques to move the virtual dynamic objects in PC games, digital actors and virtual reality programs.

Electronics: rely on path planning algorithms to develop short connections autonomously by which the energy consumption and the mutual capacitance

among connections will be decreased. Therefore, VLSI-design and PCB-layout are successful applications for such algorithms.

Robotics: is an important area for applying path planning algorithms to different types of platforms (e.g. indoor, outdoor, micro, mini, underwater, aerospace robots, manipulators etc.).

Non-holonomic constraints add an extra level of difficulty to the path planning problem. The desired path must **(1)** be collision free, **(2)** describe motions that are executable for the robot, **(3)** be as short as possible, **(4)** be computed in a limited time and **(5)** be generated autonomously without any human intervention. We refer to such paths as desired paths.

To start with, for non-holonomic robots, computation of collision-free paths is not sufficient. Not only are the admissible robot placements constrained by obstacles and the robot geometry, but also are the directions of motion subject to constraints. For example, our mobile robots (RWI-B21) moving on wheels have such non-holonomic constraints, due to the fact that their wheels are not allowed to slide. Another realistic scenario is that of selecting an appropriate solution among one of the available possibilities. In this case, apart from the restrictions imposed by the obstacles, robot geometry, and possible non-holonomic constraints, one also has to avoid collisions between the robot and dynamic objects mutually. Moving obstacles, uncertainties in sensing, and inexact control add further levels of difficulty. In order to build robots that can autonomously act in real-life environments, path planning problems need to be solved. However, it has been proven that, in general, solving even the basic path planning problem requires time which is proportional to the number of possible collision to be avoided. In spite of this discouraging problem complexity, various such complete planners have been proposed. Their high complexity however makes them impractical for most applications. And every extension of the basic path planning problem adds to the computational complexity. Therefore, if we allow for moving obstacles, the problem becomes exponential in their number. Assuming uncertainties in the robots sensing and control leads to an exponential dependency on the complexity of the obstacles. The above bounds deal with the exact problem, and therefore apply to complete planners. These are planners that solve any solvable problem, and return failure for each non-solvable one. So for most practical problems it seems impossible to use such complete planners. This has lead many researchers to consider simplifications of the problem statement. A quite recent direction of research, which we just want to mention briefly here, deals with the formulation of assumptions on the robot environment that reduce the path planning complexity. This is based on the belief that there exists a substantial gap between the theoretical worst-case bounds of path planning algorithms and their practical complexity. A

number of researchers have attempted to formulate assumptions on the obstacles that prohibit the (artificial) constructions that cause the worst-case bounds. However, this line of research has been mainly of theoretical nature, and has not yet resulted in implementations of practical path planners. Moreover, it is currently not clear whether similar results can be obtained for extensions of the basic path planning problem.

Instead of assuming things about the robot environment, many researchers have simply dropped the requirement of completeness for the planner. Heuristic planners have been developed that solve particularly difficult problems in impressively low running times. However, the same planners also fail or consume prohibitive time on other problems. For autonomous robots in realistic environments this might be a problem, since one cannot predict the path planning problems such robots will face. So, on one hand, completeness is a preferred property of motion planners for autonomous robots, while, on the other hand, only heuristic algorithms are capable of solving many of the practical problems that people are interested in. This has lead to the design of path planners that satisfy weaker forms of completeness, in particular resolution completeness and probabilistic completeness.

In this chapter we deal with our proposed planner called straight line navigation (SLN) developed to generate a relatively short path autonomously. This idea is successful if, for a given problem, the estimated path guaranteed to be solved autonomously, in a definite elapsed time and optimized to a certain limit. Moreover, the planner exhibits high flexibility to deal with different types of obstacles and has the capability to generate general purpose path data that can be treated by any platform from any vendor [6, 113].

5.2 Relevant Research Work

For mobile robots, the existence of a desired path between two configurations is equivalent to the existence of a collision free path, due to the fact that for any collision free path there exists a desired path lying arbitrarily close to it, e.g. the linearized and smoothed paths of SLN algorithm discussed later. This fundamental property has led to a family of algorithms, decomposing the search in two phases. They first try to solve the geometric problem (i.e., the geometrical description of the robot platform and its surrounding obstacles). Then they use the obtained collision-free path to build a desired one. So in the first phase the decision problem is solved, and only in the second phase are the non-holonomic constraints taken into account. Such approaches were developed for non-holonomic robots, using optimal

control to approximate the geometric path. Several path planning algorithms have been developed to generate a relatively short path for non-holonomic robots, as

5.2.1 Visibility Graph Path Planning

This method produces the shortest path with respect to the metric. A visibility graph is formed by connecting all visible vertices, the start point and the end point, to each other. For two points to be visible, no obstacle can exist between them. Paths exist on the perimeter of obstacles. However, the close proximity of paths to obstacles makes it dangerous. It starts with a map of the world and draws lines of sight from the start and goal to every corner of the world and vertex of the obstacles, not cutting through any obstacles. After that lines of sight are drawn from every vertex of every obstacle like above. Lines along edges of obstacles are lines of sight, too, since they don't pass through the obstacles. If the map is in configuration space(C-space), each line potentially represents part of a path from the start to the goal.

Barraquand and **Latombe** have proposed a heuristic brute-force approach to motion planning for non-holonomic robots. It consists of heuristically building and searching a graph whose nodes are small axis-parallel cells in C-space. Two such cells are connected in the graph if there exists a basic path between two particular configurations in the respective cells. The completeness of this algorithm is guaranteed up to an appropriate choice of certain parameters, and it does not require local controllability of the robot. The main drawback of this planner is that when the heuristics fail it requires an exhaustive search in the discretized C-space. Furthermore, only the cell containing the goal configuration is reached, not the goal configuration itself. Hence the planner is inexact. Nevertheless, in many cases the method produces nice paths with a minimum number of reversals. For systems of higher dimension, however, it becomes too time consuming. Ferbach builds on the approach of Barraquand and Latombe method in his progressive constraints algorithm in order to solve the problem in higher dimensions. First a geometric path is computed. Then the non-holonomic constraints are introduced progressively in an iterative algorithm. Each iteration consists of exploring a neighborhood of the path computed in the previous iteration, searching for a path that satisfies more accurate constraints. Smooth collision-free paths in non-trivial environments were obtained. The algorithm, however, does not satisfy any form of completeness [91, 111].

5.2.2 Voronoi Diagram

The concept of Voronoi diagrams has been around for at least four centuries. In his treatment of cosmic fragmentation in "Le Monde de Descartes, ou Le Traite de la Lumiere", published in 1644, Descartes uses Voronoi-like diagrams to show the disposition of matter in the solar system and its environment. The first presentations of this concept appeared in the work of G. L. Dirichlet (1850) and G. M. Voronoi (1908). Although the concept has been around for a long time, algorithms for computing Voronoi diagrams did not start appearing until the 1970's.

K.E. Hoff et. al. presented an approach for computing generalized 2D and 3D Voronoi diagrams using interpolation-based polygon rasterization hardware. He computed a discrete Voronoi diagram by rendering a three dimensional distance mesh for each Voronoi site. The polygonal mesh is a bounded-error approximation of a (possibly) non-linear function of the distance between a site and a 2D planar grid of sample points. For each sample point, he computed the closest site and the distance to that site using polygon scan-conversion and the Z-buffer depth comparison. He constructed distance meshes for points, line segments, polygons, polyhedra, curves, and curved surfaces in 2D and 3D. Then, he generalized to weighted and farthest-site Voronoi diagrams, and presented efficient techniques for computing the Voronoi boundaries, Voronoi neighbors, and the Delaunay triangulation of points. He also showed how to adaptively refine the solution through a simple windowing operation. His algorithm has been implemented on SGI workstations and PCs using OpenGL, and applied to complex datasets. Moreover, he demonstrated the application of his algorithm to fast motion planning in static and dynamic environments, to selection in complex user-interfaces, and the creation of dynamic mosaic effects [208, 86, 66].

5.2.3 Potential Field Method

The potential field method has commonly been used for autonomous mobile robot path planning in the past decades e.g. Andrews and Hogan (1983), Krogh (1984), Khatib (1985), Khosla and Volpe (1988), Warren (1989-1990), Borenstein and Koren (1989-1991), Hussien (1989), Rimon (1990), Canny and Lin (1990), Latombe (1991), Rimon and Koditschek (1992), Kim and Khosla (1992), Ko and Lee (1996), Chuang and Ahuja (1998), Veelaert and Bogaerts (1999). The basic concept of the potential field method is to fill the robot's workspace with an artificial potential field in which the robot is attracted to its target position and is repulsed from the obstacles. This method is particularly attractive because of its elegant mathematical analysis and simplicity. Most of the previous studies use potential

field methods to deal with mobile robot path planning in stationary environments where targets and obstacles were all stationary. However, in many real-life implementations, the environments are dynamic. Not only the obstacles are moving, but also the target. The most destructing disadvantage of the potential field method is the presence of local minima.

In an effort to solve the problem of motion planning in a dynamic environment where the obstacles are moving, one approach is to include time as one of the dimensions of the modeled world and thus the moving obstacles can be regarded as stationary in the extended world e.g. Fujimura and Samet (1989), Shih and Lee (1990), Conn and Kam (1998). Accordingly, the dynamic motion planning problem is reduced to motion planning in stationary environments. The major problem in this approach is that it always assumes that the trajectories of the moving obstacles are known a priori, which is often inapplicable in real applications. Another approach was proposed in Ko and Lee (1996) and Hussien (1989) which extended the potential field method to the problems of moving obstacle avoidance by constructing repulsive potential functions which take into account the velocity information. In Ko and Lee (1996), though the velocity of the obstacle is considered when building the repulsive potential, the velocity of the robot is not taken into account. This is inadequate because the possibility of the collision between the robot and obstacle depends on the relative position and velocity between them. The repulsive potential function in Hussien (1989) makes fully use of the velocity information of the robot and the obstacle. However, it was assumed that the relative velocity of the robot with respect to the obstacle is invariant irregardless of the position of the robot and its partial derivatives in terms of position is zero. This assumption is unrealistic as the relative velocity is actually a function of the position of the robot and its derivatives in terms of position cannot be considered as zero all the time. Both methods deal with the obstacle avoidance problem with a stationary target [165, 88, 103, 107, 71].

5.2.4 The Probabilistic Path Planner (PPP)

The probabilistic path planner has been applied to various types of non-holonomic robots. An advantage of this approach is the fact that a road-map is constructed just once, from which paths can subsequently be retrieved quasi-instantaneously. Also, local robot controllability is not required. A critical point of the PPP when applied to non-holonomic robots is however the speed of the non-holonomic local planner. For non-holonomic robots very fast local planners have been developed. Thanks to this, PPP applied to the non-holonomic robots resulted in fast and probabilistically complete planners for non-holonomic robots that move both for-

wards and backwards. Moreover, a local planner is integrated into PPP, that uses exact closed form solutions for the kinematic parameters of a robot [193].

5.2.5 Bug Algorithms

These algorithms pertain to the reactive path planning family. The Bug algorithms were developed mainly by V. J. Lumelsky et. al. [122, 90]. The idea behind both algorithms (Bug1 and Bug2) is that the robot knows the direction in which it should head from its initial position toward its goal, but it does not know anything about the obstacles that lie in between. The basic idea is to start heading towards the goal (since the correct direction is known) and see whether any obstacles present themselves. We will call the straight line between the robot's initial position and the goal the S-line. If an obstacle is in the way of the robot, i.e., if it is interposed on the S-line between the starting point and the goal, the two Bug algorithms deal with that obstacle slightly differently:

Bug1 specifies that the robot should circumnavigate the entire obstacle. As it does so, it should remember whatever point along the obstacle's perimeter is closest to the goal. (The insect analogy would be a scent that grows stronger with proximity to a nest or meal) Once the robot has returned to (approximately) the point at which it originally hit the obstacle, it continues following the perimeter until it returns to that remembered closest point. Upon reaching that remembered point, it departs the obstacle and continues along a new line toward the goal. This process repeats if there are other obstacles in the way.

Bug2 takes sharper approach. If an obstacle presents itself along the S-line, the robot again wall-follows along the perimeter of that obstacle. Rather than circumnavigating the obstacle, however, Bug2 specifies that as soon as the robot reaches another point along the S-line that is closer than the original point of contact with the obstacle, the robot should leave the obstacle perimeter and continue heading toward the goal. This procedure repeats for any additional obstacles that arise.

Polar Bug This algorithm is proposed by R. D. Schraft et. al. to implement the Vis-Bug algorithm in real-time using a Sick LMS 200 laser scanner. This framework has been applied to develop a robot guidance system for Berlin museum visitors using three mobile robots that interact with the visitors spontaneously. The polar bug algorithm returns to the perimeter point closest to the goal. In that case, it is preferable to return along the shorter path [173, 77, 174].

The Bug algorithms are to be implemented using the wall-following. We can assume that the obstacles will consist of walls meeting at right angles (an indoor application). The S-line need not, however, be parallel or perpendicular to the obstacles' walls. In implementing these algorithms, there are a number of choices to make, e.g., which direction to start heading around an encountered obstacle. It doesn't matter which direction one chooses, as long as one stays consistent.

5.3 Straight Line Navigation (SLN) Algorithm

We now present our proposed SLN path-planning algorithm used to estimate the shortest path for non-holonomic robots autonomously. The idea of using the straight line as a base in estimation of a path is used in several algorithms such as Bug or ALG. In contrast to these on-line reactive algorithms the SLN algorithm operates on a previously computed full or partial map of the environment. The primitive definition of the straight line is "the shortest path between two points". We try to benefit from this idea in estimating the shortest path autonomously. The complete solution presented in this framework starts with a map using the vector mapping paradigm. This map is converted into a contour graph by enlarging the obstacle's boundaries. This graph is used as a base to develop the SLN-algorithm. This algorithm can be implemented in 5 phases [6], see figure (5.1). The 5 phases of the SLN algorithm can be accomplished as follows:

5.3.1 Initialization Phase

In this phase the parameters of the contour graph are defined, such as the set O of obstacles o_i, the start position S_0 and the goal position of the robot $T_{x,y}$. Each of these obstacles is defined by a set of properties: type, location and contour, see the SLN properties diagram shown in figure (5.1). The initialization phase comprises the following preprocessing operations:

- convert the irregular obstacles to regular forms (rectangle, circle, polygones).
- calculate the extended margins (safety, robot radius, relaxation).
- find the interposing obstacles on the S-line.

5.3.2 Segmentation Phase

The contours of the obstacles are intersected with a straight line linking the start with the goal position. Hence, the contour of each obstacle will be split into two segments and one of them is shorter than the other or equal. If we have n

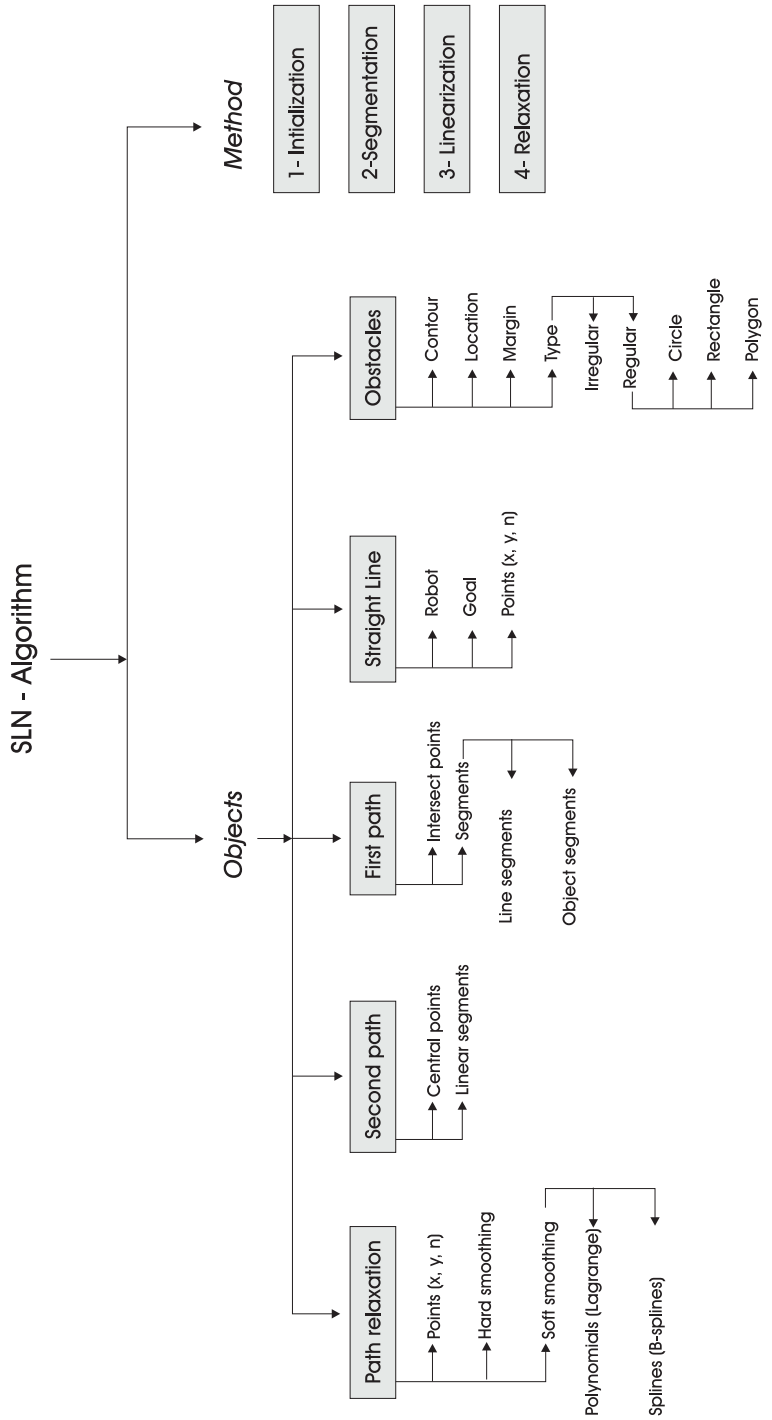


Figure 5.1: The structure of the SLN Algorithm

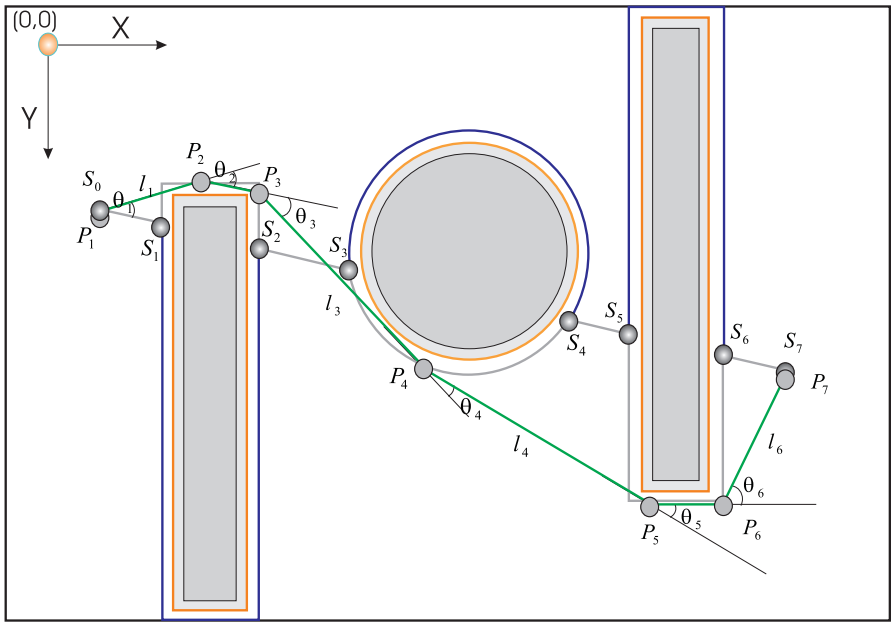


Figure 5.2: SLN path planning

obstacles and we start with point S_0 then the obstacles divide the line in $n+1$ line segments $line_i = \overrightarrow{S_{2i}, S_{2i+1}}$ for $0 \leq i \leq n$ and $2n+1$ points $S_0, S_1, \ldots, S_{2n+1}$ and line segments for, and object contour segments cont_i connecting point S_{2i-1} with point S_{2i} over a series of unnamed intermediate points on the shorter side of the object contour, with $1 \leq i \leq n$. Line segments $line_{i-1}$ and $line_i$ are connected by an object contour segment $cont_i$. See figure (5.2, 5.9). A question is, how can we find the intersection between the S-line connecting the start and goal points with the obstacle boundaries. The following algorithms show how this is done:

Line-Circle Intersection

A line determined by two points (x_1, y_1) and (x_2, y_2) may intersect a circle of radius r and center $(0, 0)$ in two imaginary points, a degenerate single point (corresponding to the line being tangent to the circle), or two real points. Defining:

$$d_x = x_2 - x_1 \tag{5.1}$$

$$d_y = y_2 - y_1 \tag{5.2}$$

$$d_r = \sqrt{d_x^2 + d_y^2} \tag{5.3}$$

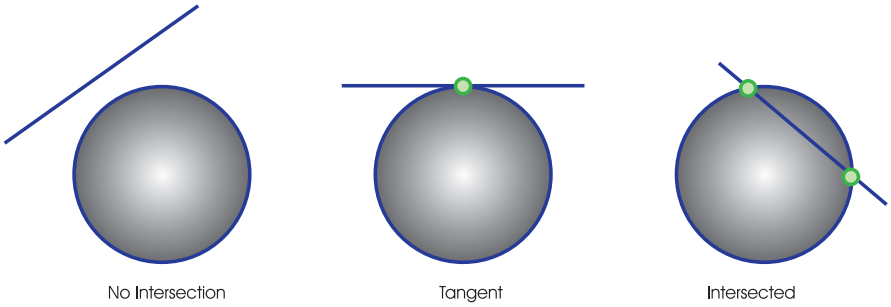


Figure 5.3: Line-circle intersect

$$D_c = \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} = x_1 y_2 - x_2 y_1 \tag{5.4}$$

gives the points of intersection as

$$x = \frac{D_c d_y \pm d_y \, d_x \sqrt{r^2 d_r^2 - D_c^2}}{d_r^2} \tag{5.5}$$

$$y = \frac{D_c d_x \pm |d_y| \sqrt{r^2 d_r^2 - D_c^2}}{d_r^2} \tag{5.6}$$

The discriminant

$$\Delta \equiv r^2 d_r^2 - D_c^2 \tag{5.7}$$

therefore determines the incidence of the line and circle as summarized in the following table.

Δ	**Incidence**
$\Delta < 0$	no intersection
$\Delta = 0$	tangent
$\Delta > 0$	intersection

Table 5.1: Definitions of intersections

More about line-circle intersection can be reviewed in [84, 157]

Line-Line Intersection

The intersection of two lines, in two dimensions containing the points (x_1, y_1) and (x_2, y_2) on the first one, while (x_3, y_3) and (x_4, y_4) lies on the second line, see figure (5.4), is given by:

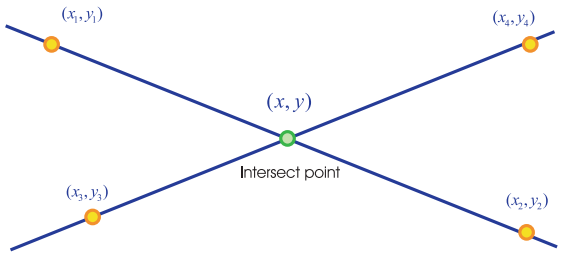


Figure 5.4: Line-line intersect

$$x = \frac{\begin{vmatrix} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} & \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix} \\ \begin{vmatrix} x_3 & y_3 \\ x_4 & y_4 \end{vmatrix} & \begin{vmatrix} x_3 & 1 \\ x_4 & 1 \end{vmatrix} \end{vmatrix}}{\begin{vmatrix} \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix} & \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \end{vmatrix} \\ \begin{vmatrix} x_3 & 1 \\ x_4 & 1 \end{vmatrix} & \begin{vmatrix} y_3 & 1 \\ y_4 & 1 \end{vmatrix} \end{vmatrix}} = \frac{\begin{vmatrix} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} & x_1 - x_2 \\ \begin{vmatrix} x_3 & y_3 \\ x_4 & y_4 \end{vmatrix} & x_3 - x_4 \end{vmatrix}}{\begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_3 - x_4 & y_3 - y_4 \end{vmatrix}} \tag{5.8}$$

$$y = \frac{\begin{vmatrix} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} & \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \end{vmatrix} \\ \begin{vmatrix} x_3 & y_3 \\ x_4 & y_4 \end{vmatrix} & \begin{vmatrix} y_3 & 1 \\ y_4 & 1 \end{vmatrix} \end{vmatrix}}{\begin{vmatrix} \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix} & \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \end{vmatrix} \\ \begin{vmatrix} x_3 & 1 \\ x_4 & 1 \end{vmatrix} & \begin{vmatrix} y_3 & 1 \\ y_4 & 1 \end{vmatrix} \end{vmatrix}} = \frac{\begin{vmatrix} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} & y_1 - y_2 \\ \begin{vmatrix} x_3 & y_3 \\ x_4 & y_4 \end{vmatrix} & y_3 - y_4 \end{vmatrix}}{\begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_3 - x_4 & y_3 - y_4 \end{vmatrix}} \tag{5.9}$$

where $\begin{vmatrix} a & b \\ c & d \end{vmatrix}$ denote determinant. Other treatments are given by Antonio (1992) and Hill (1994) [10, 72, 85].

5.3.3 Linearization Phase

In this phase the second path is generated by linearizing the sections of the first path. We successively try to combine two or more path segments to a larger path segment by vector addition of their vectors under one condition that the resultant vector doesn't intersect with any obstacles. We keep the new combined path segment if it does not intersect one of the obstacles (except on the contour only). The points on this second path are denoted by $P_0, \ldots, P_m$, where $P_0 = S_0$ is the original robot starting position, $P_m = S_{2n+1}$ is the target position, and the other P_i are points on the contour of an object, mostly not identical to any S_j. The second path may then be described as a sequence of rotations θ_i and translations l_i of the robot at the central control points $P_i = (x_i, y_i)$ with $1 \leq i \leq m$.

$$\|l_i\| = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{5.10}$$

$$\theta_{\mathrm{i}} = \mathrm{asin}\left(\frac{\mathrm{y_i - y_{i-1}}}{\mathrm{l_{i-1}}}\right) - \mathrm{asin}\left(\frac{\mathrm{y_{i+1} - y_i}}{\mathrm{l_i}}\right) \tag{5.11}$$

5.3.4 Minimization Phase

The maximum number of possible paths between the start and goal position is 2^n, where n is the number of obstacles along the line connecting the start to the goal position. In real world scenarios n is usually very small. Minimizing the length of the path can be obtained by selection of the minimum set of segments. The resulting path is the shortest one.

5.3.5 Relaxation Phase

The Linearization phase generates a set of central points of the second path. To relax/smooth this path, the interpolation function models this set of tabulated function values or discrete data into a continuous function. We call such a process data fitting. The continuous function (smoothed path) may characterize the relation between variables x and $y = f(x)$ more than their correspondence at the discrete central points [195]. It can be used to estimate variable y corresponding to a non-nodal point $x \in [a; b] - \{x_i\}$ (interpolation) or to a point outside of $[a; b]$ (extrapolation). For the same set of data, the interpolation changes with the selection of subspaces. The following are commonly used for interpolations:

a) polynomials (Lagrange)

b) splines (B-spline)

Throughout this framework we will discuss both approaches and the interconnection between them.

Interpolation with Polynomials

Let's discuss the theory of Lagrange on interpolations and how could we benefit from this theory to relax a path for a robot based on our proposed algorithm (SLN) [58].

There exists only one n^{th} degree polynomial that passes through a given set of $n+1$ central points $(x_i,\ y_i)$. It's form is (expressed as a power series):

$$y = f_n(x) \quad = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \tag{5.12}$$

where, $x = [x_0,\ x_1,\ \ldots\ , x_n]$ and $y = [y_0,\ y_1,\ \ldots\ , y_n]$ that passes through all the central points. The Lagrangian polynomial approach employs a set of n^{th} order polynomials, $L_i(x)$, such that:

$$f_n(x) = \sum_{i=0}^{N} L_i(x) * y_i \tag{5.13}$$

where $L_i(x)$ satisfies:

$$L_i(x) = \begin{cases} 1 & at\ \ x = x_i \\ \\ 0 & at\ \ x = x_j,\ j \neq i \end{cases} \tag{5.14}$$

$$L_i(x) = \frac{\prod\limits_{\substack{j=0 \\ j \neq i}}^{n} (x - x_j)}{\prod\limits_{\substack{j=0 \\ j \neq i}}^{n} (x_i - x_j)} \tag{5.15}$$

Example:
Let us apply Lagrange interpolation on a dataset of 6 central points, where, $x = [100, 150, 250, 550, 650, 800]$ and $y = [250, 350, 350, 50, 50, 150]$. Figure (5.5) shows the interpolation of the given central points. The same dataset is used later to show the capability of different B-spline interpolations to smooth the linearized path of the SLN-algorithm by which the comparison of quality measure among the presented algorithms will be eased.

From the previous conditions we note:

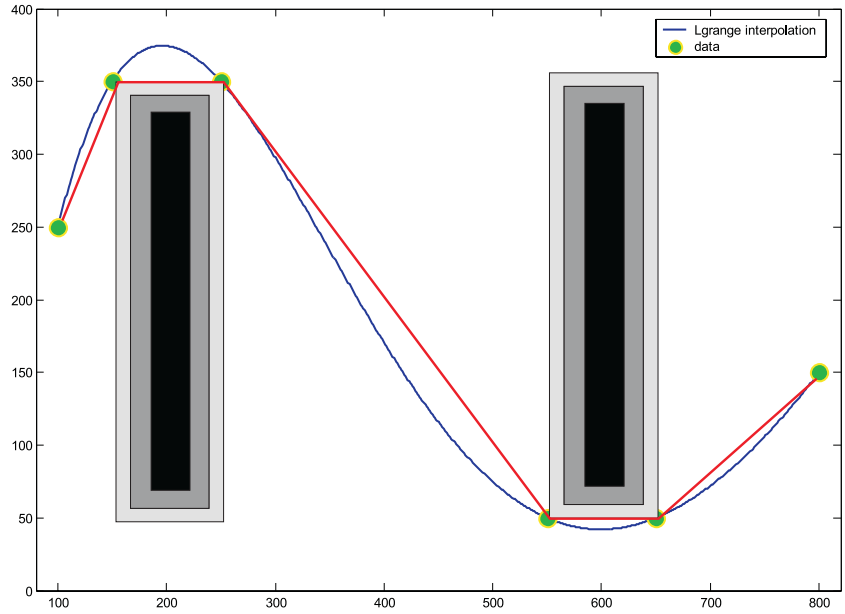


Figure 5.5: Lagrange interpolation of SLN linearized path

- The interpolating function is used to replace or simplify the original linear path with certain smoothness properties preserved at the discrete interpolation central nodes and their neighborhood. To evaluate a complicated function one may pre-compute the function at certain reference or central nodal points and evaluate the function at the other points by the interpolating function. We may call such a process path simplification.
- Polynomial evaluation with the Lagrange representation is of high complexity when the size n of central nodes dataset is large.
- Since each of the Lagrange polynomials is of degree $n-1$, there are cancellations in degree when the data are from a polynomial of a lower degree.
- The evaluation complexity can be reduced for the case that the interpolation central points are equally spaced, and uniformly distributed.

Lagrange Interpolation Accuracy:
We want to estimate the accuracy of interpolation at a non-nodal point in x. The following accuracy estimation is based on a smoothness assumption and is in the sense of L_∞ norm.

$$\varepsilon_x = L_i(x) f^{(n+1)}(\xi) \qquad x_0 \leq \xi \leq x_N \tag{5.16}$$

where $f^{(n+1)} = (n+1)^{th}$ derivative of f w.r.t. x evaluated at ξ

- The Taylor theorem may be used for the proof of the above theorem.
- The approximation error at the nodal central points is zero.
- The interpolation error at any non-nodal point decreases as the number of interpolation nodes increases.

Path relaxation based on Lagrange interpolation takes into considerations the whole segments of the linearized path. Another approach is to relax the path segments in intervals using basic spline (B-spline) interpolations [43, 57]. Moreover, the B-spline is used as a membership control function in adaptive systems discussed in chapter (6).

B-splines Interpolation

B-splines form a truly piecewise basis for the spline family [201, 202, 43] for a set of infinite data points that represent the central points of the linearized path, $(x_i,\ i = -\infty, \infty)$ where $x_i < x_{i+1}$ for any i, the B-splines of degree k are defined as:

$$B_i^k(x) = \frac{x - x_i}{x_{i+k} - x_i} B_i^{k-1}(x) + \frac{x_{i+k+1} - x}{x_{i+k+1} - x_{i+1}} B_{i+1}^{k-1}(x), k \geq 1 \tag{5.17}$$

For any dataset of central points $(x_i, y_i, i = 1, 2, \ldots, n)$, the B-spline is given by:

$$f(x) = \sum_{i=1}^{n} D_i B_i^0(x) \tag{5.18}$$

where D_i are constants determined from the data. Recursive definition is one basic feature of B-splines, which enables the generation of B-splines of arbitrary orders with the incremental smoothness for a given set of central nodes. The other most important properties of B-splines are:

Partition of unity: $\sum_{i=1}^{n} B_i^k(x) = 1$

Positivity: $B_i^k(x) \geq 0$ for all x

Local support: $B_i^k(x) = 0$ for all $x \notin [x_i, x_{i+k}]$

1. Linear B-spline

For any dataset of central points $(x_i, y_i, i = 1, 2, \ldots, n)$, the constant B-spline is given by:

$$f(x) = \sum_{i=1}^{n} y_i B_i^0(x) \tag{5.19}$$

and

$$B_i^0(x) = \begin{cases} 1, & x \in [x_i, x_{i+1}) \\ \\ 0, & otherwise \end{cases} \tag{5.20}$$

where i and k are integers and $1 \geq B_i^k(x) \geq 0$. Each $B_i^0(x)$ is nonzero in exactly one interval has a discontinuity at x_{i+1}. That simply extends each y_i across each i^{th} interval.

For $k = 1$, $B_i^1(x)$ can be derived from equations (5.17)and (5.18) as:

$$\begin{aligned} B_i^1(x) &= \tfrac{x-x_i}{x_{i+1}-x_i} B_i^0(x) + \tfrac{x_{i+2}-x_i}{x_{i+2}-x_{i+1}} B_{i+1}^0(x) \\ &= \begin{cases} \frac{x-x_i}{x_{i+1}-x_i}, & x \in [x_i, x_{i+1}] \\ \\ \frac{x_{i+2}-x}{x_{i+2}-x_{i+1}}, & x \in [x_{i+1}, x_{i+2}] \\ \\ 0, & elsewhere \end{cases} \end{aligned} \tag{5.21}$$

which has two non-zero parts, in intervals $x \in [x_i, x_{i+1}]$ and $x \in [x_{i+1}, x_{i+2}]$. These intervals are called the support of $B_i^1(x)$. On each of its support intervals, $B_i^1(x)$ is a linear function and its location and slope are solely determined by the distribution of the x_i's. $B_i^1(x)$ has a peak value of 1 at x_{i+1} and is continuous there.

It is interesting to note that $B_i^1(x)$ has some connection to the elementary Lagrange interpolating polynomial. If we write the linear Lagrange interpolating polynomial for the two-point dataset of $(x_j, \ \ j = i \ \ and \ \ i + 1)$, the two elements would be:

$$\begin{aligned} L_1^2(x) &= \tfrac{x-x_{i+1}}{x_i-x_{i+1}} \\ L_2^2(x) &= \tfrac{x-x_i}{x_{i+1}-x_i} \end{aligned} \tag{5.22}$$

which are the second part of $B_i^1(x)$ and the first part of $B_i^1(x)$, respectively. This agreement does not extend to higher order B-splines. In general, the 1^{st} order B-spline is constructed from:

$$f(x) = \sum_{i=-\infty}^{\infty} D_i B_{i-1}^1(x) \tag{5.23}$$

The linear spline on a given data interval is also the Lagrange interpolation formula for the dataset consisting of the two end points of the same data interval. So let's discuss the relationship of $B_i^1(x)$ with the linear splines. It turns out that for any dataset $(x_i,\ y_i,\ i = 1, 2, \ldots, n$, where n can be infinity), the following function is a linear spline:

$$f(x) = \sum_{i=1}^{n} y_i B_{i-1}^1(x) \tag{5.24}$$

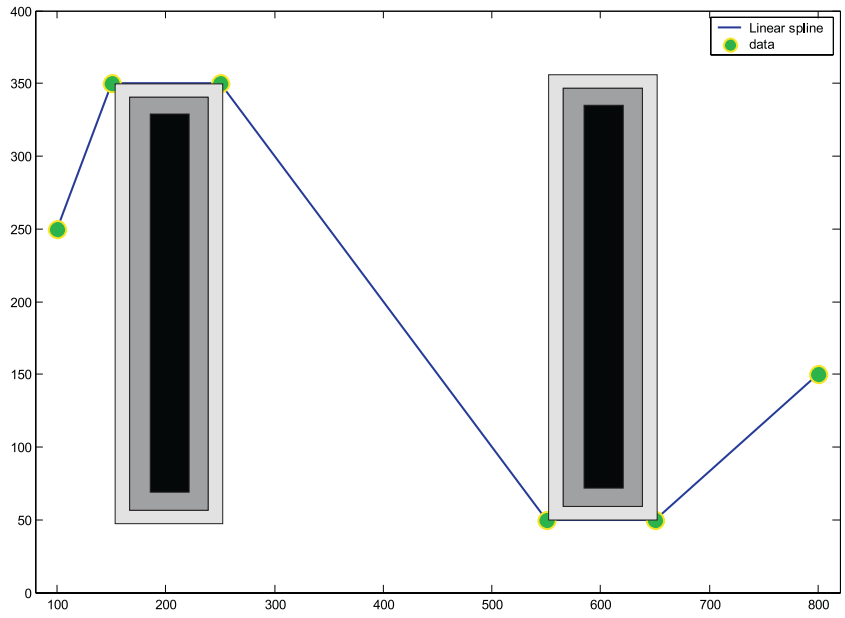


Figure 5.6: Linear spline interpolation of SLN central nodes

Generally for linear splines, two points are used to draw a straight line for each interval. Then for each data point, its x and y information will contribute to two splines. In the case of $B_i^1(x)$, this contribution is built into its expression through its two legs. This is the essence of all B-splines and one reason why they are called basis splines.

Polynomial splines and Lagrange interpolating polynomials discussed before can all be viewed as functions over path segment(s). The emphasis here is on the intervals. A polynomial is set up over the path segment(s), then, we go to its end

points for help on determining the coefficients of the polynomials.

B-splines, however, should be viewed more as belonging to the central points. Each point function covers certain path segment(s). To build splines through B-splines, we will always use the form similar to that in equation (5.23) The y information is applied to the point function and they are done (almost, anyway) with the building of the splines. This feature will be more evident in the case for $B_i^2(x)$.

2. Quadratic B-spline
Now for $k = 2$, the quadratic spline can be estimated from equation (5.17) and (5.18), we have

$$\begin{aligned} B_i^2(x) &= \tfrac{x-x_i}{x_{i+2}-x_i} B_i^1(x) + \tfrac{x_{i+3}-x}{x_{i+3}-x_{i+1}} B_{i+1}^1(x) \\ B_i^2(x) &= \tfrac{x-x_i}{x_{i+2}-x_i} \left(\tfrac{x-x_i}{x_{i+1}-x_i} B_i^0(x) + \tfrac{x_{i+2}-x}{x_{i+2}-x_{i+1}} B_{i+1}^0(x) \right) \\ &\quad + \tfrac{x_{i+3}-x}{x_{i+3}-x_{i+1}} \left(\tfrac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1}^0(x) + \tfrac{x_{i+3}-x}{x_{i+3}-x_{i+2}} B_{i+2}^0(x) \right) \end{aligned} \tag{5.25}$$

$$B_i^2(x) = \begin{cases} \frac{(x-x_i)^2}{(x_{i+1}-x_i)(x_{i+2}-x_i)}, & x \in [x_i, x_{i+1}] \\ \frac{1}{x_{i+2}-x_{i+1}} \left[\frac{(x-x_i)(x_{i+2}-x)}{x_{i+2}-x_i} + \frac{(x-x_{i+1})(x_{i+3}-x)}{x_{i+3}-x_{i+1}} \right], & x \in [x_{i+1}, x_{i+2}] \\ \frac{(x_{i+3}-x)^2}{(x_{i+3}-x_{i+1})(x_{i+3}-x_{i+2})}, & x \in [x_{i+2}, x_{i+3}] \\ 0, & elsewhere \end{cases} \tag{5.26}$$

Also, the components of $B_i^2(x)$ differ from the quadratic Lagrange polynomials. The L_1^3 term of Lagrange interpolation goes negative. Therefore, they are different, see figure (5.7)

3. Cubic B-spline
The interpolation of cubic spline 3^{rd} B-spline can be accomplished using the same roles from Equation (5.17) and (5.18), see figure (5.8).

The cubic B-spline gives the minimum interpolation error in almost all cases. The comparison of figures (5.5), (5.7) and (5.8) reveals this result.

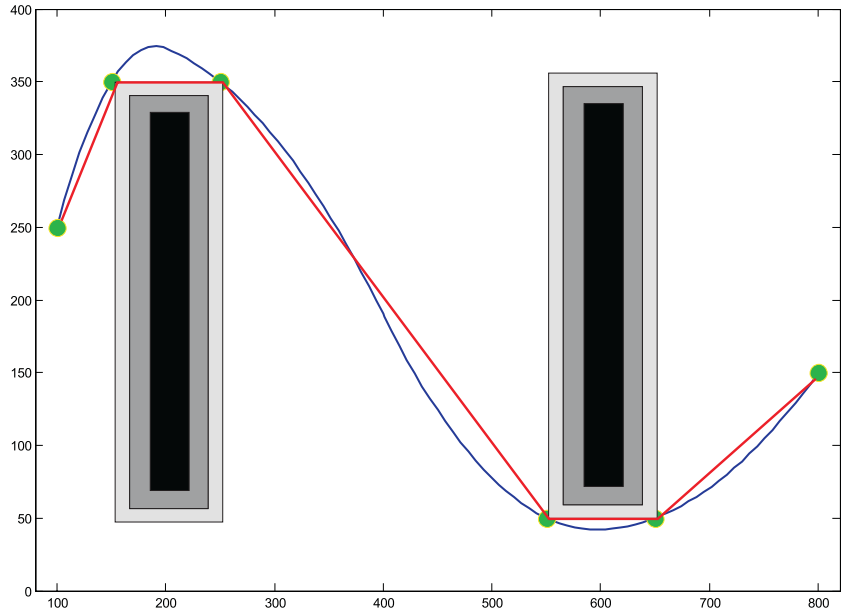


Figure 5.7: Quadratic spline interpolation of SLN linearized path

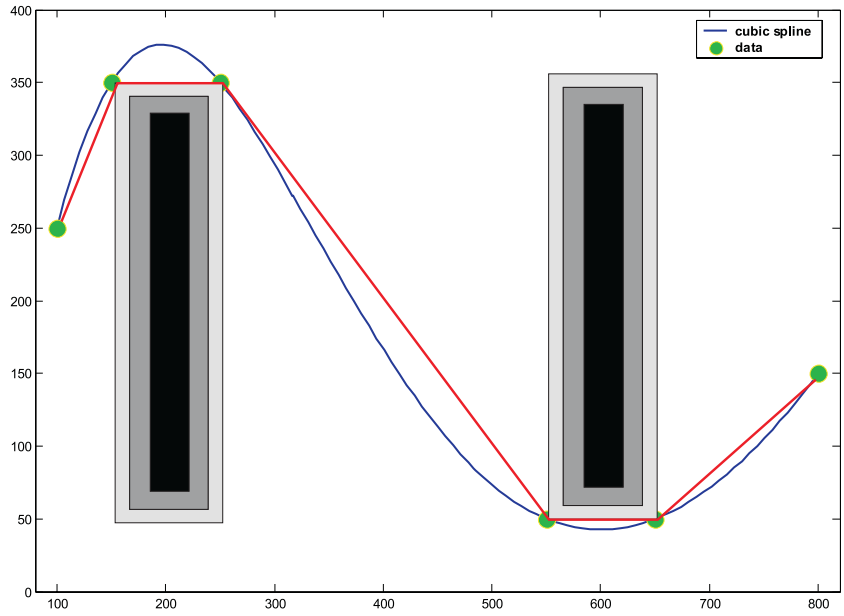


Figure 5.8: Cubic spline interpolation of SLN linearized path

B-Spline Approach versus Lagrange Polynomials

To understand the similarities and differences between Lagrange and B-spline interpolations the following points have to be considered.

- B-Spline functions are defined locally.
- Translation of a central point in B-spline interpolation has only local influence on the shape of the curve while the central points have global influence on the shape of the Lagrange curve.
- In B-spline interpolation, additional central points can be inserted without degree elevation and the smoothness of the junction of adjoining segments can be influenced easily.
- Lagrange polynomial evaluation with the Lagrange representation is of high complexity when the size n of the dataset of central nodes is large.
- B-spline functions are extended to be applied to several application techniques e.g. control systems, fuzzy logic etc.

Based on the preceding discussion we conclude that Lagrange interpolation is suitable for small size set of segments while B-spline based interpolation is useful if the set of segments is large.

5.4 Notes on the SLN Algorithm

The SLN simulator is a unit of the robot software kit (RSK) that has been developed to implement this algorithm. This simulator has a map designer and also accepts the contour graph format of the working environment. This algorithm provides the autonomous path planning and minimizes the path length. Figure (5.9) presents a screen shot of the SLN simulator (path designer). The map can be generated easily using the map designer or can be introduced as contour format files.

The elapsed time needed for the algorithm depends on many factors:

1. number of obstacles interposing on the S-line connecting the start and the goal points,
2. complexity of the map even if the number of the obstacles is equal,
3. obstacle distribution over the path,

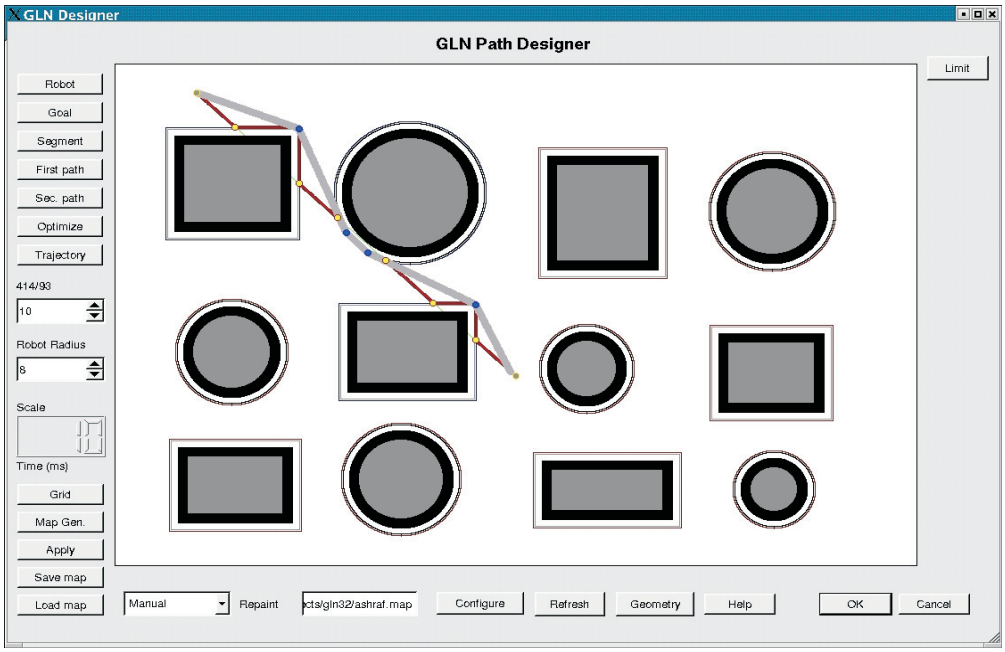


Figure 5.9: SLN simulator

No. of intersected objects	Total no. of objects	Elapsed time [ms]
1	3	7
2	6	18
3	8	38
4	8	49
4	15	27
2	4	32
4	12	29

Table 5.2: Elapsed time of the SLN-Algorithm in 7 different cases of study

4. size and type of obstacles to be avoided.

Table (5.2) indicates the elapsed time of 7 different cases of study. In general the elapsed time for 5 obstacles is approximately 35 ms, and we must take into consideration that the processing time has no linear relationship with the map construction. But in general the results encourage the implementation of the SLN algorithm on-line. Figure (5.11) shows two simulators the first one designs the path based on the SLN algorithm and the second relies on the Bug algorithm.

5.5 SLN versus Voronoi

Compared with the currently popular path planners based on generalized Voronoi diagrams the SLN algorithm has the following advantages and disadvantages:

- Using a Voronoi diagram, the robot must follow a path centered between the obstacles during navigation. This minimizes the probability of a collision, but makes the path longer than necessary. Using the SLN algorithm, the robot does not need to follow such a path, which can be considerably shorter, if the space among obstacles is large.
- The path generated by the SLN algorithm is not subject to the following problem of the generalized Voronoi algorithm: following corridors with open sideway doors or over hallways, the generalized Voronoi diagram path planner usually trigger some unexpected sideway robot motion.
- The SLN algorithm might be a bit more complicated to implement than the generalized Voronoi diagram algorithm.
- The theoretical time complexity of SLN algorithm is exponential in the number of obstacles along the straight line from start to goal, but not in the number of obstacles present in the map.

5.6 SLN versus Bug

Initially, the Bug algorithms (Bug1, Bug2, polar Bug and ALG) are classified as reactive path planning, this means that the robot uses its sensor data to design a path between two known points, start and goal points. Therefore , there is no map required to implement these algorithms and that is applicable in the presence of a single obstacle. In contrary, the SLN algorithm relies mainly on a given map that defines the obstacle characteristics and the start and goal points.

Bug specifies that the robot should circumnavigate the entire obstacle (Bug1) or the robot follows the wall along the perimeter of that obstacle (Bug2) to reach the goal. This method is repeated in the presence of multiple obstacles. On the other side, SLN selects the shortest path segments and linearizes the resultant path to shorten the final trajectory. Moreover, The SLN smooths the path that helps the motion of the robot not to be jerky as in Bug.

Figure (5.10) shows the difference between our proposed SLN algorithm and the original Bug algorithm.

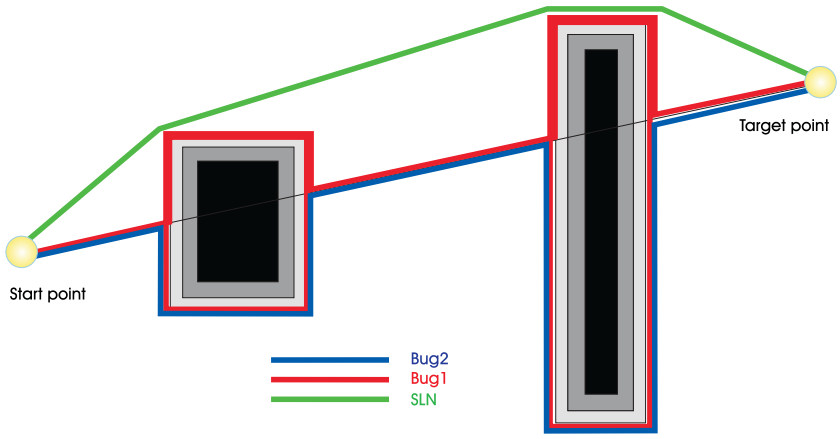


Figure 5.10: Bug and SLN generated paths

To span from the pure theoretical geometry of the SLN motion planning algorithm to the practical world, we designed the SLN-simulator which generates the path between the start and the goal points autonomously. This simulator builds a map of regular forms of obstacles (Circles, polygons and rectangles) and uses this map to design a path. We expect that the SLN approach will be extended in 3D environment under spatial and temporal conditions. A similar approach is adopted to design the path of mars rovers autonomously. Figure (5.11) shows two simulators the first one applies the SLN algorithm (Uni. of Tübingen) and the second one pertains to the Bug algorithms (Harvey Mudd College).

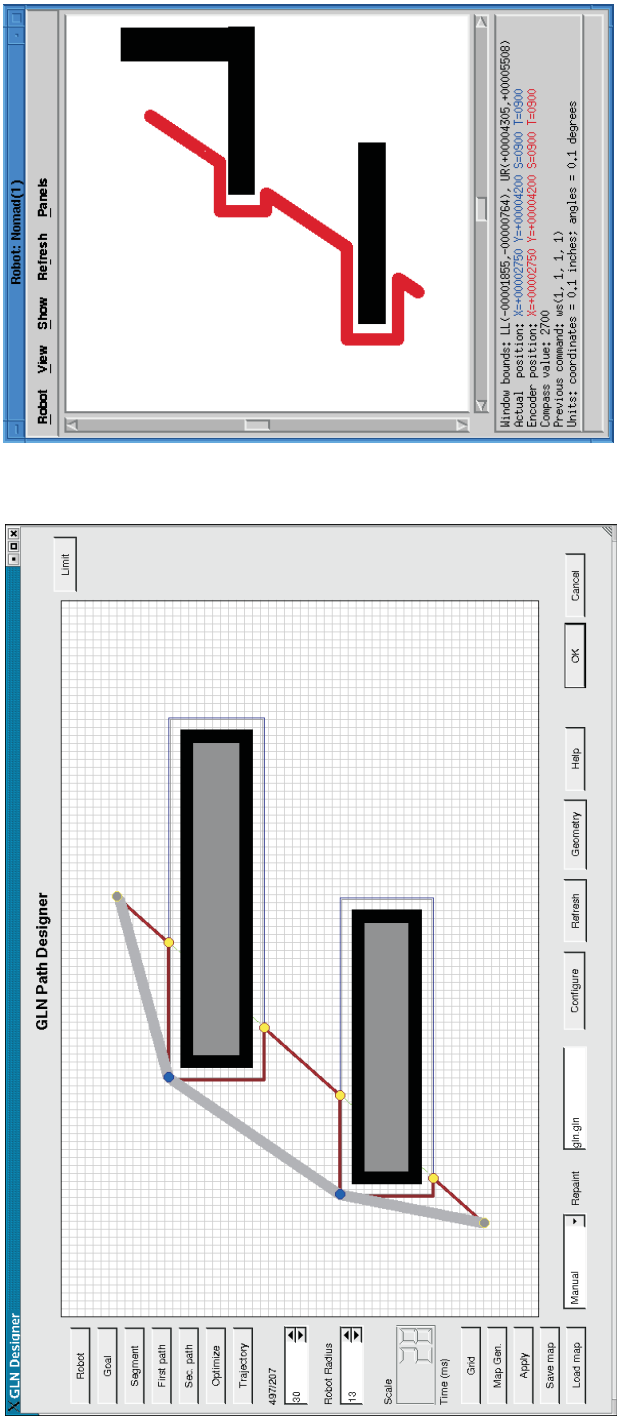


Figure 5.11: SLN and Bug path designers

Chapter 6

Adaptive Navigation

In the past few years it has become imperative for mobile robots to be more autonomous, adaptive, flexible and robust. Mobile robots tasks in indoor and outdoor terrains are being more complicated and simple fixed gain controls, e.g. PID controllers, do not fit the requirements of such duties. As these systems are conferred with increased autonomy, their control system must be able to overcome changing conditions as well as individual failures. We address the problem of designing robust, flexible control for mobile robots by endowing them with a self-adaptive control system. Self-adaptive control analyzes its performance and takes action to bring the performance to a satisfactory level. Using several novel approaches, our controller can diagnose, overcome, and adapt to adverse and rapidly changing conditions.

Throughout this chapter, an adaptive control technique is investigated to underlie safe robot navigation. The main objective of that is to preserve the stability, smoothness, efficiency and robustness of the overall system against fluctuations of system parameters or set points. The fine motion attained by adaptive behaviors is highly desirable in dynamic environments. Generally, that leads to strengthen system safety, reduce energy consumption, improve of precision and shortened elapsed task time.

In adaptive theory, two main approaches are prevalent, the first one is the quantitative, parametric, **stochastic Kalman filter** and the other one is the linguistic, non-parametric, **fuzzy inference system**. Both approaches are successful and are capable to undertake various tasks of adaptive control systems professionally. A variety of techniques have been developed to cover the topics of modeling, estimation, forecasting, controller design, optimization, digital processing and adaptation in both theories.

Throughout this framework we combine both approaches to benefit from their advantages as far as possible. The resultant system is an intelligent deliberative integration that inherits the strengths of both the universe of discourse of fuzzy logic control (FLC) and the state space of Kalman filter. This fusion of rules leads to an improvement in the overall quality measure of our mobile robot.

6.1 Related Work

Autonomous robots operating in an unknown and uncertain environment have to cope with changes in the environment. Traditionally, robot software architecture has been built up according to deliberative reasoning in the manner of sensing, planning and action. It is obviously difficult to accommodate the sensory uncertainty and the environmental dynamics [33].

In [12, 13] the reactive or behavior-based architectures are able to handle the problems appearing in the deliberative architecture. The basic component in this method is a group of behaviors. Behaviors directly map sensory information into motor actions without complex reasoning. Behaviors can also operate concurrently to produce emergent behaviors for unknown environments. With this ability, robots become more robust. The key issues raised from behavior-based architectures are the behavior design and behavior co-ordination.

In [50, 21, 26, 204] the authors concentrate on the behavior design in which problems are caused by uncertain sensor data and imperfect motor action. Recently there is an increasing tendency to build up the mapping of sensory-action pairs by FLC. The mechanism of a FLC is that the uncertainty is represented by fuzzy sets and an action is generated co-operatively by several rules that are triggered to some degrees, and produce smooth and robust control outputs.

In [75, 52, 203] the problems in the design of a FLC are the setting of parameters of membership functions and the composition of fuzzy rules. They are classified into two types: **structure identification** and **parameter identification**. The structure identification of a FLC includes the partition of its input space, the selection of antecedent and consequent variables, the determination of the number of IF-THEN rules, and the initial position of membership functions. The parameter identification determines the parameters of membership functions.

Many learning approaches have been proposed to implement the adaptive controllers, including neural network (NN) based [22, 94], and genetic algorithms (GA) approaches [95]. The NN-based systems automatically determine or modify

their structure and parameters with unsupervised or supervised learning by representing the controller in a connectionist way. The problem for the NN-based control is that enough data have to be provided to train the networks. Moreover, the NN-based controllers are unable to deal with unexpected patterns. The GA-based and reinforcement learning (RL)-based controllers are two learning schemes that need a scalar response from the environment to show the action performance. Although the feedback from the environment for actions is needed, it can be a scalar value that is easier to collect than the desired-output data pairs in the real robot application and can be in any form without the differentiable requirement [143, 145].

6.2 Modeling of Robot Dynamics

Recently the importance of system analysis and modeling have been greatly increasing in several disciplines e.g. economics, biology, medicine, ecology and certainly in the field of robotics. Especially in technical applications, system analysis and model building are the most important factors for making the use of different design and control methodologies and theories. Applications of different kinds of mathematical models in these fields require more and more precise models of the system under consideration. Mathematical modeling is widely used to analyze and model system dynamics of stationary manipulators, but the greatest challenge is to model mobile system dynamics without any background about its kinematics and that is what we are going to present in this section.

Mathematical models were classified on a broad scale of aspects in previous studies. It is still practical to recall model classes, which are the most important from the perspective of system analysis. Mathematical models can be represented in the time domain, in the operator domain (e.g. Laplace, Z, etc. transforms) or in frequency domain. Regarding time domain, the continuous time models are closer to physical considerations, whereas the discrete-time system behavior is considered to be defined at a sequence of time instants related to measurements. The discrete-time models are closely related to implementation problems of digital processing. Many models have been defined by a given form but are dependent on a finite number of real parameters. Such models are said to be parameterized or **parametric models**, although there is no clear cut distinction from other models which are sometimes referred to as **non-parametric models**. Examples of parametric models are algebraic equations, differential equations, difference equations, systems of these equations, transfer functions, etc. In these cases, the model building consists of the determination of parameters in fixed structures or in flexible structures as what we applied. The theoretical model building always yields

a parametric model. A non-parametric model is the response obtained directly from an experimental analysis of the robot's performance. By various analytical methods a parametric model can be deduced from a non-parametric model. By experimental analysis of a parametric model can be obtained directly if the structure of the system can be defined a priori. Very often, especially in control theory, systems even if they exhibit non-linear behavior and therefore the use of non-linear models would be appropriate, the system dynamics is described by linear or quasi linear models which allows simpler mathematical manipulation and we relied on this method in modeling of mobile robot dynamics.

6.2.1 Signal and System Modeling

There are two different approaches to the characterization of dynamic systems: In linear systems theory, one can assume either some structure in the sensor signals (laser, compass and sonar signatures) or some structure in the system (robot modeling and control). Both techniques can be used for modeling. Attempts have been made to combine these two approaches e.g. harmonic identification techniques in the Fourier domain.

1^{st}approach: Structure on the sensor signal can be found using linear transforms. This approach does not take into account that the system has some structure. In the time domain filtering is a linear transformation. The **Fourier**, **Wavelet**, and **Karhunen-Loeve** transforms have compression ability and can be used to identify some structure in the signals. When we are using these transforms we do not take into account any structure in the system. Filtering a signal $s(t)$ (either output $\upsilon(t)$ or input $\lambda(t)$ is essentially finding some structure in the following vector

$$\begin{bmatrix} s(t) \\ q^{-1}s(t) \\ \vdots \\ q^{-n}s(t) \end{bmatrix} \tag{6.1}$$

where q^{-1} is the delay or back shift operator in the time domain, where $q^{-1}s(t) = s(t-1)$.

2^{nd}approach: Structure on the system can be found by fitting a model to the system using Gauss, Wiener, Kalman and soft computing techniques.

Throughout this study we employed both approaches to study the robot as a system equipped with a group of sensors and testify the success of our algorithms.

First, we applied the signal modeling to analyze sonar, laser and compass signals and we applied adaptive control techniques based on kalman filter and fuzzy rules to steer the robot adaptively during free navigation or in visual guidance.

6.2.2 A Priori and A Posteriori Modeling

Mathematical models can be developed in different ways: either purely theoretically based on the physical relationships, which are a priori known about the system, or purely empirically by experiments on the already existing system, or by a sensible combination of both. Models obtained by the first method are often called a **priori** or first principle or theoretical models, while models obtained in the second way are called a **posteriori** or experimental models. In case of theoretical analysis, the dynamic properties of the system are primarily taken care of by the respective balance equations. The laws of conservation supplemented with the necessary state-equations and phenomenological laws establish these balances. Theoretical model building becomes unavoidable if experiments in the respective plant cannot or must not be carried out. If the plant to be modeled does not yet exist, theoretical modeling is the only possibility to obtain a mathematical model.

6.2.3 Application of A Posteriori Models

In many cases the available knowledge is very limited and it can be very expensive to elaborate a priori models of robots. System or process identification is the modeling method based on experimental data. If experimental analysis of a process is performed, the input and output signals are measured. The measurements are then evaluated in an analysis procedure yielding a mathematical model of the robot dynamics. This analysis enables model building for robot dynamics of arbitrary structure. Analysis methods can be applied to diverse, complicated processes. By measuring input and output signals one obtains a model for the input-output behavior of the robot. The input signal can either be operating signals, e.g laser, sonar and compass, of the robot (usually of stochastic nature), or artificial test signals. The result of the analysis is an experimental model. Utilizing modern computers, very efficient analysis and parameter estimation methods have been developed. The mathematical models developed by these methods can be applied for a wide variety of diverse cases:

- Prediction or forecasting can serve to obtain better knowledge of the robot's performance, to verify theoretical models, to predict new phenomena, etc. It is important to represent external actions and external disturbances and using knowledge of statistical characteristics of random variables, as there is

usually little theoretical or practical possibility of determining such characteristics in advance.

- System analysis and design provide a rich field for the application of modeling and analysis, since application of models allows quantitative predictions to be made concerning crucial features of robot systems such as stability conditions and the development of predefined behavior in unknown terrains.
- Simulation based on mathematical models is widely used for the assessment of model complexity, for engineering design or for operator training, all of which require adequate modeling and adequate input.
- Models often serve as the basis of monitoring or supervision, error detection and process diagnosis in robot systems.
- Robot platforms in continuous operation require system optimization for their economy, which in turn requires very accurate modeling.
- Soft sensors can provide information on robot status variables which directly cannot be measured or when measurement is complicated or too slow. These variables can usually be computed from other measurements based on suitable models.

6.2.4 Stochastic Kalman Filter

The Kalman filter, named after Rudolf E. Kalman, is an optimal solution to the discrete data linear filtering problem [98, 78, 211, 89, 92]. The filter is derived by finding the estimator for a linear system, subject to additive white Gaussian noise, that meets the following three criteria: **1)** the estimator provides an unbiased estimate of the system state, **2)** the estimator provides a minimum variance estimate of system state, **3)** the estimator is a linear function. The Kalman filter comprises two phases, the first one maps model inputs to system states, while the second one maps the states to the outputs, see figure (6.1). The Kalman filter derivation begins with a definition of the system and assumptions. Consider a process $f(\upsilon, \lambda, \epsilon)$ given by the **state space model** defined by its linear difference equation form

$$\upsilon_{k+1} = f\left(\upsilon_k, \lambda_k, \epsilon_k\right) = A_k \upsilon_k + B_k \lambda_k + \epsilon_k \tag{6.2}$$

where υ_k is the $(n \times 1)$ system state vector at time step k and $\upsilon \in R^n$, λ_k is the system control vector at time step k and $\lambda \in R^l$, A_k is the $(n \times n)$ state transition matrix mapping υ_k to υ_{k+1}, B_k is the control matrix mapping λ_k to υ_k, ϵ_k is the system noise vector $(n \times 1)$ given by an n dimensional Wiener process of known

covariance C.

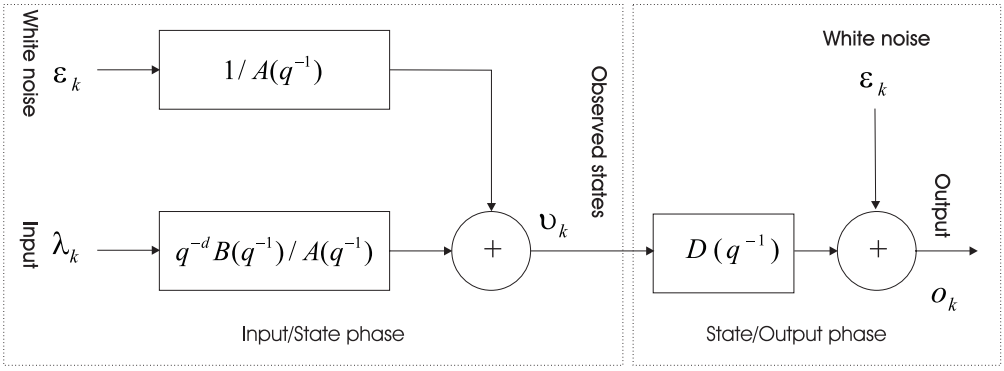


Figure 6.1: Standard phases of Kalman Filter

Note that many descriptions of the Kalman filter do not include the control vector λ_k. The control vector is included here in anticipation of the application to the robot kinematics models. The process $f(\upsilon, \lambda, \epsilon)$ may not be directly observable. This means that the state υ cannot be completely determined from measurements. To account for this, an observation model $f(\upsilon, \epsilon)$ is introduced. The observation or measurement model is a mapping of the actual system state υ to an observed state o. The observation model is given by the discrete time linear difference equation

$$o_k = f(\upsilon_k, \epsilon_k) = D_k \upsilon_k + \epsilon_k \tag{6.3}$$

where o_k is the observed state vector at time step k and $o \in R^m$, D_k is the $(n \times n)$ measurement matrix mapping υ_k to o_k at time step k, ϵ_k is the measurement noise vector given by n dimensional Wiener process of known covariance C. Note that equations (6.2) and (6.3) constitute the normal state space formulation of a linear system used extensively in control theory. The system and measurement noise are assumed to be unbiased normally distributed random processes

$$p(\epsilon) \sim N(0, \mathbf{C}) \tag{6.4}$$

The covariance matrix of ϵ is given by

$$E\left[\epsilon_k \epsilon_i^T\right] = \begin{cases} \mathbf{R}_k & , i = k \\ 0 & , i \neq k \end{cases} \tag{6.5}$$

where $E[\upsilon] = \sum_{i=1}^{n} p_i \upsilon_i$ denotes the mathematical expectation of the discrete random variable υ where p_i is the probability of the occurrence event υ_i. It is

assumed that the ϵ are independent random processes and hence have zero cross-correlation

$$E\left[\epsilon_k \epsilon_i^T\right] = 0, \ \forall i, k \tag{6.6}$$

Given the above definitions, the filtering problem can be defined as arriving at an optimal estimate of the state υ_k conditioned by the measurement o_k. We wish to implement a recursive linear estimator which provides the optimal, in the least squares sense, estimate of system state υ_{k+1} given the measurement o_k and the current state υ_k.

Transparency Condition: for simplicity, we considered $D(q^{-1})$ unity, consequently the second phase of Kalman filter will be transparent and both the output o_k and the observed state υ_k will be equal. This form of modeling is called auto regressive exogenous (ARX), which will be presented next.

6.2.5 ARX Modeling of Robot Dynamics

The discrete ARX modeling paradigm of robot system dynamics is derived from Kalman filter as explained in the preceding section. The sequentially recursive structure of this paradigm is shown in figure (6.2). This structure takes into account both the observed state υ_k and the reference control signal λ_k which is given by:

$$\upsilon_k = \sum_{i=1}^{na} a_i \upsilon_{k-i} + \sum_{i=1}^{n_b} b_i \lambda_{k-i} + \epsilon_k \tag{6.7}$$

Where ϵ_k is a modeling residual, Gaussian white noise, n_a is the model order of the observed state (also called the number of poles), while n_b is the model order of the control signal (also called the number of zeros). The operator q^{-1} is the back shift operator or delay, where $q^{-1}\,\upsilon_k = \upsilon_{k-1}$, as follows:

$$\begin{aligned} A(q^{-1})\upsilon_k &= B(q^{-1})\lambda_{k-1} + \epsilon_k \\ A(q^{-1}) &= 1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_{n_a} q^{-n_a} \\ B(q^{-1}) &= b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{n_b} q^{-n_b} \end{aligned} \tag{6.8}$$

In this case the observed state υ_k is the robot's longitudinal velocity, heading and heading speed. The reference control signal λ_k is the guide's range, acquired from distributed robot sensors (vision system, laser scanner and sonar). The Gaussian distributed noise, associated with the observed output, enables applying weights (coefficients) identification algorithms such as; maximum likelihood (ML), least mean square (LMS) and recursive least squares (RLS). The ARX paradigm is applicable within linear and quasi-linear systems. Therefore, we applied it to

underlie the navigation control, while it failed to cope with position control due to the presence of large non-linear odometric errors [6].

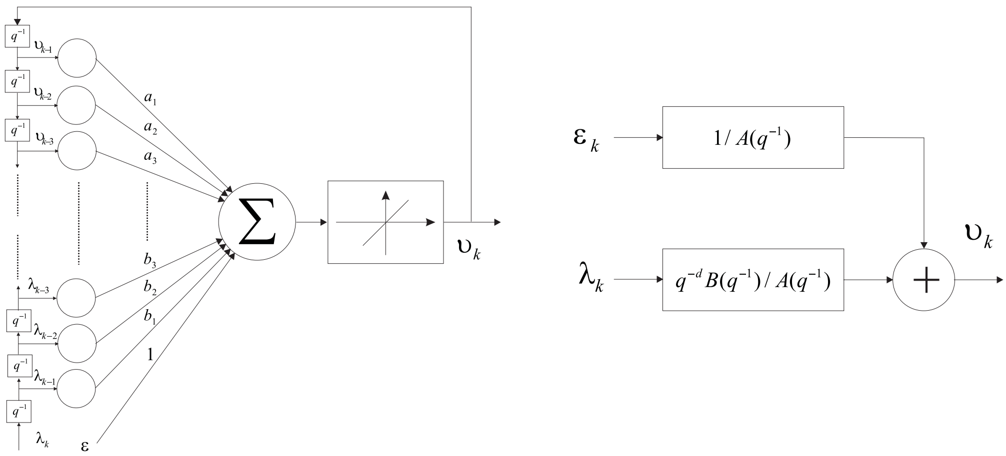


Figure 6.2: ARX modeling paradigm of mobile robot dynamics

6.2.6 Matrix Inversion Lemma (MIL) Learning Rule

In general, the weights estimation problem of the ARX paradigm can be treated as a linear regression problem. The linear regression is the simplest type of parametric models. Its origin can be traced back to Gauss (1809), who used such a technique, equation (6.9), for calculating orbits of the planets.

Now let us explain, how to teach the ARX model to deduce its weights Θ. The MIL is a stepwise learning algorithm, which means that the estimation of ARX's weights yields a gradual convergence. Compared with the LMS and ML algorithms, the MIL needs lower computation and it is reliable to be applied on-line.

For weights estimation purposes, it is convenient to write the ARX paradigm in a form, which emphasizes the weights vector to be estimated and the data available. This is achieved by using the backward shift interpolation of (q^{-1}) to cast the ARX model in the form

$$\upsilon_k = \phi_k^T \Theta + \epsilon_k \tag{6.9}$$

Where Θ is the vector of unknown weights, defined by:

$$\Theta^T \quad = [-a_1, \dots ,\ -a_{n_a},\ b_1,\ \dots ,b_{n_b}] \tag{6.10}$$

and ϕ_k is a regression vector partly consisting of measured input/ouput variables and defined by:

$$\phi_k^T = [\upsilon_{k-1}, \dots, \upsilon_{k-n_a}, \lambda_{k-1}, \dots, \lambda_{k-n_b}] \tag{6.11}$$

LMS: The least squares estimator for the weights vector $\widehat{\Theta}$ is directly calculated as follows:

Batchwise: The LMS algorithm can be employed to estimate model parameters of mobile robot dynamics. Moreover, the LMS is a successful method for tracking of dynamic targets such as airborne/space systems and offshore/onshore vehicles. We can describe the output of the robot (υ_{k+1}) as a time series, which is a function of the past output readings (υ_k), ranging data acquired from laser, geometrical vision, and sonar (λ_k) and Gaussian noise ϵ, see model (6.9).

If the system is unknown, however, then the polynomial coefficients are treated as parameters to be determined by measurement or estimation. For estimation purposes, it is convenient to write equation (6.9) in a form, which emphasizes the object to be estimated and the data available. This is achieved by using the backward shift interpolation of (q^{-1}).

Now we assume that the preceding equations are an exact description of the robot dynamics, i.e. it is the true data-generating mechanism, and that we wish to determine from available data the vector (θ) of true system parameters. In order to do this we further assume a model of the system of the correct structure:

$$\upsilon_k = \phi_k^T \widehat{\theta} + \widehat{\epsilon}_k \tag{6.12}$$

where $\widehat{\theta}$ is a vector of adjustable model parameters and ϵ_k is the corresponding modeling or fitting error at time step k. Our aim is to select $\widehat{\theta}$ so that overall modeling error is minimized in some sense. ϵ_k is defined by

$$\widehat{\epsilon}_k = \epsilon_k + \phi_k^T(\theta - \widehat{\theta}) \tag{6.13}$$

so that $\widehat{\epsilon}_k$ depends on $\widehat{\theta}$ and, in some cases, the minimized modeling errors will be equal to the white noise sequence corrupting the system output data. Assume that the system described by equations (6.9 - 6.11) has been running for sufficient time to form n conductive data

vector. The data obtained in this way allows the model (6.12) to be expressed in the vector/matrix form:

$$\begin{bmatrix} \upsilon(1) \\ \upsilon(2) \\ . \\ . \\ . \\ \upsilon(n) \end{bmatrix} = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ . \\ . \\ . \\ \phi^T(n) \end{bmatrix} \widehat{\theta} + \begin{bmatrix} \epsilon(1) \\ \epsilon(2) \\ . \\ . \\ . \\ \epsilon(n) \end{bmatrix} \tag{6.14}$$

To be able to estimate the parameters uniquely the number n of equations (6.14) must not be less than m, the number of unknown parameters in the vector (θ). In the noise-free case (e_k=0), the equations can be solved as a set of linear equations in $n = m$ unknowns, where $m = n_a + n_b + 1$. The resulting modeling errors are identically zero. When noise is present (and, in practice, even in normally noise-free systems) we must have n much larger than m and use an alternative procedure to reduce estimation errors induced by the noise. The technique most widely used in this connection is linear least squares, which we now introduce.

Rewrite equation (6.14) in the stacked notation

$$\Upsilon = \Phi\widehat{\theta} + \widehat{\epsilon} \tag{6.15}$$

in which

$$\Upsilon = \begin{bmatrix} \upsilon(1) \\ \upsilon(2) \\ . \\ . \\ . \\ \upsilon(n) \end{bmatrix}, \quad \widehat{\epsilon} = \begin{bmatrix} \widehat{\epsilon}(1) \\ \widehat{\epsilon}(2) \\ . \\ . \\ . \\ \widehat{\epsilon}(n) \end{bmatrix} \text{ and } \Phi = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ . \\ . \\ . \\ \phi^T(n) \end{bmatrix} \tag{6.16}$$

Rearrange (6.15) in terms of the error vector $\widehat{\epsilon}$:

$$\widehat{\epsilon} = \Upsilon - \Phi\widehat{\theta} \tag{6.17}$$

and select an estimate $\widehat{\theta}$ of the true vector of parameters which minimizes J, the sum of squares errors:

$$J = \sum_{i=1}^{n} \widehat{\epsilon}_i^2 = \widehat{\epsilon}^T\widehat{\epsilon} \tag{6.18}$$

To find the least squares estimate, rewrite equation (6.18) in terms of the data vector and the parameter vector:

$$\begin{aligned} J &= (\Upsilon - \Phi\widehat{\theta})^T(\Upsilon - \Phi\widehat{\theta}) \\ &= \Upsilon^T\Upsilon - \widehat{\theta}^T\Phi^T\Upsilon - \Upsilon^T\Phi\widehat{\theta} + \widehat{\theta}^T\Phi^T\Phi\widehat{\theta} \end{aligned} \tag{6.19}$$

Setting the derivative of J with respect to $\widehat{\theta}$ for a stationary point to zero

$$\frac{\partial J}{\partial\widehat{\theta}} = -2\Phi^T\Upsilon + 2\Phi^T\Phi\widehat{\theta} = 0 \tag{6.20}$$

yields the normal equations:

$$\Phi^T\Phi\widehat{\theta} = \Phi^T\Upsilon \tag{6.21}$$

and these solve for a unique minimum if the second derivative matrix

$$\frac{\partial J^2}{\partial\widehat{\theta}^2} = 2(\Phi^T\Phi) \tag{6.22}$$

is positive definite. Hence the least squares estimator for the parameter vector is

$$\widehat{\theta} = [\Phi^T\Phi]^{-1}[\Phi^T\Upsilon] \tag{6.23}$$

The resulting modeling error $\widehat{\epsilon}$ is denoted by

$$\widehat{\epsilon}^T = [\widehat{\epsilon}_1, ..., \widehat{\epsilon}_n] \tag{6.24}$$

whose components are called residuals.

LMS has several attractive features for the purposes of tracking and identification. First, large errors are heavily penalized. Second, least squares estimates can be obtained by straightforward matrix algebra. Third, the least squares criterion is related to statistical variance and the properties of the solution can be analyzed according to statistical criteria. On the other side, the LMS algorithm depends on a direct matrix inversion, so applying this technique to some models seems critical. Moreover, its estimates approach an optimal solution as the number of weights reaches infinity.

Stepwise: The Widrow-Hoff learning rule (1967) is widely used for static pattern classification purposes of linear regression models. It depends mainly on a stepwise change of the weights to learn the model. This rule has a limited precision and has a limited processing range. If υ_i is the desired output and α is the learning speed constant, then the adjustment of the weight $\theta_{i,j}$ linking input i with output j is accessed as follows:

$$\Delta\theta_{i,j} = \alpha(\upsilon_i - \Phi^T\Theta)\phi_j, \quad for \;\; j = 1, 2, \ldots, n \tag{6.25}$$

ML: The ML method is one of the few generally applicable methods for parameter estimation. The basic idea is to construct a function of the data and the unknown parameters called likelihood function. The estimate is then obtained as the parameter value which maximizes the function. The likelihood function is essentially the probability density function of the observations. Thus maximum likelihood identification means that we select the estimate which renders the given observations most probable.

Suppose that the observations are represented by a random vector variable $\Upsilon = [\upsilon_1, \upsilon_2, \ldots, \upsilon_n]^T$ and denote the probability density function

$$f(\theta; \upsilon_1, \upsilon_2, \ldots, \upsilon_n) = f_\upsilon(\theta; \Upsilon) \tag{6.26}$$

Here the unknown parameter vector θ that describes the properties of the observed variable will be estimated using the observations of Υ. Then the likelihood function for the observed values can be expressed as

$$L(\theta) = f_\upsilon(\theta; \Upsilon) \tag{6.27}$$

The likelihood function is a deterministic function of the parameter vector θ once the numerical value of Υ is inserted; in contrast the probability density function is function of random variable Υ for fixed parameter vector θ. The estimate can then be given as

$$\max_\theta L(\theta) = L(\hat{\theta}) \tag{6.28}$$

The principle of the maximum likelihood method is simple and it can be shown that it gives asymptotically (when $n \to \infty$) unbiased and efficient (minimum variance) estimates. Let us consider the system described by equation (6.15). And assume that ϵ is normally distributed and it has n

elements, furthermore its mean $E[\epsilon] = 0$ and its covariance $E\left[\epsilon\,\epsilon^T\right] = C_\epsilon$ are known. It is equally valid to use the probability density function of the noise since there is a one-to-one transformation between Υ and ϵ. Then the corresponding multivariable Gaussian probability density function of ϵ takes the following form

$$f(\epsilon) = [(2\pi)^n \det(C_\epsilon)]^{-1/2} \exp(-\frac{1}{2}\epsilon^T\, C_\epsilon^{-1}\, \epsilon) \tag{6.29}$$

Under the assumption of equation (6.15) we get the likelihood function

$$f(\epsilon) = [(2\pi)^n \det(C_\epsilon)]^{-1/2} \exp(-\frac{1}{2}\left[\Upsilon - \Phi\ \theta\right]^T\ C_\epsilon^{-1}\ \left[\Upsilon - \Phi\ \theta\right]) \tag{6.30}$$

We may consider equation (6.30) as a function of the parameter vector θ. It is practical to take the logarithm of the likelihood function

$$\log L(\theta) = -\frac{1}{2}\log\left[(2\pi)^n \det(C_\epsilon)\right] - \frac{1}{2}\left[\Upsilon - \Phi\ \theta\right]^T\ C_\epsilon^{-1}\ \left[\Upsilon - \Phi\ \theta\right] \tag{6.31}$$

Then at the maximum of the log-likelihood function we obtain the estimate $\hat{\theta} = \theta$. In the case when the noise is a normally distributed white noise with a covariance matrix $C_\epsilon = \sigma^2 I$, it is easy to show that the maximization of the likelihood function (equation (6.31)) is equivalent to the minimization of the loss function

$$Var(\theta) = \frac{1}{2}\left[\Upsilon - \Phi\theta\right]^T\left[\Upsilon - \Phi\theta\right] = -\sigma^2 \log L(\theta) + \text{constant} \tag{6.32}$$

If σ is unknown, maximization of equation (6.31) with respect to the parameters and σ can be done separately. First minimize equation (6.32) with respect to θ. The maximum of equation (6.31) with respect to σ can be given in the following form

$$\hat{\sigma}^2 = \frac{2}{n}\min\ Var(\theta) \tag{6.33}$$

The problem of maximizing equation (6.31) can be solved using suitable numerical tools, like the Newton-Raphson method.

The ML is applied to a system with noise which is described in the form of an autoregressive exogenous model defined by equation (6.7). Then, in case of normally distributed noise the log-likelihood function can be given as:

$$\log L(\theta) = -\frac{1}{2}\log\left[(2\pi)^n \det(C_\epsilon)\right] - \frac{1}{2}\varepsilon^T C_\epsilon^{-1}\,\varepsilon \tag{6.34}$$

The problem is that $\epsilon_{k-1}, \ldots, \epsilon_{k-n}$ are not known. It is therefore only feasible to make approximate solutions by finding successively better estimates of the covariance matrix C_ϵ and the parameters through an iterative procedure.

MIL: Before going to introduce the MIL algorithm let us explore some common properties of LMS and ML techniques [151, 92, 89, 211, 188, 150, 45].

1. The LMS and ML estimate of weights approaches the optimum Wiener solution as the data length n approaches infinity, if the input and the desired response are jointly stationary ergodic processes.
2. The LMS estimate of the weights vector is unbiased if the error signal ϵ_k has zero mean for all k.
3. The computation of both LMS and ML algorithms is relatively high and that is cumbersome especially if we apply them to real-time processes.
4. Both LMS and ML are slow learning algorithms and that does not suit adaptive systems.

Let A and B be two positive definite, m by m matrices related by

$$A = B^{-1} + CD^{-1}C^T \tag{6.35}$$

where D is another positive definite, n by n matrix and C is an m by n matrix. According to the matrix inversion lemma, we may express the inverse of the matrix A as follows:

$$A^{-1} = B - BC\left[D + C^T BC\right]^{-1} C^T B \tag{6.36}$$

special case: if $n = 1$ and for simplicity let $D = 1$, then we get:

$$A^{-1} = B\left[I_m - \frac{CC^T B}{1 + C^T BC}\right] \tag{6.37}$$

The MIL is widely used as an estimator in different machine learning algorithms such as Kalman filter, ANFIS fuzzy, recursive least squares (RLS) etc. We will employ this rule to estimate the weights of the ARX paradigm which servers as a model of mobile robot dynamics. The estimation process can be accomplished according to the following:

At time step k:

1. Form Φ_k using the new data (input-output patterns)

$$\begin{aligned} \Phi_k^T &= [\upsilon_{k-1},\ \ldots\ , \upsilon_{k-n_a},\ \lambda_{k-1},\ \ldots\ , \lambda_{k-n_b}] \\ &= [\phi_1,\ \phi_2,\ \ldots\ \phi_{n_a+n_b}] \end{aligned} \tag{6.38}$$

2. Calculate the estimation error ϵ_k using

$$\epsilon_k = \upsilon_k - \Phi_k^T \widehat{\Theta}_{k-1} \tag{6.39}$$

3. Apply the MIL rule as in equation (6.37) to calculate the covariance matrix Ψ_k, sometimes called Kalman gain based update matrix. So if $A^{-1} = \Psi_k$, $B = \Psi_{k-1}$, $D = 1$ and $C = \Phi_k$, then the MIL will be

$$\Psi_k = \Psi_{k-1}\left[I_m - \frac{\Phi_k \Phi_k^T \Psi_{k-1}}{(1 + \Phi_k^T \Psi_{k-1} \Phi_k)}\right] \tag{6.40}$$

4. Update weights

$$\widehat{\Theta}_k = \widehat{\Theta}_{k-1} + \Psi_k \Phi_k \epsilon_{k-1} \tag{6.41}$$

where

$$\begin{aligned} \widehat{\Theta}_k^T &= [-\widehat{a}_1,\ \ldots\ , -\widehat{a}_{n_a},\ \widehat{b}_1,\ \ldots\ , \widehat{b}_{n_b}] \\ &= [\widehat{\theta}_1,\ \widehat{\theta}_2,\ \ldots\ \widehat{\theta}_{n_a+n_b}] \end{aligned} \tag{6.42}$$

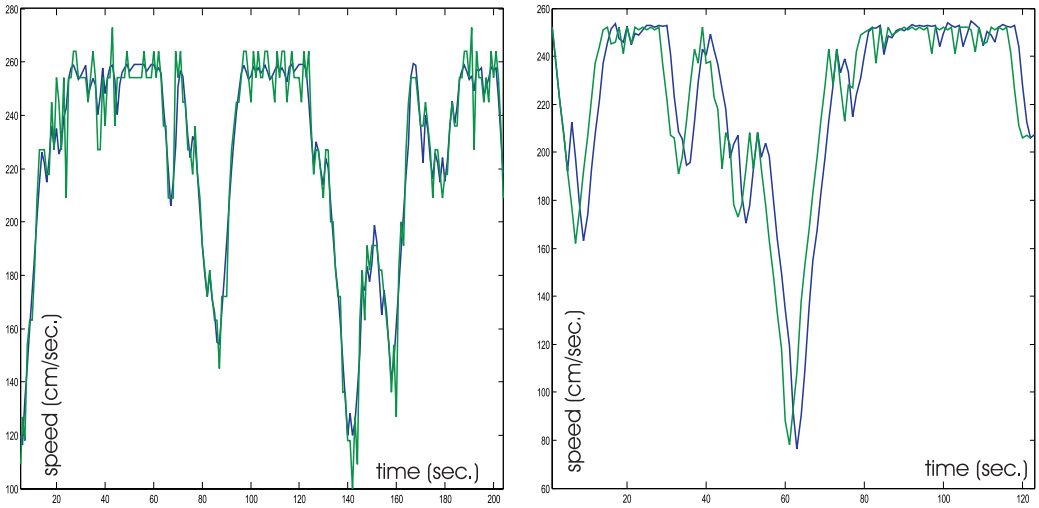
5. Wait for the next time step to elapse and loop back to step (1)

MIL parameters have been initialized using an empty vector. Figure (6.3.a) shows the output of the ARX model w.r.t. the actual output of the robot, while figure (6.3.b) shows the the capability of the model to predict the output if the system input is known. Figure (6.4) shows the convergence of model weights and the error during the learning phase. The output of the ARX model seems smooth due to

the filtering of high frequencies.

The most important advantages achieved from applying MIL are:

- parameter estimates converge quickly, allowing fast adaptation under unknown and changing conditions.
- it requires relatively small computational effort, which is crucial for real-time control of robots, which require high sample rates.
- the high signal-to-noise ratios typical of robotic applications enable MIL to maintain high quality parameter estimates. The other methods outperform MIL when there is a low signal-to-noise ratio.



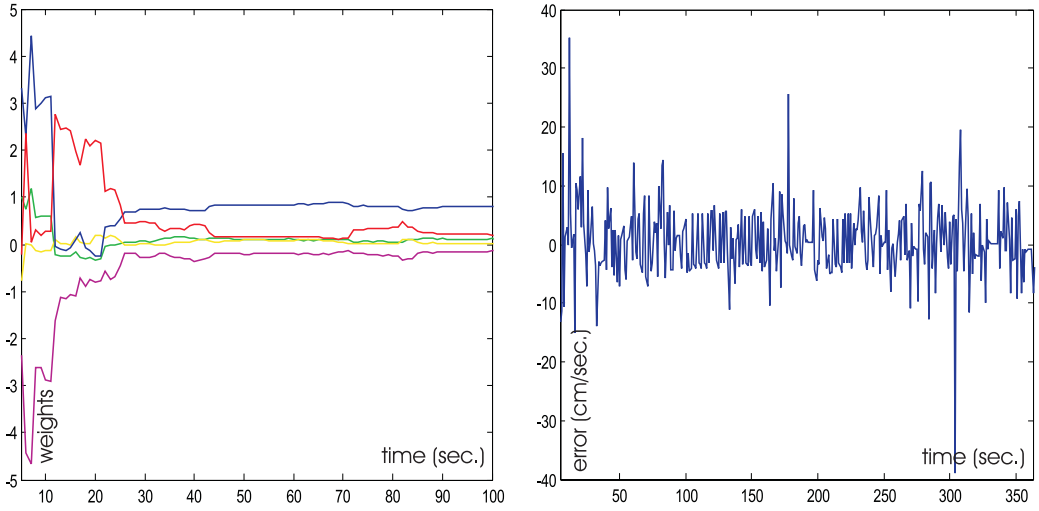
(a) Model output w.r.t. system output (b) Model in prediction phase

Figure 6.3: ARX model performance in learning and prediction phases

ARX Models have the capability to model mobile robot dynamics and predict their behavior if the input (i.e. the obstacles histogram or the visual ranging entries) is known, see figure(6.3).

6.3 Visual Guide Extraction

To adapt the navigation of a robot we have to close the loop between the sensors and actuators. The direct feedback of ranging sensors such as sonar, IR and laser is simple to be incorporated in a control system. The most difficult problem in



(a) Model weights change (b) Model learning error

Figure 6.4: ARX model weights and learning error

control system is the coupling of indirect sensors such as vision system, where there is no definite scalar feedback. Our goal in the section is to describe how we could use the vision system as a direct scalar sensor. This means that, the vision system serves to extract the geometrical features of a specific object (guide).

To resemble human behavior in object tracking, we have to close the loop between the binocular vision system and the robot's actuators. The main goal of the vision system is to extract the object (guide) and to estimate its size and range geometrically, see figure (6.8).

The process of visual guide extraction includes the following steps: **1)** select the appropriate color model, **2)** define the color of the guide, **3)** apply a Gaussian filter to smooth the image, **4)** filter the object, **5)** deduce the x and y projections of the guide's center of gravity COG.

6.3.1 Selection of an Appropriate Color Model

The Universal Color Language classifies colors into color spaces. These three-dimensional coordinate systems quantitatively define colors. Although many of defined color spaces exist, RGB and HSV are the most commonly used models. The RGB color space represents all colors as a mixture of red, green and blue, which constitute the primary colors used by video cameras, televisions and PC monitors. When combined, these colors can create any color on the spectrum.

The HSV color space is spanned by axes into hue, saturation and intensity. Hue refers to pure color, saturation refers to the degree or color contrast, and intensity refers to color brightness. Modeled on how human beings perceive color, this color space is considered more intuitive than RGB.

To analyze and process images in color, robot vision systems typically use data from either the RGB or HSI color spaces, depending on a given task's complexity. For example, in simple applications such as tracking highly saturated homogeneous colored guides it is enough to use the RGB model, see figures 6.5.c and 6.5.d. With more complex applications, however, such as persons tracking based on the skin color under changing illumination, a vision system may require hue, saturation and intensity information to perform the operation, see figures 6.5.a 6.5.b.

HSV color model

This model is sometimes referred to as HSV instead of HSI. The main advantages of this model are that chrominance (H, S) and luminance (V) components are decoupled. In short, the RGB model is suited for image color generation, whereas the HSV model is suited for image color description. It is related to the RGB model as follows:

$$V = \tfrac{1}{3}(R + G + B) \tag{6.43}$$

$$S = 1 - \frac{3[\min(R, G, B)]}{(R + G + B)} \tag{6.44}$$

$$H = \begin{cases} \theta & B \leq G \\ 360 - \theta & B > G \end{cases} \tag{6.45}$$

$$\theta = \cos^{-1}\left\{ \frac{0.5(2R - G - B)}{\sqrt[2]{(R - G)^2 + (R - B)(G - B)}} \right\} \tag{6.46}$$

Taking a look at equations (6.43 - 6.46) we notice that this model is critical when $R = G = B$, it returns no definite solution. This is one of the weaknesses of the HSV color model, another one is the computation time required to estimate the model which can be eliminated using high speed cameras provided with special chips to get HSV components as well as RGB.

a) RGB based face tracking under two different illumenations

b) HSV based face tracking under two different illuminations

c) RGB based object tracking under two different illuminations

d) HSV based object tracking under two different illuminations

Figure 6.5: RGB and HSV based tracking

Gaussian Filter

To reduce the high frequency noise we used a Gaussian filter which is powerful not only in noise reduction but also in edge detection. Figure (6.6) shows the 3D Gaussian distribution. The idea of Gaussian smoothing is to use this 2D distribution as a point-spread function, and this is achieved by convolution. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function before we can perform the convolution. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero more than about three standard deviations from the mean, and so we can truncate the kernel at this point, for simplicity let $\sigma = 1.0$.

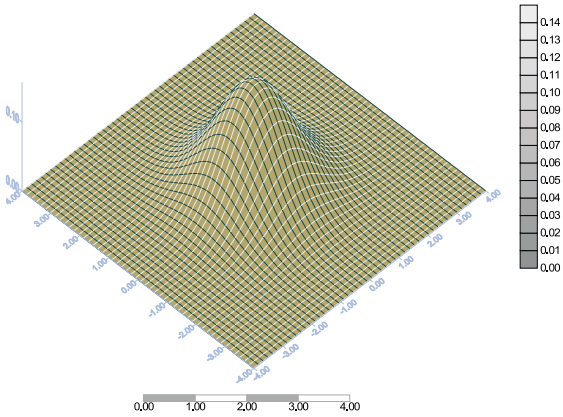


Figure 6.6: Gaussian filtering of high frequencies

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{6.47}$$

6.3.2 Deduction of Guide's COG

To track a guide dynamically its center of gravity is calculated and the results are delivered to the control system to steer the robot adaptively, see figure (6.7).

Compared with contour detection presented in [61, 62, 63], the COG is more accurate because the object to be tracked is represented with a single point in 3D dimensions. Moreover, the elapsed time required to extract the geometrical LUT of the object is shorter. This representation is easy to be manipulated in control system as a definite feedback. The COG is calculated according to the following:

$$x_c = \frac{\sum_{i=1}^{n} x_i}{n} \quad and \quad y_c = \frac{\sum_{i=1}^{n} y_i}{n} \tag{6.48}$$

6.3.3 Geometrical Calibration

According to figure (6.8) we can deduce the distance between the robot and the guide mathematically using the following form:

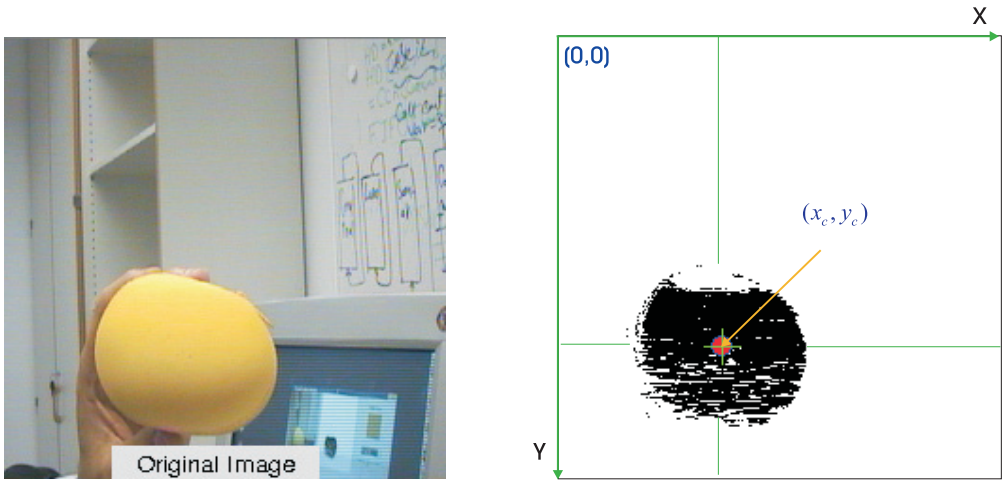


Figure 6.7: Center of gravity

$$n = e^{-k\Lambda} \tag{6.49}$$

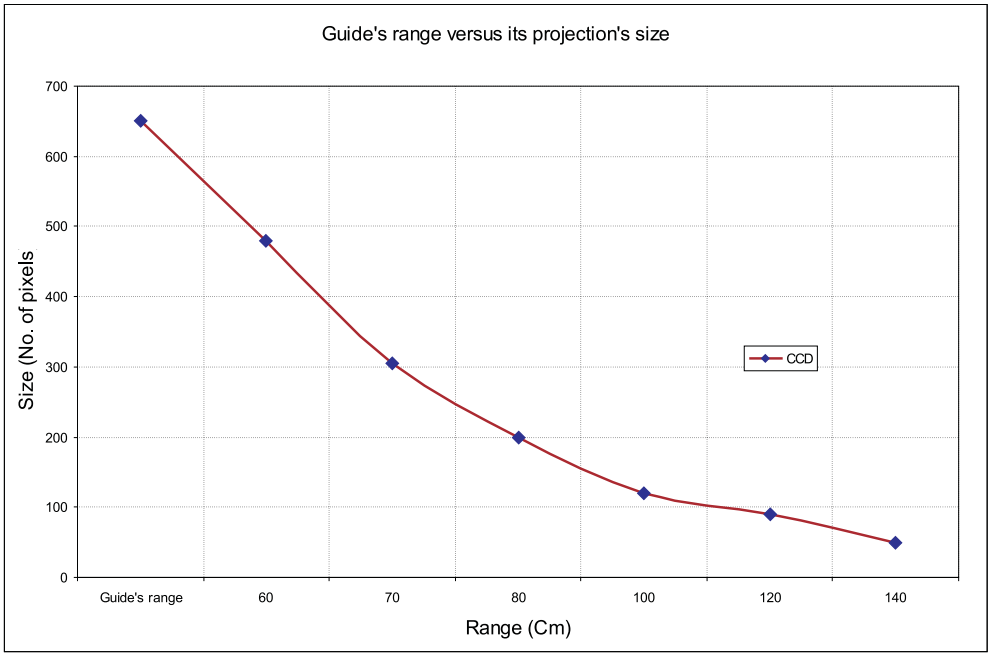


Figure 6.8: Range calibration of visual guides

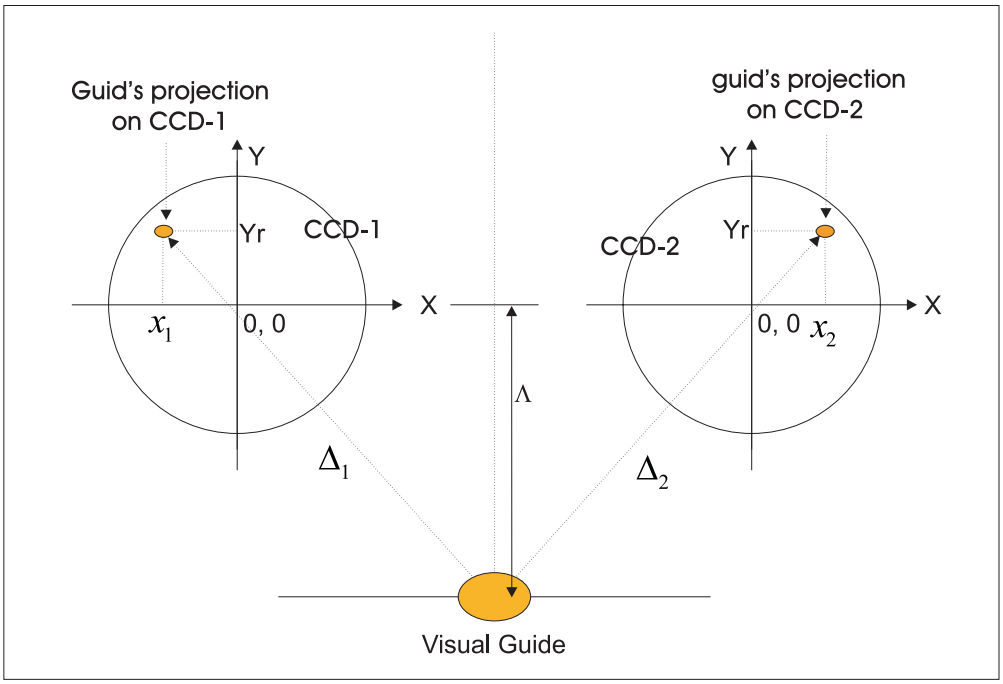


Figure 6.9: Principles of geometrical vision

whereas Λ is the distance between the robot and the guide, k is a constant depending on the geometry of the camera's lens and n is the projection size. To reject the noise a threshold number is used to delimit the projection size.

From the size of the object n we estimate the distance Λ to the guide while the horizontal projections x_1 and x_2 indicate the heading angle to the guide. To calculate the pose of the guide in 3D dimensions geometrically, first we use the COG rule to get x and y axes parameters x_1, x_2, y_1, y_2. Then, we use the range calibration method to calculate the z axis parameter, distance Λ.

6.4 Sensor Integration

To resemble the human behavior in control and adaptation, we have to close the loop between robot's distributed sensors and its actuators. The main goal of that is to steer the robot adaptively while chasing a dynamic object or to navigate safely in unknown terrain without collision, based on multi-sensor integration.

Figure (6.10) shows a scheme of sensor integration. The binocular vision sys-

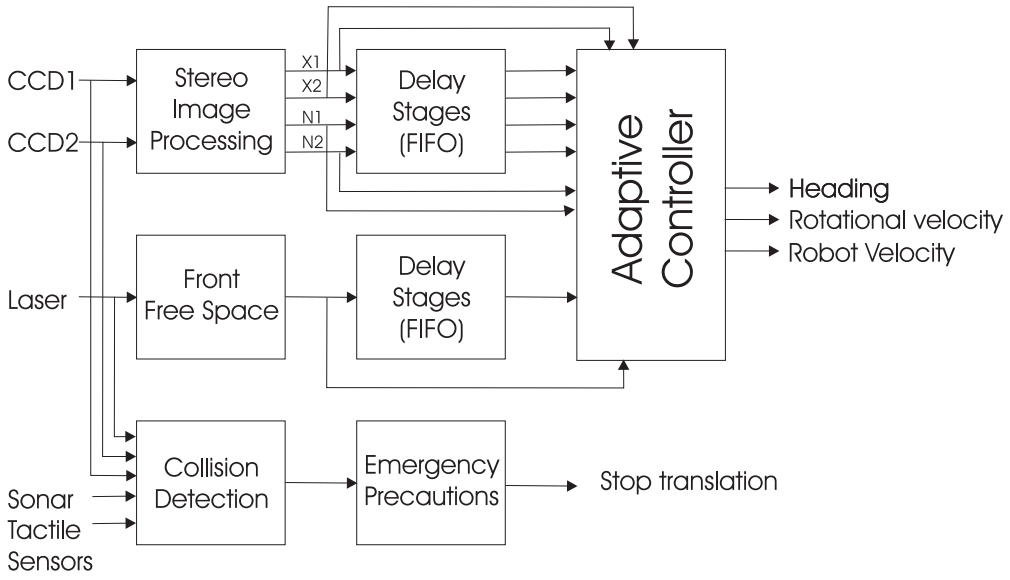


Figure 6.10: Multi-sensor integration for adaptive steering of mobile robots

tem uses the projection of a colored object on two CCD cameras to deduce its horizontal components of the center of gravity x_1 and x_2 with the number of projection pixels n_1 and n_2 [2]. The obstacles histogram, acquired from the laser scanner, is used to control the robot velocity, heading and heading speed. The multiplexing unit is a fuzzy membership function which interpolates the control signal from references (visual and laser ranging modes).

The strategy is to build an integrated system, which enables the robot to pursue a dynamic object using a vision system. Moreover, other distributed sensors collaborate to support the robustness of the overall system. The control of the robot's navigation (heading and translation) relies primarily upon outputs of both the laser range finder and the binocular vision system. In emergencies, all associated distributed sensors such as vision, laser, sonar, tactile and button sensors collaborate to prevent collisions, see figure (6.10).

6.5 Adaptive Control

The papers of Wellstead [211] and Åstrom [188] were among the first to exploit the idea of a self-tuning pole placement controller, though the concept of a self-tuning system was first proposed by Kalman in the late 1950s. The theory developed here follows these original formulations, and also includes additional concepts relevant

to practical implementation.

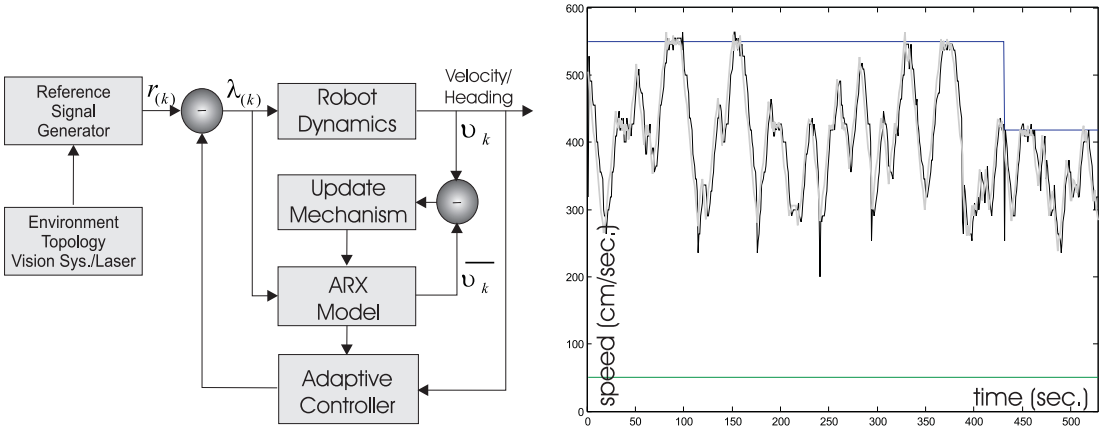
Self-tuning pole placement controllers work on the principle of continually adjusting the controller gains, such that the closed loop system poles remain constant, irrespective of changes in the system's dynamics. The controller has three components; **1)** a low-order linear model of the robot under control, **2)** an on-line system identifier and **3)** a pole placement controller. Figure (6.11.a) shows these fundamental components.

Adaptive control, referred to as self-tuning control, utilizes a low order linear approximation of the robot model, whose parameters are estimated on-line from past input and output values using the MIL system identification algorithm. The model structure used can be either single input single output (SISO) or multi input multi output (MIMO), and have joint space or Cartesian space outputs, as appropriate. The use of a linear autoregressive model allows the use of efficient recursive identification algorithms, such as LMS, MIL, ML, etc.. The controller parameters are then designed based on this linear model, so that the closed loop system meets some prescribed performance criterion. Any variations in the dynamics of the system will be tracked by the identification algorithm, and hence automatically accommodated by the controller.

6.5.1 Adaptive Pole Assignment

The use of pole assignment controller (PAC) controllers can be used when the delays are present, which can be problematic for fixed gain controllers. PAC schemes work by automatically adjusting the gains so that the poles of the closed loop system are placed at some specified location. Consequently, the system response remains constant irrespective of changes in the underlying system. This type of self-tuning controller has found several applications of robot control compared with fixed gain controllers, the main reason for this being the significantly higher flexibility and precision. Moreover, PAC is more intuitive for the designer as it resembles a classical control design method, rather than using weighting variables which require careful selection and do not have a clear physical meaning. Pole placement controllers also yield smoother control signals.

The structure of the ARX paradigm is the most reliable form to implement adaptive control systems. Figure (6.11.a) shows the structure of an adaptive control system using linear regressive nets, while (6.11.b) shows how the system output follows the reference signal precisely.



(a) The structure of an adaptive control system

(b) System response w.r.t. reference signal

Figure 6.11: Adaptive control of autonomous mobile robots

To achieve the adaptability of the system under study, the weights of the controller can be calculated according to the following rules.

$$\frac{\upsilon_k}{r_k} = q^{-d}.\frac{B(q^{-1}).H_c}{T(q^{-1})} \;\; where \;\; H_c = \lim_{q\to 1}\frac{T(q^{-1})}{B(q^{-1})}, \;\; and \;\; d = 1 \tag{6.50}$$

$$F(q^{-1})A(q^{-1}) + B(q^{-1})G(q^{-1}) = T(q^{-1}) \tag{6.51}$$

$$\begin{aligned} T = &\;\; 1 + t_1 q^{-1} + \ldots + t_{n_t} q^{-n_t} \\ G = &\;\; g_0 + g_1 q^{-1} + \ldots + g_{n_g} q^{-n_g} \\ F = &\;\; 1 + f_1 q^{-1} + \ldots + f_{n_f} q^{-n_f} \end{aligned} \tag{6.52}$$

where, $n_f = n_b, \;\; n_g = n_a - 1, \;\; and \;\; n_t \leq n_a + n_b$.

Integrating Fuzzy Systems under Kalman Filters

Sensor measurements are accompanied with uncertainties and noise. Therefore, applying a Kalman filter directly to such readings is fraught with complications. Obeying the rule used in (3.4) page (41), the linguistic treatment of Kalman filter inputs can solve this problem simply. Instead of the traditional form of the Kalman filter equation (6.2) page (90), it is as follows:

$$\upsilon_{k+1} = f\left(\upsilon_k, \lambda_k, \epsilon_k\right) = A_k \upsilon_k + B_k \mu(\lambda_k) + \epsilon_k \tag{6.53}$$

Whereas μ is the union (AND) fuzzy logic operation, equation (6.54), applied to generate the control signal λ_k as an interpolation between visual and laser

reference signals. This operation selects the minimum value by which the system attains more safety, see figure (6.12).

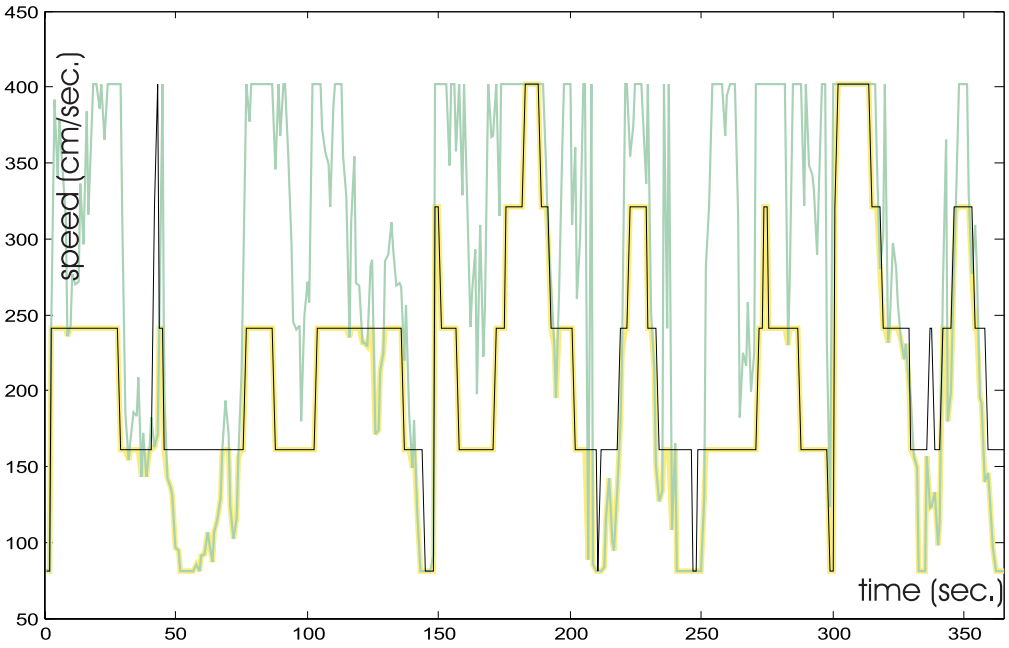


Figure 6.12: Membership function of multi references

$$\lambda_k = \mu_{v \cap l} = min(\mu_v, \ \mu_l) \tag{6.54}$$

The controller comprises three parts; the first one is the compensator (H), feedback filter G and the preconditioning unit $1/F(q^{-1})$. The design of the PAC controller is shown in figure (6.13).

$T(q^{-1})$ is the selected poles function, the main job of this function is to preserve the overall stability of the system. The compensator H corrects the offset error of the output, while the $G(q^{-1})$ and $F(q^{-1})$ components are the controller poles and zeros that adapt the performance. The controller parameters can be calculated from equation (6.51).

The F and G parameters of the adaptive controller can be calculated using equation 6.51. If we consider $n_a = n_b = 2$ then $n_f = 2$ and $n_g = 1$. Multiplying out and equating coefficients of q^{-i} for $i = 1, 2, \ldots, n$ leads to the following equation set:

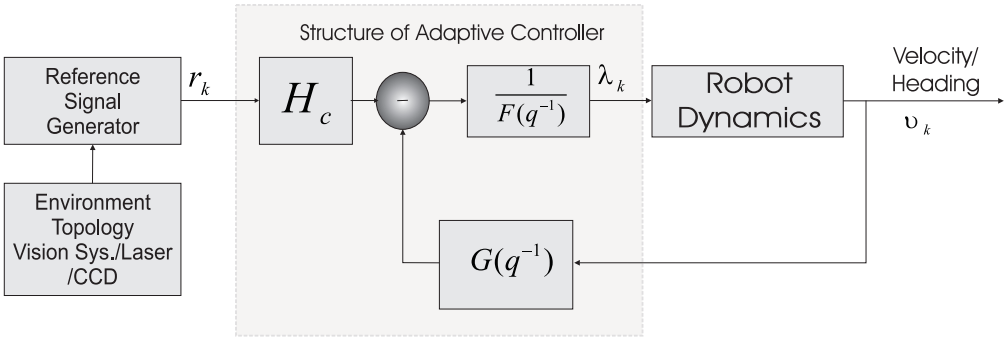


Figure 6.13: The structure of the PAC controller

$$\begin{bmatrix} 1 & 0 & b_0 & 0 \\ a_1 & 1 & b_1 & b_0 \\ a_2 & a_1 & b_2 & b_1 \\ 0 & a_2 & 0 & b_2 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ g_0 \\ g_1 \end{bmatrix} = \begin{bmatrix} t_1 - a_1 \\ -a_2 \\ 0 \\ 0 \end{bmatrix} \tag{6.55}$$

The estimated controller parameters f_1, f_2, g_0 and g_1 are shown in figure(6.14). The selection of the closed loop poles is the most important factor in quality measuremt. As the assigned poles tend to reach the zero of the Z-domain or -∞ of the Laplace S-domain, the cross correlation, as in section (3.3.2) page (29), between the reference signal (Laser histogram, visual ranging or sonar ranging in our case) and the response reaches unity. In general, that is not too cheap to consume the system resources and inputs to attain the required quality. The cost function of a system plays an important role in the definition of task priorities based on multi-criteria optimization. Figure (6.15) shows the response of the system under different assigned poles.

6.5.2 Fuzzy Logic Control

In 1965, Zadeh published his paper (Fuzzy Sets). After that scientists worldwide developed different algorithms to design a fuzzy logic controller (FLC) such as E. Mamdani 1975, Takagi-Sugeno 1985, B-spline Fuzzy and Tsukamoto fuzzy model. Fuzzy inference systems (FIS) introduce a considerable solution to the subject of mobile robots control. The fundamental three phases of FLC are: fuzzification, inference engine design and defuzzification. These three phases are analogous to three phases of stochastic control systems: modeling, identification and controller design. A FLC is an intelligent control system that smoothly interpolates between rules. A fuzzy set may be represented by a mathematical formulation known as a membership function. That is, associated with a given linguistic variable (e.g.

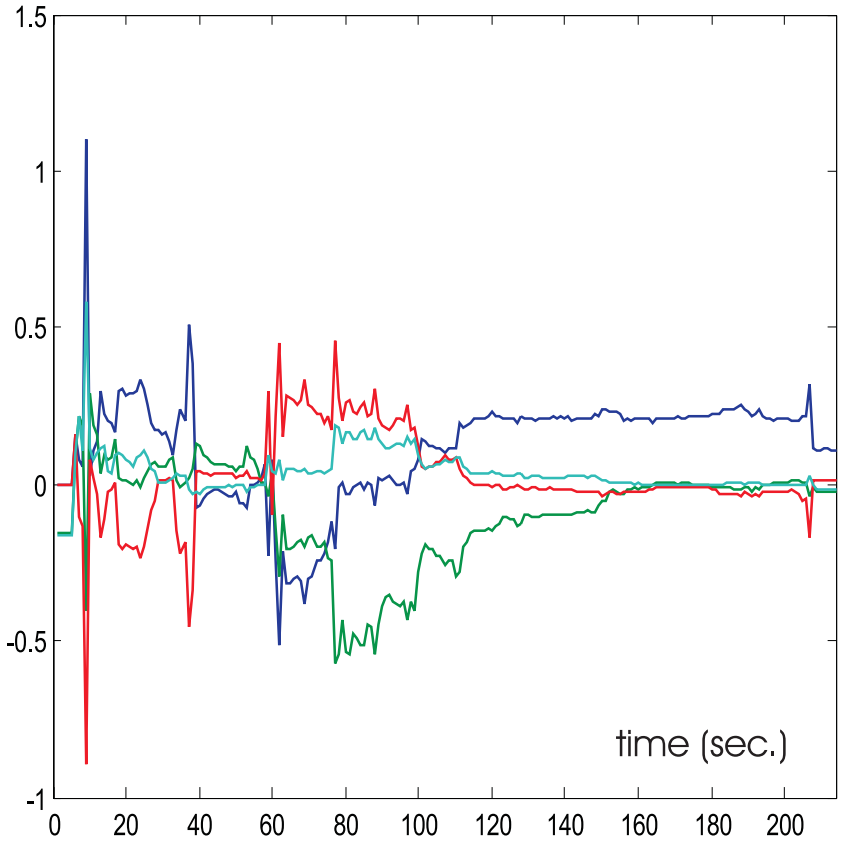
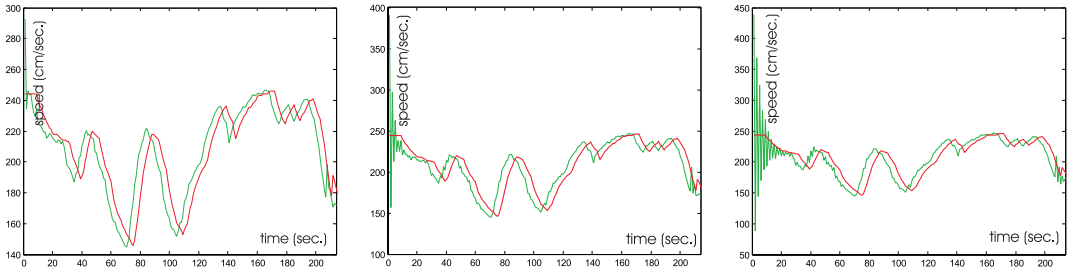


Figure 6.14: Controller parameters



(a) System response at t_1=0.1

(b) System response at t_1=0.3

(c) System response at t_1=0.4

Figure 6.15: System response under different assigned poles (0.1, 0.3, 0.5)

mobile robot velocity) are linguistic values or fuzzy subsets (e.g. slow, fast, etc.), expressed as membership functions, which represent uncertainty, vagueness, or imprecision in values of the linguistic variable, see figure (6.16). This function assigns a numerical degree of membership,in the closed unit interval, to a crisp (precise) number. Within this framework, a membership value of zero/one corresponds to an element that is definitely not/definitely a member of the fuzzy set. Partial membership is indicated by values between 0 and 1. Implementation of a fuzzy controller requires assigning membership functions for inputs and outputs. Inputs are usually measured variables, associated with the state of the controlled plant that are assigned membership values before being processed by an inference engine. The heart of the controller inference engine is a set of if-then rules whose antecedents and consequents are made up of linguistic variables and associated fuzzy membership functions. Fuzzy set intersection, or conjunction, operators in the antecedent are generally referred to as t-norms. They commonly employ algebraic min or product operations on fuzzy membership values. Consequents from different rules are numerically aggregated by fuzzy set union and then defuzzified to yield a single crisp output as the control for the plant.

Integrating Kalman Filters under Fuzzy Systems

The most popular FLC algorithm is the discrete Takagi-Sugeno (TS) fuzzy model, the consequent part of the rules is described by nonfuzzy analytical functions. The TS fuzzy rule is an indirect interpolation of the traditional Kalman filter. The Generalized Rule of fuzzy logic is:

IF *antecedent* **Then** *Consequent*

According to the conception of TS, the interpretation of the preceding fuzzy rule is "***if*** *the model is Kalman based (non linguistic)* ***then*** *fuzzify the output (COG rule)*". Hence, the TS fuzzy control is understood as a reformulation of the Kalman filter. The positive side of TS rule is its capability to damp ripples and fluctuations of the output signal due to using the COG function. The discrete TS considered in this thesis is defined by the following implications:

1. The centroid or the center of gravity (COG) defuzzification rule g is expressed by:
$$g = \frac{\sum_{i=1}^{n} \omega_i g_i}{\sum_{i=1}^{n} \omega_i} \tag{6.56}$$
2. Formulating the state representation as follows
$$g_{k+1} = \frac{\sum_{i=1}^{n} \omega_{(i,k)} (A_c(q^{-1}) g_{(k)} + B_c(q^{-1}) \upsilon_{(k)})}{\sum_{i=1}^{n} \omega_{(i,k)}} \tag{6.57}$$

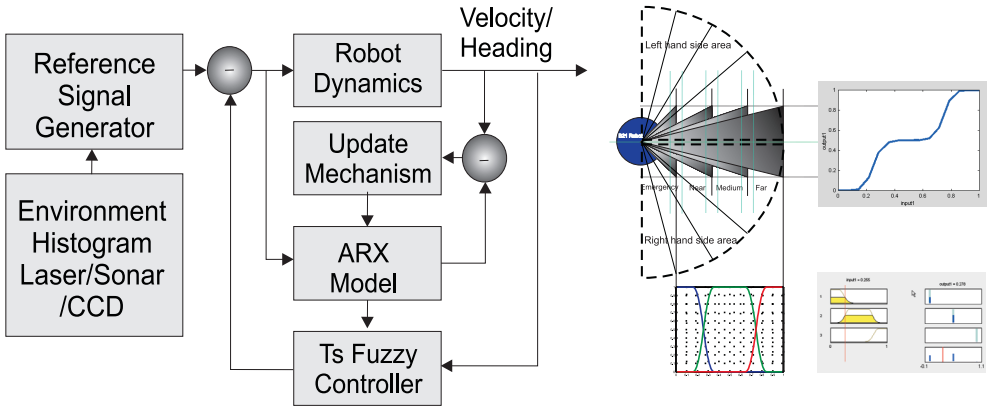


Figure 6.16: Fuzzy control of robot's velocity and heading

3. Calculating the robot control signal $\lambda_{(k)}$

$$\lambda_{(k)} = r_{(k)} - g_{k+1} = r_{(k)} - \frac{\sum_{i=1}^{n} \omega_{(i,k)}(A_c(q^{-1})g_{(k)} + B_c(q^{-1})\upsilon_{(k)})}{\sum_{i=1}^{n} \omega_{(i,k)}} \tag{6.58}$$

Figure 6.17 shows how the robot's output follows the reference signal smoothly, υ and r are highly correlated $\rho > 90\%$. Compared with results of pole assignment control, shown in figure (6.11.a), $\rho \approx 80\%$, the FLC exhibits better performance.

Based on the presented results, we can find that the FLC can deal with model based and behavior based control systems. In collision avoidance, FIS have a better performance than what has been achieved by Kalman filters. The random dynamic obstacles have no definite motion model, therefore Kalman filters failed to cope with them, while FIS succeeded to interact adaptively with those events. On the other side, the model of robot velocity has a clear modeling scheme, which can be easily handled using Kalman filters. We must take into consideration that the COG function fails to deal with some types of membership functions. We developed a switched mode controller that steers the robot under two control rules, the first is the PAC and the second is T-S fuzzy controller. The driving signals are sonar, laser and visual outputs, see figure (6.18). The major differences between both approaches are explained in detail in the next section.

6.5.3 Comments on Kalman and Fuzzy Approaches

To distinguish between the stochastic based and FIS based control systems, the following points have to be recognized:

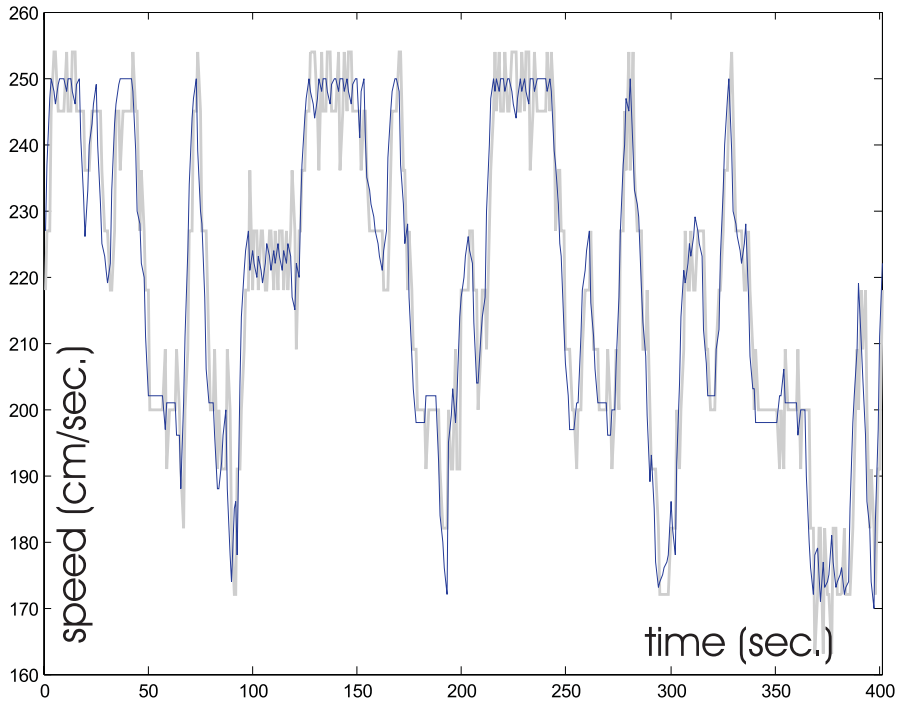


Figure 6.17: Adaptive regulation of mobile robot velocity using FLC

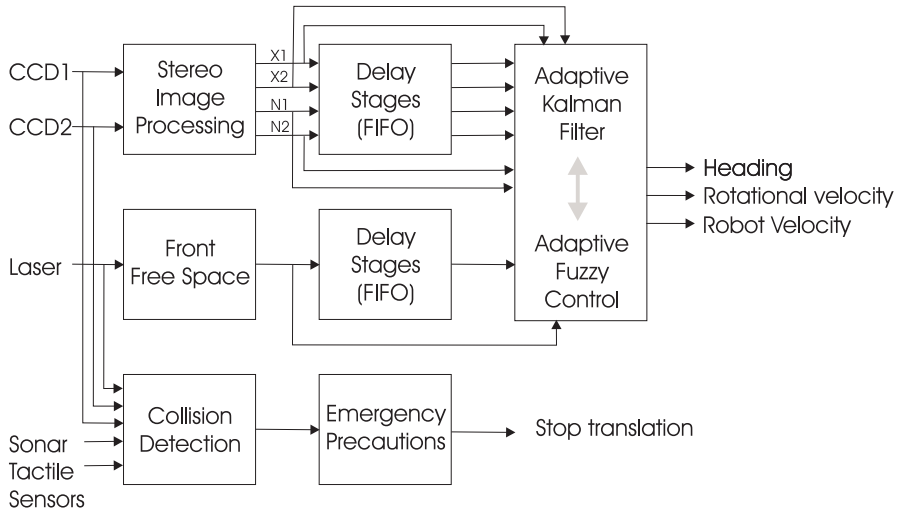


Figure 6.18: Switched MPC Fuzzy controller

- Stochastic control systems are applicable only to well-structured problems such as feedback control of linear or piecewise linear dynamic systems, where our knowledge about the problems is deep and extensive. FIS do not need models to control a system.

- Almost all applications of FIS to traditional system problems amount to nothing more than interpolating or extrapolating among well-known controller designs implemented with stochastic control techniques. However, the world is full of control problems beyond this narrow confine.

- Most problems are poorly understood and described only in natural language terms. For these problems, FIS can play a role either by quantifying imprecise natural language and/or by converting human experience to systematic but logic if-then rules.

- The difference between a logic and a stochastic system is simply an illustration of generality versus depth.

- Stochastic control system analysis tends to be sophisticated compared with the simplicity of FIS.

- Contrary to the stochastic control system, FIS pay a poor reaction to the full interconnection and to the parallelism (multi I/O).

- Logic computing systems support the independent pattern recognition while stochastic systems support the behavior learning.

6.6 Conclusion

This chapter presents an adaptive technique to steer mobile robots. The most important contribution of this technique arises from a formulation of new methods for system modeling and controller design for mobile systems instead of previously dominant approaches. By manipulating the manner in which feature information of system data is incorporated into processing, it can be shown that significant improvements in the performance of the presented algorithms can be attained. Moreover, the simplicity and the efficiency of the adaptive techniques succeeds to reinforce the robustness of the overall system in static and dynamic environments. It is worth mentioning that the intelligent integration of fuzzy systems and Kalman filters leads to achieve the following targets: reinforcement of the overall stability by fusing the sensor data according to fuzzy linguistic rules, enhancement of precision by employing the COG processing function, increment of flexibility by

relying on multichannel data processing and reduction of energy consumption due to eliminating of output fluctuations.

Chapter 7

Conclusion and Future Work

Throughout the presented research work the autonomy of mobile robots is considered as the key subject of the study. To reinforce the autonomy of mobile robots, several novel algorithms are developed to improve the overall system performance.

First of all, the integration of sensor data is the basic tool that underlies robot's mapping, localization and navigation capabilities. The integration of both laser and sonar succeeds to inherit advantages of both sensors and to eliminate their drawbacks by which system safety is assured. Also, the integration between the laser range finder and the magnetic compass is a considerable method to detect features of the natural landmarks. Moreover, the integration between the laser range finder and the binocular vision system is used successfully to steer mobile robots adaptively while the integration of all associated sensors is applied to design the proposed adaptive system.

The development of the vector mapping paradigm (VMP), based on laser ranging, is considered one of the most important contributions of this framework. It preserves the consistency of the conventional mapping algorithms e.g. occupancy grid, topological graph and integration of both of them and at the same time reduces the size of the map drastically. Related to the same algorithm, the model reference is employed to correct the odometric errors relying on a network of central nodes and their corresponding vectors. Compared with the traditional techniques used to correct map odometric errors based on neural networks and evolutionary algorithms, this algorithm is simpler, faster and generates high precision maps.

Regarding matching of laser signatures, the spatial, or spectral analysis and combinations of both of them under wavelets lead to speeding up the matching process whereby the localization capabilities of mobile robots have been improved.

Concerning mapping of a static environment, we employ not only visible odometric data acquired from laser and sonar sensors but also electromagnetic fields detected by a digital compass. Employing the invisible geomagnetic signature as a signature by detecting anomalies of compass deviation is a novel trend that replaces the conventional concept of considering the errors of geomagnetic compasses as a Gaussian white noise. Also in static environments, extending the Bug algorithm to our proposed SLN algorithm succeeds to create a new competitor to the Voronoi diagram.

In dynamic environments, the integration between the universe of discourse in fuzzy logic and the state space in stochastic Kalman filters provide mobile robots with reliable adaptive navigation that maintains their stability, robustness and leads to fine motion.

For the future work, it is considered that the presented work will be extended to develop a high precision localization technique.

Bibliography

[1] A. Aboshosha. Adaptation of Rescue Robot Behaviour in Unknown Terrain Based on Stochastic and Fuzzy Logic Approaches. In *proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2003*, Las Vegas USA, 2003.

[2] A. Aboshosha. Adaptive Visual Guidance of Mobile Robots based on Takagi-Sugeno Model. In *proceedings of the Applied Machine Vision - From Image Processing to Vision Systems conference, GMA - VDI*, Stuttgart, Germany, 2003.

[3] A. Aboshosha. An Introduction to Robot Distributed Sensors. Technical Report WSI-2003-13, WSI, University of Tübingen, BW, Germany, November 2003.

[4] A. Aboshosha. Employing a Recurrent Linear Network to guide Mobile Robots Adaptively. In *Proceedings of the* 3^{rd} *Workshop on Self Organization of Adaptive Behavior SOAVE2004*, Ilmenau, Germany, 2004.

[5] A. Aboshosha, H. Tamimi, and A. Zell. Matching of 2D Laser Signatures based on Spatial and Spectral Analysis. In *Proc. of Robotik 2004*, International Congress Center (ICM), Munich, Germany, June 17./18. 2004.

[6] A. Aboshosha and A. Zell. Robust Mapping and Path Planning for Indoor Robots based on Sensor Integration of Sonar and a 2D Laser Range Finder. In *proceedings of the* 7^{th} *International Conference On Intelligent Engineering Systems INES03*, Assiut-Luxor, Egypt, 2003.

[7] A. Aboshosha and A. Zell. Disambiguating Robot Positioning using Laser and Geomagnetic Signatures. In *proceedings of the* 8^{th} *Conference on Intelligent Autonomous Systems IAS-8*, Amsterdam, The Netherlands, March 10-13 2004.

[8] O. Amidi. *An Autonomous Vision-Guided Helicopter*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1996.

[9] P. Anandan. A Computational Framework and an Algorithm for the Measurement of Visual Motion. *Int. J. Computer Vision*, 2(3):283–310, 1989.

[10] F. Antonio. *Ch IV.6 in graphics Gems III (Ed. D. Kirk)San Diego*, chapter Faster Line Segment Intersection, pages 199–202 and 500–501. Academic press.

[11] M. A. Arbib. *The Handbook of Brain Theory and Neural Network*, chapter Schema theory, pages 830–834. MIT Press, Cambridge, MA, 1995.

[12] R. C. Arkin. Integrating Behavioral, Perceptual, and World Knowledge in Reactive Navigation. *Robotics and Autonomous Systems*, 6:105–122, 1990.

[13] R. C. Arkin. Cooperation without Communication: Multiagent Schema based Robot Navigation. *Journal of Robotic Systems*, 1992.

[14] R. C. Arkin. *Behavior-based Robotics*. The MIT Press, Cambridge, MA., 1998.

[15] M. Ashour and A. Aboshosha. A Computer-Based Digital Signal Processing for Nuclear Scintillator Detectors. In *proceedings of the 7^{th} International Conference on Nuclear Science and Applications*, Cairo, Egypt, 2000.

[16] R. Aufrere, J. Gowdy, C. Mertz, C. Thorpe, C. Wang, and T. Yata. Perception for Collision Avoidance and Autonomous Driving. *Mechatronics*, 13(10):1149–1161, December 2003.

[17] R. Aufrere, C. Mertz, and C. Thorpe. Multiple Sensor Fusion for Detecting Location of Curbs, Walls, and Barriers. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV2003)*, 2003.

[18] T. Belker, M. Beetz, and A. B. Cremers. Learning of Plan Execution Policies for Indoor Navigation. *AI Communications*, 15(1):3–16, 2002.

[19] O. Bengtsson and A-J. Baerveldt. Localization in Changing Environments by Matching Laser Range Scans. In *proceedings of the EURobot*, Zurich, Switzerland, 1999.

[20] O. Bengtsson and M. Jonasson. Robot Localization in Changing Environments Using a Laser Range Scanner. Technical Report CCA-9803, Halmstad University, Halmstad, Sweden, 1998.

[21] H. R. Beom and H. S. Cho. A Sensor-based Navigation for a Mobile Robot Using Fuzzy Logic and Reinforcement Learning. *IEEE Trans. on SMC*, 25(3):464–477, 1995.

[22] H. R. Berenji and P. Khedkar. Learning and Tuning Fuzzy Logic Controller through Reinforcements. *IEEE Trans. On Neural Networks*, 3(5):724–740, 1992.

[23] A. Berler and S. E. Shimony. Bayes Networks for Sonar Sensor Fusion. In *Uncertainty in Artificial Intelligence: Proceedings of the 13th Conference (UAI-1997)*, pages 14–21, San Francisco, CA, 1997. Morgan Kaufmann Publishers.

[24] P. Biber and W. Straßer. The Normal Distributions Transform: A New Approach to Laser Scan Matching. In *IEEE/RJS International Conference on Intelligent Robots and Systems*, Las Vegas, 2003.

[25] P. D. Bitetto, E. N. Johnson, E. Lanzilotta, C. Trott, and M. Bosse. The 1996 MIT/Boston University/Draper Laboratory autonomous helicopter system. In *proceedings of the 1996 Association for Unmanned Vehicle Systems 23rd Annual Technical Symposium and Exhibition*, pages 911–920, Orlando, 1996.

[26] A. Bonarini. Evolutionary Learning of Fuzzy Rules: Competition and Cooperation. In W. Pedrycz, editor, *Fuzzy Modelling: Paradigms and Practice*, pages 265 – 284. Kluwer Academic Press, Norwell, MA, 1997.

[27] J. Borenstein, H. R. Everett, and L. Feng. Where am I? - Sensors and Methods for Mobile Robot Positioning. Technical report, University of Michigan, Michigan, USA, 1996.

[28] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe. Mobile Robot Positioning: Sensors and Techniques. *Journal of Robotic Systems*, 14(4):231–249, 1997.

[29] J. Borenstein and L. Feng. Measurement and Correction of Systematic Odometry Errors in Mobile Robots. *IEEE Transactions on Robotics and Automation*, 12:869–880, 1996.

[30] J. Borenstein and Y. Koren. The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots. *IEEE Trans. Robotic. and Autom.*, 7(3):278–288, 1991.

[31] J. Borenstein and Y. Koren. High-speed Obstacle Avoidance for Mobile Robots. In *proceedings of the IEEE Symposium on Intelligent Control*, pages 382–384, Arlington, Virginia, August 1998.

[32] V. Brajovic and T. Kanade. Tracking Computational Sensor with Attention. *IEEE Journal of Solid-State Circuits*, 33(8), August 1998.

[33] R. A. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal Robotics and Automation*, 2(1):14–23, 1986.

[34] R. Brunelli and T. Poggio. Face Recognition: Features versus Templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[35] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The Interactive Museum Tour-Guide Robot. In *proceedings of the AAAI 13th National Conference on Artificial Intelligence*, 1998.

[36] P. J. Burtand, C. Yen, and X. Xu. Local Correlation Measures for Motion Analysis: a Comparitive Study. In *proceedings of the IEEE Conf. Pattern Recognition Image Processing*, pages 269–274, 1982.

[37] F. Calugi, A. Robertsson, and R. Johansson. Output Feedback Adaptive Control of Robot Manipulators Using Observer Backstepping. In *Proceeding of IROS*, EPFL Lausanne, Switzerland, 2002. IEEE.

[38] J. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, Mass., 1988.

[39] J. Canny, A. Rege, and J. Reif. An Exact Algorithm for Kinodynamic Planning in the Plane. In *the 6th annual symposium on Computational geometry*, pages 271–280, Berkley, California, United States, June 07-09 1990.

[40] C. Cavalcante, J. Cardoso, J. G. Ramos, and O. R. Neves. Design and Tuning of a Helicopter Fuzzy Controller. In *proceedings of the 1995 IEEE International Conference on Fuzzy Systems*, pages 1549–1554, 1995.

[41] D. Challou, M. Gini, and V. Kumar. Parallel Search Algorithm for Robot Motion Planning. In *IEEE Int'l Conference on Robotics and Automation*, pages 46–51, 1993.

[42] K-P. Chan and A. W-C. Fu. Efficient Time Series Matching by Wavelets. In *ICDE*, pages 126–133, 1999.

[43] W. Cheney and D. Kincaid. *Numerical Mathematics and Computing, Third Edition*. Brooks/Cole Publishing Company, 1994.

[44] H. Choset and K. Nagatani. Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization without Explicit Localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, 2001.

[45] A. Cimponeriu and H. Kihl. Intelligent control with the growing competitive linear local mapping neural network for robotic hand-eye coordination. In *Proceedings of the Second International Conference on Knowledge-Based Intelligent Electronic Systems KES '98*, pages 46 –52, 1998.

[46] F. Crow. Summed-Area Tables for Texture Mapping. *Computer Graphics*, 18(3):207–212, 1984.

[47] H. F. Silverman D. I. Barnea. A Class of Algorithms for Fast Digital Image Registration. *IEEE Trans. Computers*, 21:179–186, 1972.

[48] G. Dedeoglu, M. Mataric, and G. S. Sukhatme. Cooperative Robot Mapping. In *proceedings of the 1999 SPIE Conference*, 1999.

[49] G. L. Dirichlet. Über die Reduktion der Positiven Quadratischen Formen mit Drei Unbestimmten Ganzen Zahlen, in German. *Reine Angew. Math.*, 40:209–27, 1850.

[50] D. Driankov and A. Saffiotti. *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*. Physica-Verlag, 2001.

[51] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[52] J. T. Economon, A. Tsourdos, and B. A. White. Takagi-Sugeno Model Synthesis of a Quasi-Linear Parameter Varying Mobile Robot. In *IROS*, EPFL, Lausanne, Switzerland, 2002. IEEE.

[53] A. Elfes. Sonar-based Real-World Mapping and Navigation. *IEEE Journal of Robotics and Automation*, RA 3(3), June 1987.

[54] A. Elfes. Using Occupancy Grids for Mobile Robot Perception and Navigation. *IEEE Computer Magazine, special issue on Autonomous Intelligent Machines*, 22(2):46–58, June 1989.

[55] D. Fox et. al. Position Estimation for Mobile Robots in Dynamic Environments. In *proceedings of the 15^{th} National Conference on Artificial Intelligence*, Madison, Wisconsin, 1998.

[56] O. Wijk et. al. Triangulation based Fusion of Ultrasonic Sensor Data. In *proceedings of the ICRA'98*, Leuven, Belgium, 1998.

[57] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide 4th edition*. Academic Press, San Diego, 1997.

[58] G. Feng. Data Smoothing by Cubic Spline. *IEEE Transactions on Signal Processing*, 46(1):2790–2796, October 1998.

[59] S. Feyrer and A. Zell. Ein integrierter Ansatz zur Lokalisierung von Personen in Bildfolgen, in German. In *Stefan Posch and Helge Ritter (editors), Workshop Dynamische Perzeption*. Infix-Verlag, 1998.

[60] S. Feyrer and A. Zell. Videobasierte Erkennung von Personen und hindernisvermeidendes Folgeverhalten mit einem mobilen Serviceroboter, in German. In *Heinz Woern, Ruediger Dillmann and Dominik Henrich (editors), Autonome Mobile Systeme 1998*, Berlin, Heidelberg, 1998. ľ Springer-Verlag.

[61] S. Feyrer and A. Zell. Detection, Tracking, and Pursuit of Humans with an Autonomous Mobile Robot. In *International Conference on Intelligent Robots and Systems (IROS)*, Kyongju, Korea, 1999. IEEE.

[62] S. Feyrer and A. Zell. Personentracking auf einer mobilen Roboterplattform unter Verwendung eines multimodalen Detektionsansatzes, in German. In *KI, Organ des Fachbereichs 1 (Künstliche Intelligenz) der Gesellschaft fuer Informatik (GI)*, Bremen, Germany, 1999. arenDTaP Desktop Publishing.

[63] S. Feyrer and A. Zell. Tracking and Pursuing Persons with a Mobile Robot. In *International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS)*, Corfu, Greece, 1999.

[64] S. Fomel. Helical Preconditioning and Splines in Tension. *SEP*, 103:289–301, 2000.

[65] T. W. Fong, I. Nourbakhsh, and K. Dautenhahn. A Survey of Socially Interactive Robots: Concepts, Design, and Applications. Technical Report CMU-RI-TR-02-29, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 2002.

[66] M. Foskey, M. Garber, M. Lin, and D. Manocha. A Voronoi-based Hybrid Motion Planner. In *International Conference on Robotic Systems*, 2001.

[67] D. Fox, S. Thrun, F. Dellaert, and W. Burgard. *A. Doucet and N. de Freitas and N. Gordon, Sequential Monte Carlo Methods in Practice*, chapter Particle Filters for Mobile Robot Localization. Springer Verlag, New York, 2000.

[68] N. Franceschini, J. M. Pichon, and C. Blanes. From insect vision to robot vision. *Phil. Trans. R. Soc. B*, 337:283–294, 1992.

[69] M. O. Franz and H. A. Mallot. Biomimetic Robot Navigation. *Robotics and Autonomous Systems*, 30:133–153, 2000.

[70] T. Fukuda, S. Nadagawa, Y. Kawauchi, and M. Buss. Structure Decision for Self Organizing Robots based on Cell Structures - Cebot. In *proceedings of the IEEE International Conference on Robotics and Automation*, pages 695–700, Los Alamitos, CA, 1989.

[71] S. S. GE and Y. J. CUI. Dynamic Motion Planning for Mobile Robots Using Potential Field Method. *Autonomous Robots*, 13:207Ű222, 2002.

[72] R. Goldman. *Ch IV.6 in graphics Gems I (Ed. A. S. Glassner)San Diego*, chapter Intersection of two lines in three-space, page 304. Academic press.

[73] Jack Golten and Andy Verwer. *CONTROL SYSTEM DESIGN AND SIMULATION*. McGraw-Hill, 1991.

[74] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Reading, Massachusetts: Addison-Wesley, third edition, 1992.

[75] S. G. Goodridge, M. G. Kay, and R. C. Luo. Multi-Layered Fuzzy Behaviour Fusion for Reactive Control of an Autonomous Mobile Robot. In *Proceeding of the 6th International Conf on Fuzzy Systems*, Barcelona, SP, 1997. IEEE.

[76] A. Goshtasby, S. H. Gage, , and J. F. Bartholic. A Two-Stage Cross-Correlation Approach to Template Matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(3):374–378, 1984.

[77] B. Graf, R. D. Schraft, and J. Neugebauer. A Mobile Robot Platform for Assistance and Entertainment. In *ISR-2000*, pages 252–253, Montreal, Canada, 2000.

[78] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering Theory and Practice*. Prentice Hall, 1993.

[79] J-S. Gutmann. Vergleich von Algorithmen zur Selbstlokalisierung eines mobilen Roboters, in German. Technical report, Ulm University, BW, Germany, 1996.

[80] J-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An Experimental Comparison of Localization Methods. In *proceedings of the International Conference on Intelligent Robots and Systems*, Victoria, Canada, 1998.

[81] J-S. Gutmann and C. Schlegel. AMOS: Comparison of Scan Matching Approaches for Self Localization in Indoor Environments. Technical Report IEEE 0-8186-7695-7/96, Ulm University, BW, Germany, 1996.

[82] T. V. Guy, M. Kárný, and J. Böhm. Linear adaptive controller based on smoothing noisy data algorithm. In *Proceedings of European Control Conference ECC*, Karlsruhe, Germany, 1999.

[83] R. S. Harvey. Development of a precision pointing system using an integrated multi-sensor approach. Technical Report UCGE Reports, 20117, Department of Geomatics Engineering, University of Calgary, Alberta, Canada, April 1998.

[84] D. Hilbert and S. Cohn-Vossen. *Geometry and the Integration.* New York: Chelsea, 1^{st} edition, 1999.

[85] F. S. Jr. Hill. *Ch II.5 in graphics Gems IV (Ed. P. S. Heckbert)San Diego*, chapter The Pleasure of Per Dot Products, pages 138–148. Academic press.

[86] K. Hoff, T. Culver, J. Keyser, M. Lin, and D. Manocha. Interactive Motion Planning using Hardware Accelerated Computation of Generalized Voronoi Diagrams. In *IEEE Conference on Robotics and Automation*, pages 2931–2937, 2000.

[87] J. Hoschek and D. Lasser. *Grundlagen der geometrischen Datenverarbeitung, in German.* Teubner, Stuttgart, BW, Germany, 1992.

[88] B. Hussien. *Robot Path Planning and Obstacle Avoidance by Means of Potential Function Method.* PhD thesis, University of Missouri-Columbia, Usa, 1989.

[89] R. Isermann, K.-H. Lachmann, and D. Matko. *Adaptive Control Systems.* 1992.

[90] I. Ivanisevic and V. Lumelsky. Human Augmentation in Teleoperation of Arm Manipulators in an Environment with Obstacles. Technical Report RL-99002, University of Wisconsin-Madison, Robotics Laboratory, USA, 1999.

[91] J. C. Latombe J. Barraquand. Robot Motion Planning: A Distributed Representation Approach. *the International Journal of Robotics Research*, 10(6):628–649, Dec. 1991.

[92] O. L. R. Jacobs. *Introduction to Control Theory.* Oxford University Press, 2 edition, 1993.

[93] A. Jense and A. la Cour-Harbo. *Ripples in Mathematics: the Discrete Wavelet Transform*. Springer, 2001.

[94] L. Jouffe. Fuzzy Inference System Learning by Reinforcement Methods. *IEEE Trans. On SMC-Part B*, 28(3):338–355, 1998.

[95] C. F. Juang, J. Y. Lin, and C. T. Lin. Genetic Reinforcement learning through Symbiotic Evolution for Fuzzy Controller Design. *IEEE Transactions on SMC-Part B*, 30(2):290–301, 2000.

[96] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, Seattle, Washington, USA, 1995.

[97] M. Kalandros and L. Y. Pao. Covariance Control for Multisensor Systems. In *IEEE Trans. Aerospace Electronic Systems*, 2002.

[98] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME Journal of Basic Engineering*, pages 35–45, MARCH 1960.

[99] E. Keogh, S. Lonardi, and W. Chiu. Finding Surprising Patterns in a Time Series Database in Linear Time and Space, 2002.

[100] E. J. Keogh and M. J. Pazzani. Scaling up Dynamic Time Warping for Data Mining Applications. In *Knowledge Discovery and Data Mining*, pages 285–289, 2000.

[101] O. Khatib. *Commande Dynamique dans l'Espace Operationnel des Robots Manipulateurs en Presence d'Obstacles, in French*. PhD thesis, Ecole Nationale Superieure de l'Aeronautique et de l'Espace Toulouse, France, 1980.

[102] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1), Spring 1986.

[103] P. Khosla and R. Volpe. Superquadric Artificial Potentials for Obstacle Avoidance and Approach. In *IEEE Conf. Robotic. Autom.*, pages 1778–1784, 1988.

[104] B. S. Kim. *Nonlinear Flight Control Using Neural Networks*. PhD thesis, Department of Aerospace Engineering, Georgia Institute of Technology., Georgia, USA, 1993.

[105] F. Kobayashi, Y. Tanabe, T. Fukuda, and F. Kojima. Acquisition of Sensor Fusion Rule Based on Environmental Condition in Sensor Fusion System. In *proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS Int'l Conf. (IFSA 2001)*, pages 2096–2101, 2001.

[106] G.E. Krasner and S.T. Pope. A cookbook for using the model-view-controller user interface paradigm in smalltalk-80. *Journal of Object-Oriented Programming*, 1(3):26–49, August/September 1988.

[107] B. H. Krogh. A Generalized Potential Field Approach to Obstacles Avoidance Control. In *First World Conference on Robotics Research, RIA*, 1984.

[108] C. R. Kube and H. Zhang. Collective Robotic Intelligence, in From Animals to Animats. In *proceedings of the International Conference on Simulation of Adaptive Behavior*, pages 460–468, 1992.

[109] C. Kuglin and D. Hines. The Phase Correlation Image Alignment Method. In *proceedings of the International Conference of Cybernetics and Society*, pages 163–165, 1975.

[110] D. Lambrinos, R. Moeller, T. Labhart, R. Pfeifer, and R. Wehner. A Mobile Robot Employing Insect Strategies for Navigation. *Robotics and Autonomous Systems*, 30:39–64, 2000.

[111] J. Latombe. *Robot Motion Planning*. Kluwer Academic Publications, Boston, USA, 1995.

[112] S. L. Laubach. *Theory and Experiments in Autonomous Sensor-Based Motion Planning with Applications for Flight Planetary Microrovers*. PhD thesis, California Institute of Technology, Pasadena, California, USA, May 24 1999.

[113] J. P. Laumond and P. Soueres. Metric Induced by the Shortest Paths for a Car-Like Mobile Robot. In *IEEE IROS'93*, Yokohama, Japan, July 1993.

[114] J. Leonard and H. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Press, 1992.

[115] J. Leonard, H. DurrantWhyte, and I. Cox. Dynamic Map Building for an Autonomous Mobile Robot. *Journal of Robotics Research*, 11:89–96, 1992.

[116] J. P. Lewis. Fast Template Matching. In *proceedings of the Vision Interface Conference*, pages 120–123, 1995.

[117] A. R. Lindsey. The Non-Existence of a Wavelet Function Admitting a Wavelet Transform Convolution Theorem of the Fourier Type T. Technical Report C3BB, Rome Laboratory, 1995.

[118] K. J. LOHMANN, N. D. PENTCHEFF, G. A. NEVITT3, G. D. STETTEN, R. K. ZIMMER-FAUST, H. E. JARRARD3, and L. C. BOLES. Magnetic orientation of spiny lobsters in the ocean: Experiments with undersea coil systems. *The Journal of Experimental Biology*, 198:2041–2048, 1995.

[119] L. Seabra Lopes, N. Lau, and L.P. Reis. Intelligent control and decision-making demonstrated on a simple compass-guided robot. In *Proceedings of the 2000 IEEE International Conference on Systems, Man and Cybernetics*, Nashville, Tennessee, USA, October 2000.

[120] F. Lu and E. E. Milios. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. Technical Report RBCV-TR-94-46, University of Toronto, Toronto, Canada, 1995.

[121] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *proceedings of the IJCAI*, 1981.

[122] V. J. Lumelsky and A. A. Stepanov. Path-Planning Strategies for a Point Mobile Automaton Moving amidst Obstacles of Arbitrary Shape. *Algorithmica*, 2(403), 1987.

[123] D. Y. Maharaj. *The Application of Nonlinear Control Theory to Robust Helicopter Flight Control*. PhD thesis, Department of Aeronautics, Imperial College of Science, Technology, and Medicine, 1994.

[124] S. Mallat. A Theory of Multi Resolution Signal Decomposition: the Wavelet Representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[125] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA, USA, 1998.

[126] S. Mallat and W. L. Hwang. Singularity Detection and Processing with Wavelets. *IEEE Transaction on Information Theory*, 38(2):617–643, March 1992.

[127] S. Mallat and S. Zhong. Characterization of Signals from Multiscale Edges. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14:710–732, March 1992.

[128] E. H. Mamdani. Application of Fuzzy Logic to Approximate Reasoning using Linguistic Synthesis. *IEEE trans. on Computers*, C-26(12):1182–1191, 1977.

[129] M. C. Martin and H. Moravec. Robot Evidence Grids. Technical Report CMU-RI-TR-96-06, CMU RI, USA, March 1996.

[130] M. Mataric. Integration of Representation into Goal-Driven Behavior-based Robots. *IEEE Transactions on Robotics and Automation*, 8(3):304–312, 1992.

[131] M. Mataric. *Interaction and Intelligent Behavior*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA, 1994.

[132] M. J. Mataric. Minimizing Complexity in Controlling a Collection of Mobile Robots. In *proceedings of the IEEE International Conference on Robotics and Automation*, pages 830–835, Nice, France, 1992.

[133] M. J. Mataric. Interaction and Intelligent Behavior. Technical Report AI-TR-1495, MIT Artificial Intelligence Lab., USA, 1994.

[134] M. J. Mataric. Issues and Approaches in the Design of Collective Autonomous Agents. *Robotics and Autonomous Systems*, 16(2–4):321–331, 1995.

[135] P. McKerrow. *Introduction to Robotics*. Addison-Wesley, 1998.

[136] S. K. Mitra and J. F. Kaiser. *Handbook for Digital Signal Processing*. New York: Wiley, 1993.

[137] R. Miyagawa and T. Kanade. Integration-Time Based Computational Image Sensors, in German. Technical Report 65, ITE, November 1995.

[138] M.Juan, Martín Sánchez, and José Rodellar. Adaptive predictive control: From the concepts to plant optimization. 1996.

[139] A. Mojaev. *Umgebungswahrnehmung, Selbstlokalisation und Navigation mit einem mobilen Roboter, in German*. PhD thesis, University of Tübingen, 2001.

[140] A. Mojaev and A. Zell. Sonardaten-Integration für autonome mobile Roboter, in German. In *Tagungsband zum 20. DAGM-Symposium Mustererkennung*, pages 556–563, Stuttgart, Germany, 1998.

[141] J. F. Montgomery and G. A. Bekey. Learning Helicopter Control through Teaching by Showing. In *proceedings of the IEEE Conference on Decision and Control*, 1998.

[142] J. F. Montgomery, A. H. Fagg, and G. A. Bekey. The USC AFV-I: A Behavior-based Entry in the 1994 International Aerial Robotics Competition. *IEEE Expert*, 10(2):16–22, 1995.

[143] A. W. Moor and L. P. Kaelbling. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence*, 4:237–285, 1996.

[144] H. Moravec and A. Elfes. High Resolution Maps from Wide Angle Sonar. In *proceeding of the 1985 IEEE International Conference on Robotics and Automation*, pages 116–121, St. Louis, March 1985.

[145] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette. Evolutionary Algorithms for Reinforcement Learning. *Journal of Artificial Intelligent Research*, 11:241–276, 1999.

[146] W. Nelson. Continuous-Curvature Paths for Autonomous Vehicles. In *IEEE Int. Conf. Robotics Automat.*, pages 1260–1264, Scottsdale, AZ, May 1989.

[147] P. Newman. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem*. PhD thesis, Univ. of Sydney, 2000.

[148] N. Nilsson. A Mobile Automaton: An application of Artificial Intelligence Techniques. In *proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-69)*, Washington D.C., USA, 1969.

[149] B. Nodland and J. P. Ralston. Indication of Anisotropy in Electromagnetic Propagation over Cosmological Distances. *Physical Review Letters*, 78:3043–3046, 1997.

[150] E. L. Oberstar. Advanced Digital Signal Processing Adaptive Linear Prediction Filter (Using The RLS Algorithm), 2001.

[151] K. Ogata. *Modern Control Engineering*. Pearson Education, 4^{th} edition edition, 13 November 2001.

[152] L. Ojeda and J. Borenstein. Experimental Results with the KVH C-100 Fluxgate Compass in Mobile Robots. In *proceedings of the IASTED International Conference Robotics and Applications*, Honolulu, Hawaii, USA, 2000.

[153] N. Okello, D. Tang, and D. W. McMichael. Tracker: A Sensor Fusion Emulator for Generalised Tracking. In Robin Evans, Lang White, Daniel McMichael, and Len Sciacca, editors, *Proceedings of Information Decision and Control 99*, pages 359–364, Adelaide, Australia, February 1999. Institute of Electrical and Electronic Engineers, Inc.

[154] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing*. Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

[155] L. Paletta, M. Prantl, and A. Pinz. Learning Temporal Context in Active Object Recognition Using Bayesian Analysis. In *proceedings of the 15th International Conference on Pattern Recognition (ICPR 2000)*, volume 1, pages 695–699, Piscataway, NJ, 2000. IEEE Press.

[156] D. Payton, D. Keirsey, D. Kimble, J. Krozel, and J. Rosenblatt. Do Whatever Works: A robust Approach to Fault-Tolerant Autonomous Control. *Applied Intelligence*, 2(3):225–250, 1992.

[157] D. Pedoe. *Circles: A Mathematical View*. Washington, DC: Math. Assoc. Amer., rev. edition, 1951.

[158] Percival and Walden. *Wavelet Methods for Time Series Analysis*. Cambridge University Press, 2000.

[159] R. Pfeifer and Ch. Scheier. *Understanding Intelligence*. The MIT Press, 1999.

[160] T. Pfeifer and J. Thiel. Untersuchung eines absolutmessenden Interferometers mit Halbleiterlaser (Weiterführende Ansätze in der Interferometrie, in German. Technical report, VDI Technologiezentrum Physikalische Technologien, 1995.

[161] C. Phillips, C. L. Karr, and G. Walker. Helicopter Flight Control with Fuzzy-Logic and Genetic Algorithms. *Engineering Applications of Artificial Intelligence*, 9(2):175–184, 1996.

[162] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Interpolation Artefacts in Mutual Information-based Image Registration. *Computer Vision and Image Understanding*, 77(2):211–232, 2000.

[163] W. Pratt. *Digital Image Processing*. John Wiley, New York, 1978.

[164] W. Reeb. Varianten der Entfernungsbestimmung mit dem Laser. *Laser+Photonik*, pages 52–55, June 2003.

[165] E. Rimon. *Exact Robot Navigation using Artificial Potential Functions*. PhD thesis, Yale University, 1990.

[166] T. Ritz. Disrupting Magnetic Compass Orientation with Radio Frequency Oscillating Fields . In *RIN01: Orientation & Navigation - Birds, Humans & other Animals*, Royal Institute of Navigation, Oxford, 2001.

[167] T. Röfer. Using Histogram Correlation to Create Consistent Laser Scan Maps. In *the IEEE International Conference on Robotics Systems (IROS-2002)*, pages 625–630, EPFL, Lausanne, Switzerland, 2002.

[168] N. C. Rowe and R. S. Alexander. Finding Optimal-Path Maps for Path Planning Across Weighted Regions. *International Journal of Robotics Research*, 19(2):83–95, 2000.

[169] N. Roy, G. Dudek, and M. Daum. Mobile Robot Navigation and Control: A Case Study. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, volume 2, page 1801, Portland, OR, 1996.

[170] T. W. Ryan. *The Prediction of Cross-Correlation Accuracy in Digital Stereo-Pair Images*. PhD thesis, University of Arizona, USA.

[171] I. Savnik, G. Lausen, H-P. Kahle, H. Spiecker, and S. Hein. Algorithm for Matching Sets of Time Series. In *Principles of Data Mining and Knowledge Discovery*, pages 277–288, 2000.

[172] A. Schmolke and H. A. Mallot. *Polarisation Compass for Robot Navigation*, chapter In: Bülthoff, Gegenfurtner, Mallot, Ulrich (Eds.) TWK 2002. Knirsch Verlag, 2002.

[173] R. D. Schraft, B. Graf, A. Traub, and D. John. PolarBugŰein effizienter Algorithmus zur reaktiven Hindernisumfahrung, in German. In *AMS-2000*, 2000.

[174] R. D. Schraft, B. Graf, A. Traub, and D. John. A Mobile Robot Platform for Assistance and Entertainment. *Industrial Robot Journal*, 28:29–34, 2001.

[175] D. Schulz and W. Burgard. Probabilistic State Estimation of Dynamic Objects with a Moving Mobile Robot. *Robotics and Autonomous Systems*, 34(2-3), 2001.

[176] A. Scott, E. L. Parker, and C. Touzet. Quantitative and Qualitative Comparison of Three Laser-Range Mapping Algorithms using Two Types of Laser Scanner Data. In *proceedings of Systems, Man, and Cybernetics Conference*, 2000.

[177] Y. P. Shan. An event-driven model-view-controller framework for smalltalk. *SIGPLAN Notices*, 24(10):347–352, Oktober 1989.

[178] M. Sharir. Robot Motion Planning. *Comm. Pure Applied Math.*, 48:1173–1186, 1995.

[179] J. Shi and C. Tomasi. Good Features to Track. In *proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, 1994.

[180] Z. Shiller. Motion Planning for Mars Rover. In *First workshop on Robot Motion and Control (ROMOCO '99)*, Kiekrz, Poland, June 28 1999.

[181] Z. Shiller. Obstacle Traversal for Space Exploration. In *IEEE Int. Conf. on Robotics and Automation*, San Francisco, USA, April 2000.

[182] T. Simeon and B. Darce-Wright. A Practical Motion Planner for All-Terrain Mobile Robots. In *International Conference on Intelligent Robots and Systems*, 1993.

[183] R. Smith and P. Cheesman. on the estimation and representation of spatial uncertainty. *International Journal of robotics research*, 5(4):56–68, Winter 1987.

[184] A. Speck. *Objektorientierte Software-Technik für industrielle Steuerungssysteme.* PhD thesis, der Fakultät für Informatik der Eberhard-Karls-Universität, Tübingen, BW, Deutschland, 2000.

[185] M. V. Srinivasan, J. S. Chahl, and S. W. Zhang. Robot Navigation by Visual Dead-Reckoning: Inspiration from Insects. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(1):35–47, 1997.

[186] M. V. Srinivasan, S. W. Zhang, and N. J. Bidwell. Visually Mediated Odometry in Honeybees. *Journal of Experimental Biology*, 200:2513–2522, 1997.

[187] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, 1997.

[188] K. J. Åstrom and B. Wittenmark. *Computer-Controlled Systems, Theory and Design.* Prentice Hall, New Jersey, USA, 3 edition, 1997.

[189] A. Stroupe, M. C. Martin, and T. Balch. Distributed Sensor Fusion for Object Position Estimation by Multi-Robot Systems. In *proceedings of the IEEE International Conference on Robotics and Automation.* IEEE, May 2001.

[190] M. Sugeno, M. F. Griffin, and A. Bastian. Fuzzy Hierarchical Control of an Unmanned Helicopter. In *proceedings of the 17th IFSA World Congress*, 1993.

[191] M. Sugeno and T. Yasukawa. A Fuzzy Logic based Approach to Qualitative Modeling. *IEEE Transactions on Fuzzy Systems*, 1(1):7–31, 1993.

[192] S. Suksakulchai, S. Thongchai, D. M. Wilkes, and K. Kawamura. Mobile Robot Localization using an Electronic Compass for Corridor Environment. In *proceedings of the 2000 IEEE International Conference on Systems Man and Cybernetics*, Nashville, Tennessee, USA, 2000.

[193] P. Svestka and M. H. Overmars. Multi-Robot Path Planning with Super-Graphs. In *CESA'96 IMACS Multiconference*, Lille, France, July 1996.

[194] T. Takagi and M. Sugeno. Fuzzy Identification of Systems and Its Application to Modeling and Control. *IEEE Transactions on Systems Man and Cybernetics*, 15(1):116–132, 1985.

[195] C. Thorpe. Path relaxation: Path planning for a mobile robot. In *The National Conference on Artificial Intelligence,aaai-84*, August 1984.

[196] C. Thorpe, J. D. Carlson, D. Duggins, J. Gowdy, R. MacLachlan, C. Mertz, A. Suppe, and C. Wang. Safe robot driving in cluttered environments. In *proceedings of the 11th International Symposium of Robotics Research*, 2003.

[197] C. Thorpe, O. Clatz, D. Duggins, J. Gowdy, R. MacLachlan, J. R. Miller, C. Mertz, M. Siegel, C. Wang, and T. Yata. Dependable Perception for Robots. In *Proceedings of International Advanced Robotics Programme IEEE*, Seoul, Korea, 2001. Robotics and Automation Society.

[198] S. Thrun and A. Bücken. Integrating Grid-Based and Topological Maps for Mobile Robot Navigation. In *proceedings of the National Conference on Artificial Intelligence AAAI-96*, 1996.

[199] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Henning, T. Hofmann, M. Krell, and T. Schmidt. Map Learning and High-Speed Navigation in RHINO. In D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors, *AI-based Mobile Robots: Case Studies of Successful Robot Systems*. MIT Press, 1998.

[200] Q. Tian and M. N. Huhns. Algorithms for Subpixel Registration. *CVGIP*, 35:220–233, 1986.

[201] M. Unser. Splines: A Perfect Fit for Signal and Image Processing. *IEEE Signal Processing Magazine*, 16(6):22–Ũ38, 1999.

[202] M. Unser, A. Aldroubi, and M. Eden. B-spline Signal processing: Part II-Efficient Design and Applications. *IEEE Transactions on Signal Processing*, 41(2):834–848, 1993.

[203] G. Vachkov and T. Fukuda. Identification and Control of Dynamical Systems based on Cause-Effect Fuzzy Models. *IEEE Trans. Robotic. and Autom.*, pages 2072–2077, 2001.

[204] G. Vachkov and T. Fukuda. Structured Learning and Decomposition of Fuzzy Models for Robotic Control Applications. *J. of Intelligent and Robotic Systems*, 2001.

[205] A. Varga. Robust pole assignment techniques via state feedback. In *Proceedings of IEEE Conference on Decision and Control, CDC*, Sydney, Australia, 2000.

[206] S. J. Vestli. *Fast, Accurate and Robust Estimation of Mobile Robot Position and Orientation.* PhD thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, 1995.

[207] F. P. Volpe. Erzeugung von optischen Pikosekundenimpulsen hoher Leistung mit Halbleiterlasern zur Nahbereichs-Entfernungsmessung mit Submillimetermessgenauigkeit, in German. Technical Report Reihe 9: Elektronik, Nr. 202, VDI Verlag, 2002.

[208] G. M. Voronoi. Nouvelles Applications des Parametres Continus a la Theorie des Formes Quadratiques. Deuxieme Memoire: Recherches sur les Parallelloedres Primitifs, in French. *Reine Angew. Math.*, 134:198–287, 1908.

[209] R. Wehner. Polarized-Light Navigation by Insects. *Scientific American*, 23(1):106–115, 1976.

[210] G. Weiss and E. V. Puttkamer. A Map based on Laser Scans without Geometric Interpretation. In *proceedings of the 4^{th} Intelligent Autonomous Systems (IAS-4)*, Karlsruhe, Germany, 1995.

[211] P. E. Wellstead and M.B. Zarrop. *Self-tuning Systems Control and Signal Processing.* Wiley, 1991.

[212] W. WILTSCHKO, U. MUNRO, H. FORD, and R. WILTSCHKO. Effect of a magnetic pulse on the orientation of silvereyes during spring migration. *The Journal of Experimental Biology*, 201, 1998.

[213] L. A. Zadeh. Fuzzy Sets. *Info. and Control*, 8:338–353, 1965.

[214] L. A. Zadeh. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE trans. Systems, Man & cybernetics*, 1:28–44, 1973.

[215] L. A. Zadeh. The Concept of a Linguistic Variable and its Application to Approximate Reasoning. *Information Science*, 8:43–80, 1975.

[216] L. A. Zadeh. Making Computers Think Like People. *IEEE Spectrum*, pages 26–32, 1984.

[217] L. A. Zadeh. Fuzzy logic. *IEEE Computer*, 21(4):83–93, 1988.

[218] J. S. Zelek. Dynamic issues for mobile robot real- time discovery and path planning. In *IEEE CIRA*, November 1999.

[219] U. R. Zimmer and E. Puttkamer. Realtime-learning on an autonomous mobile robot with neural networks. In *the Euromicro'94 Realtime Workshop*, Vaesteraas (Västerås), Sweden, June 15-17 1994.

[220] H. J. Zimmermann. *Fuzzy Sets, Decision Making and Expert Systems*. Kluwer, Boston, USA, 1987.

Index