Design and Implementation of Control Techniques for Differential Drive Mobile Robots: An RFID Approach

by

Suruz Miah

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For the degree Doctor of Philosophy in

Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Graduate and Postdoctoral Studies

University of Ottawa

Abstract

Localization and motion control (navigation) are two major tasks for a successful mobile robot navigation. The motion controller determines the appropriate action for the robot's actuator based on its current state in an operating environment. A robot recognizes its environment through some sensors and executes physical actions through actuation mechanisms. However, sensory information is noisy and hence actions generated based on this information may be non-deterministic. Therefore, a mobile robot provides actions to its actuators with a certain degree of uncertainty. Moreover, when no prior knowledge of the environment is available, the problem becomes even more difficult, as the robot has to build a map of its surroundings as it moves to determine the position. Skilled navigation of a differential drive mobile robot (DDMR) requires solving these tasks in conjunction, since they are inter-dependent. Having resolved these tasks, mobile robots can be employed in many contexts in indoor and outdoor environments such as delivering payloads in a dynamic environment, building safety, security, building measurement, research, and driving on highways.

This dissertation exploits the use of the emerging Radio Frequency IDentification (RFID) technology for the design and implementation of cost-effective and modular control techniques for navigating a mobile robot in an indoor environment. A successful realization of this process has been addressed with three separate navigation modules. The first module is devoted to the development of an indoor navigation system with a customized RFID reader. This navigation system is mainly pioneered by mounting a multiple antenna RFID reader on the robot and placing the RFID tags in three dimensional workspace, where the tags' orthogonal position on the ground define the desired positions that the robot is supposed to reach. The robot generates control actions based on the information provided by the RFID reader for it to navigate those pre-defined points.

On the contrary, the second and third navigation modules employ custom-made RFID

tags (instead of the RFID reader) which are attached at different locations in the navigation environment (on the ceiling of an indoor office, or on posts, for instance). The robot's controller generates appropriate control actions for it's actuators based on the information provided by the RFID tags in order to reach target positions or to track pre-defined trajectory in the environment. All three navigation modules were shown to have the ability to guide a mobile robot in a highly reverberant environment with variant degrees of accuracy.

Acknowledgements

I would like to start by expressing the sincerest gratitude to my thesis supervisor, Dr. Wail Gueaieb, for all of his insight and guidance throughout my graduate studies at the University of Ottawa. I also thank my supervisor for enlightening my own capabilities that I was not aware of at all before joining his research group. The credit goes to him for my personal success and for what I am today as a researcher. The extensive ideas highlighted in this thesis could not have been come into existence without his continuous discussions, thorough feedback, and invaluable support. I would like to pass my sincerest thanks to my undergraduate teacher, Dr. Md. Abdur Rahman who suggested me to work with such a knowledgable professor. I am basically lucky to have Dr. Wail Gueaieb as my graduate studies supervisor.

I would like to acknowledge all committee members for evaluating, highlighting vital points, and suggesting potential future research directions of my thesis. My heartiest thanks to Professor Nasir Uddin Ahmed for introducing me several mathematical concepts which were basically the corner stones of some parts of my thesis. I would like to thank our faculty lab technician Mr. Leo Denner for helping me building some of the boards I used in my research and my colleagues at the MIRaM lab for sharing some excellent moments during my graduate studies.

My best regards to my beloved parents for their unconditional love throughout my life and to open my eyes for exploring this universe. I am also grateful to my younger brother, Md. Sawkat Hossain, and my four sisters for inspiring me to study abroad and for taking care of my parents in my absence back home. Finally, I pass my heartiest love to my wife Mehnaz Tarannum for her patience and support during my PhD tenure. Besides that, all my affections are for my daughter, Mahee, who is our ever peaceful gift on this universe. Her smiling face makes me the happiest person in the world when I return home after finishing my work of whole day. At last, I would like to thank my Almighty Creator for giving me the capabilities to explore this world for knowledge.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AGV	Automated Guided Vehicle
CAD	Computer-Aided Design
DDMR	Differential Drive Mobile Robot
DOCT	Direction Of Current Target RFID tag
EKF	Extended Kalman Filter
FLC	Fuzzy Logic Controller
FEKO	Electromagnetic Simulation Software
GPS	Global Positioning System
IAEKF	Iterative Adaptive EKF
LOS	Line-Of-Sight
MC	Make Controller
NN	Neural Network
RPS	Relative Positioning System
RF	Radio Frequency
RFID	Radio Frequency IDentification
RMSE	Root Mean Squared Error
RSS	Received Signal Strength
SLAM	Simultaneous Localization and Mapping
SNR	Signal-to-Noise Ratio
UKF	Uncented Kalman Filter

List of Symbols

$\mathcal{H}_{\mathbf{K}}$	gradient of Hamiltonian $\mathcal{H}$ with respect to $\mathbf{K}$
h	height of the RFID tag position from the ground
$I \equiv [t_0, t_f]$	time interval
$\mathbf{I}_n$	Identity matrix of dimension n
$\mathbf{h}[\cdot]$	function representing the measurement model
$\mathbf{K}(t)$	feedback control gain at time t
k	discrete time index
k_f	number of discrete time steps
$\mathcal{L}$	Lebesgue measurable function space
N	number of iterations
$\mathcal{O}(\cdot)$	computational complexity using big-O notation
$\mathbf{P}(t)$	solution of differential matrix Riccati equation
$\mathbf{q}(t)$	robot's actual pose at time t
$\hat{\mathbf{q}}(t)$	robot's estimated pose at time t
$\mathbf{q}^d(t)$	robot's desired pose at time t
$\bar{\mathbf{q}}_k$	augmented state (robot plus tag) at time instant k (for SLAM problem)
$\mathbf{q}_t^j \in \mathbb{R}^3$	jth tag position
$\mathbf{S}(t)$	robot's state error covariance matrix at time t
$\bar{\mathbf{S}}_k$	augmented state covariance matrix at time instant k (for SLAM problem)

s	number of RFID tags in the environment
t_0, t_f	initial and final time instants
$\mathbf{u}(t)$	robot's control input (speed) vector at time t
$\mathbf{z}(t)$	vector of measurements at time t
$\boldsymbol{\psi}$	costate variable (Lagrange multiplier)
$Tr[\cdot]$	trace of matrix $[\cdot]$
$diag(\ldots)$	diagonal elements of a square matrix
$\boldsymbol{\zeta}(t)$	measurement noise vector at time t
$\boldsymbol{\xi}(t)$	process noise vector at time t
$(\cdot)^o, (\cdot)^\varepsilon, (\cdot)_{ad}$	optimal, perturb, admissible value of $(\cdot)$
$dJ(\cdot;\boldsymbol{\nu})$	Gateaux (directional) derivative of J in direction $\boldsymbol{\nu}$

Chapter 1

Introduction

1.1 Overview

Robust control design methods are of a paramount importance and fundamental to the successful and autonomous operation of mobile robots. A spectrum of mechanism exists in the literature for mobile robots to navigate from one point to another or to follow a pre-defined trajectory. In some cases, the simplest set of robot's movements can produce complex motion trajectories and consequently robot's motion control/navigation still stands as a challenging problem. Most of the control techniques in the literature are based on either model-based control or machine learning approaches which are tailored towards particular structured environment or driven by an overwhelming degree of computational complexity [Peters et al., 2000]. In addition, some robot navigation systems are characterized by excessive hardware configurations, static and/or dynamic environment sensors, to name just a few. For them to operate in real time, the hosting robot must be equipped with dedicated hardware with high processing power, which increases the cost of the overall system. Moreover, those techniques require a high level of expertise and effort to implement. This makes practical realization of these control techniques in most real-world robotic systems questionable.

Among the most common mobile robot navigation strategies, a large amount of effort has been put on navigation strategies where vision sensors are employed as key perception mechanisms for a robot to realize its location. Undoubtedly, these navigation schemes provide satisfactory performance in terms of robot's pose[1] estimation capability. Often, the performance is even better when sensor fusion techniques are incorporated for such navigation systems. While vision-based navigation systems have significant advantages for robot's localization, they often suffer from restrictive operating environments, deploying complex image processing techniques, dedicated hardware, and a prohibitive computational complexity. For example, they fail to operate mobile robots which are deployed for night navigation. Moreover, it is often hard to use vision information as a navigation means as opposed to robot localization one.

The rising prominence of guiding mobile robots in many real-life applications, such as service robots, for example, necessitates the development of a new generation of indoor navigation systems. Hence, the primary objective of this thesis is to design and implement modular and cost-effective navigation (or motion control) strategies for mobile robots to navigate in indoor environment with high accuracy and robustness. Undoubtedly, for a mobile robot to operate in real-time, a reliable perception system is required to collect measurements describing both the operating environment and its pose. These measurements are, however, invariably corrupted by ambient noise. As such, the secondary objective of this thesis is to reduce the navigation uncertainty introduced by the measurement noise and to record the pose of the mobile robot as accurately as possible. In order to achieve this goal, the mobile robot must have a methodology to filter out this noise. Note that the only perception system incorporated by the mobile robot in this thesis is the emerging Radio Frequency IDentification (RFID) technology.

In this thesis, a cheap and modular solution is presented for the robot navigation problem, taking advantage of the affordability of the RFID technology. The proposed control approaches

[1] It is defined in two dimensional Cartesian space as triplet (x, y, θ), with (x, y) being the position and θ being the orientation of the mobile robot.

mainly generate control actions through a feedback signal with the help of an RFID system deployed in the robot's operating environment. In particular, one key contribution of this thesis is to produce optimal control laws for mobile robots to reach a target position or to track a pre-defined trajectory. These control laws generate control actions for the robot's actuators based on the feedback signal originated from the RFID system. Although a mobile robot is confronted with many challenging tasks when deployed in a real-world environment; such as localization and navigation, path planning, obstacle avoidance, to name just a few, it is important to stress out that this dissertation pertains to the two former tasks only (localization and navigation), as depicted in Figure 1.1. Nevertheless, the proposed navigation strategies are not meant to substitue vision-based robot navigation systems, rather, they might be regarded as promising alternatives where vision systems fail to operate, night navigation environments, for example.

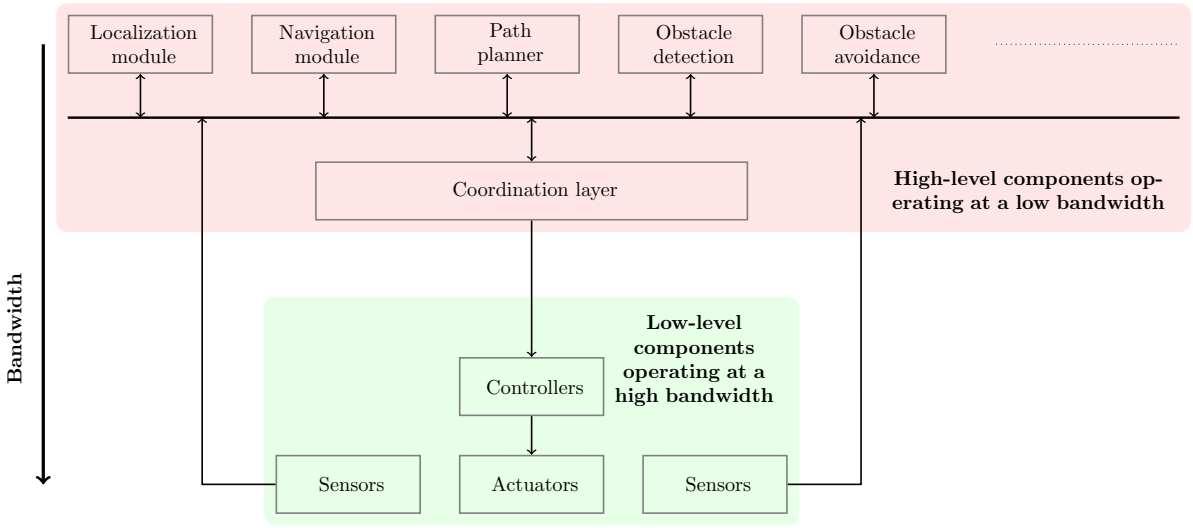


Figure 1.1: Different modules of mobile robotic systems.

1.2 Major Issues

There are three fundamental problems in mobile robotics. The first problem is simply to answer the question "Where am I?" which refers to finding the pose of the robot in world-coordinate system. This problem is commonly referred to as the *localization problem*. The second problem can be encapsulated by the question "How do I move from one point to another?" which can be termed as the *path planning/path following/trajectory tracking problem*. Finally, the third problem is expressed by the question "How do I build the operating environment?" commonly referred to as the *mapping* problem [Borenstein et al., 1997, Rekleitis, 2003]. Clearly, the first and third problems are interrelated because when an accurate map of the environment exists then the robot can localize itself based on the observation of the environment. On the other hand, when the robot knows its current position with high accuracy, it can build a map of its surroundings based on the observed features. The process of solving these two problems simultaneously is termed as *Simultaneous Localization And Mapping (SLAM)*. This thesis mainly deals with finding the solution of the second problem using a mobile robot with the exception that obstacle avoidance strategies are not considered.

In general, developing motion control algorithms for generating accurate states of a robot may be quite complex. This complexity stems from the fact that the desired mapping from measurements to control actions is unknown to the robot. In fact, what is known to the robot is its desired motion trajectory/path specified by users. The desired behavior must somehow be represented through an unknown mapping of measurements and control actions [Argall, 2009]. The challenge is how to develop a control strategy that provides accurate mapping of measurements and robot's control actions so that the robot is able to follow the desired behavior. In addition to developing appropriate control algorithms, there is a number of significant technical difficulties that include the following:

- The robot observes its environment through sensors. In most cases, these sensory meas-

urements are noisy which are often inconsistent with the robot's state and may sometimes provide misleading information.

- The mathematical models of a robot (kinematic or dynamic models) are approximations of the nominal ones and are often simplified due to computational or memory constraints. These models thus may inaccurately predict motion effects [Argall, 2009].

- The robot's control actions are executed by its actuators which depend on tuning many physical parameters yielding motion behavior with some level of imprecision.

For the robot to operate in a real world environment, it has to take into account the above technical difficulties. Hence, the ultimate goal is to trim out the difference between the desired and actual motion behavior of a robot. In most cases, researchers rely on the robot's mathematical models to develop the control policies. Undoubtedly, these mathematical models are theoretically sound but at the expense of accuracy when applied to a real robot due to structured and unstructured uncertainties. Moreover, linearization of the robot model is often introduced for computational tractability, thereby degrading the performance of the robot's motion behavior. Taking into account the aforementioned technical difficulties, robot's motion control schemes basically provide control actions to its actuator at a variety of control levels [Argall, 2009].

- **Low-level control actions:** these actions are normally continuous-valued and motion control strategies that generate low level control actions are sampled at a very high frequency. Robot's actuators are directly controlled by theses actions.

- **High-level control actions:** in this case, low-level control actions are encoded to a more abstract representation, which is then translated through other means (internal controller) to affect the actuators of the robot. High-level control actions are, however, discrete-valued and their associated control strategies are sampled less frequently than low-level ones [Argall, 2009].

In this dissertation, the priority is given to generating high-level control actions to the robot's actuators in order to observe its satisfactory performance on the navigation in its indoor operating environment. The discrete space of control actions, computational complexity, cost-effectiveness, and modularity constitute the key consideration for developing mobile robot navigation strategies.

Having said that, the design and implementation of an efficient navigation method (*i.e.*, guiding a mobile robot from one point to another) that ensures full autonomy with high precision is definitely a challenging task. This requires designing an efficient sensor network, reliable obstacle avoidance strategy, and a motion planner. Typically, wheeled mobile robots operate in environments that are either partially or completely unknown. Often the environment changes with time in an unknown manner; hence, a navigation system that can enable the robot to autonomously navigate in these working environments is well motivated.

1.3 Thesis Contributions

This thesis exploits the possibilities of using RFID technology for mobile robot navigation. It does so by designing and implementing new motion control algorithms using customized RFID systems that are modular and cost-effective. Hence, the primary contributions of this dissertation are twofold. First, a customized RFID architecture is designed and implemented which provides cost-effective feedback information for the robot. Second, a set of novel motion control algorithms are developed for solving the true real-time indoor navigation problem on a differential drive mobile robot. A non-comprehensive list of contributions reported in this dissertation can be sought in [Miah and Gueaieb, 2009a,b, 2010a,b, 2011a,b, 2012a,b,c,d].

1.4 Thesis Outline

The remainder of this thesis is organized as follows.

- Chapter 2 presents the relevant work and sets the framework in which the current research is situated.
- The robot's kinematics and the general problem formulation are studied mathematically in Chapter 3.
- Chapter 4 describes the fundamentals of operating principles of a commercially available RFID system. Moreover, this chapter illustrates how an RFID system can be customized for solving the navigation problem of a mobile robot.
- Then, Chapter 5 illustrates the mobile robot navigation system using a bearing sensitive RFID system, where reverberant effects of the environment is considered in order to realistically assess the performance of the navigation system. A comprehensive simulation results are presented to examine the in depth different aspects of the robot's controllers employed in this thesis.
- An output feedback controller is designed in Chapter 6, where RFID measurements are directly employed to generate necessary control actions for the robot's actuators to track a pre-defined trajectory in the workspace. This approach, however, suffers from a relatively high computational overhead.
- The problems in Chapters 5 and 6 are strongly tackled in Chapter 7 by partitioning the robot's tracking problem into two parts: the nominal (deterministic) and the neighboring solutions. The simulation and experimental results are presented to validate such solutions.
- Finally, some pros and cons of the proposed techniques are highlighted in Chapter 8 along with some thoughts of how they can be further extended and enhanced.

Chapter 2

Literature Review

2.1 Introduction

In this chapter we examine relevant state-of-the-art mobile robot navigation techniques. It is worth mentioning that the subject of navigation covers a large fraction of the research in mobile robotics, thus making a comprehensive survey of the available work is beyond the scope of this thesis.

The fundamental idea behind the problem of mobile robot navigation can be explored taking into account: localization, goal specification, and navigation [Leonard and Durrant-Whyte, 1991]. Localization is the problem of determining the own position in a given environment, based on what can be seen and what information was previously acquired. The goal specification and navigation are essentially the problems of identifying a particular goal in an environment and planning a path that results in achieving this goal.

Among the sensors used for mobile robot positioning and navigation are cameras, range finders using sonar, laser and infrared technology, radar, tactile sensors, compasses, and GPS. Initially, researchers focused on laser and sonar-based navigation of mobile robots. Sonars were used as an intelligent sensor. However, since sonars have very low-bandwidth capabilities,

they are subject to noise due to wave scattering. Hence, the interest was shifted to laser sensors. Although laser sensors have a much higher bandwidth, they are still subject to noise. Moreover, lasers have a limited field-of-view unless intricate mechanics such as rotating mirrors are incorporated in the sensor design. The rest of the sensors can be utilized in some other navigation techniques which also have several drawbacks in performing accurate navigation in unstructured environments.

The most common and popular navigation techniques suggested in the state of the art generally fall under one of the following categories: dead-reckoning-based, landmark-based, vision-based, and behavior-based techniques. The robot navigation problem is also solved by several control laws which are based on conventional control techniques. Each navigation method has its own advantages and disadvantages, although it is difficult to rate them objectively. Some aspects can be unequivocally compared, such as computational complexity, navigation accuracy, and the amount of information a priori required for the proper operation of the algorithm.

Basically, a dead-reckoning navigation system provides position, heading, linear, and angular velocity of an autonomous mobile robot and it is widely used due to its simplicity and easy maintenance. The odometer is the most simplistic implementation means of dead-reckoning which provides a position and an orientation of a mobile robot using onboard encoder information. The fundamental idea behind this navigation technique is the integration of incremental motion over time [Borenstein et al., 1997]. This navigation method is based on continuous encoder readings. The problem of dead-reckoning navigation system is that small precision errors and sensor drifts inevitably leading an increasing cumulative errors in the robot's position and orientation, unless an independent reference is used periodically to correct the error [Ojeda et al., 2006].

Given the shortcomings of dead-reckoning-based navigation, researchers became more in-

terested in the use of landmark-based navigation systems where landmarks are placed at various locations in the environment. These landmarks hold some sort of information which are used by the robot in order to better estimate its position in the environment. However, a landmark-based navigation strategy relies on identification and subsequent recognition of distinct features or objects in the environment that may be a priori known or extracted dynamically. Due to sensors noise and possible dynamic changes of the operating environment, the recognition process of features or objects might become quite challenging. To resolve these issues, some researchers shifted their interest to vision-based navigation systems. Vision sensors can have wide field-of-view, millisecond sampling rates, and can be easily used for trajectory planning. However, some disadvantages of vision include the lack of depth information, image occlusion, and the requirement for extensive data interpretation (recognition), which usually computationally involved and time consuming. Given the advantages and disadvantages of the various sensors, some initial work targeted the use of data fusion to merge various sensors to build a map of the robot environment [Hebert and Kanade, 1988, Thorpe et al., 1988]. To alleviate some of the problems associated to image processing based navigation, some researchers focused on implementing navigation algorithms taking advantages of some salient features of the emerging RFID technology to localize (not to navigate) mobile robots [Gharpure et al., 2006]. Most RFID-based mobile robot navigation systems, suggested in the state of the art to date, RFID tags attached in the environment and an RFID reader mounted on the mobile robot. The RFID reader continuously reads the tag IDs from its operating region and sends them to the robot. The robot then makes the decision about its position based on the tag IDs collected from the surroundings.

As noted above, the literature refers to the navigation problem in conjunction with the localization problem where the central task is to determine a robot's position. In that case, the problem of estimating where objects lie in the environment and the problem of locating the

robot have to be solved in conjunction. Some autonomous robot navigation problems have also been solved using conventional controllers that depend on complex mathematical models of sensor and actuator systems [Apolloni and Moise, 2000, Shahidi et al., 1991, Wane and Motamed, 1994]. This makes it quite difficult to control the robot using conventional control approaches due to the high complexity of the robot dynamics.

In the following, we survey the developments of the autonomous navigation of wheeled mobile robots in the last two-decades. The evolution of robot navigation techniques is presented based on their categories described above. Note that these categories are not exclusive, rather, some navigation strategies may fall in more than one categories.

2.2 Dead-reckoning-based Navigation

Dead-reckoning navigation method is a simple mathematical procedure for determining the current location by knowing the previous position through known odometric information and velocity over a given period of time. Most of the dead-reckoning-based mobile robot navigation algorithms use a combination of odometry, and landmark or vision systems.

In [Makela and Koskinen, 1991], the authors aimed at studying dead-reckoning and visual landmark recognition as a solution for navigating a vehicle along a predetermined path in forest environment. In this work, a magnetic compass for heading measurement and a ground speed Doppler radar for distance measurement were chosen to realize dead-reckoning navigation system. The path in the forest was marked with landmarks which are detected by the camera connected to the computer of the vehicle. The position of the vehicle is determined based on the position of the detected landmarks. The location of the vehicle is also calibrated based on the dead-reckoning information and direction of the landmarks from the vehicle. The fusion of the sensory data from the dead-reckoning system and the landmark direction system is performed through a Kalman filter algorithm.

Authors in [D'Orazio et al., 1993] described a strategy and control architecture to allow a mobile robot to navigate in indoor environments on a planned path. The navigation system of the mobile robot integrates the position estimation obtained by a vision system with the position estimated by odometry, using a Kalman filter framework. Obstacle detection is performed through several ultrasonic sensors. This system is suitable for a structured or quasi-structured environment and needs a-priori knowledge of the world model. This a-priori knowledge is designed by means of a computer-aided design (CAD) system. A CAD description is provided to define an initial setup of the world in order to plan the path in advance.

The indoor autonomous goal-based mobile robot navigation using a cooperative strategy of odometry and a visual self-localization method is addressed in [Stella et al., 1994]. The odometer used in this research is a dead-reckoning sensor which can estimate only relative motion, so to compute the absolute position of the vehicle. The position and orientation of the vehicle is further calibrated using a standard CCD camera. Alternatively, a combination of dead-reckoning and gyroscope can also be used to estimate vehicle's position and orientation as in [Park et al., 1996], for example. In there, an indirect Kalman filter is applied for the reliable position and heading angle estimation in an autonomous mobile robot navigation system. In this system, the encoder and gyroscope fusion algorithm using indirect Kalman filter algorithm requires the preprocessing of the encoder information. At first, a simple encoder based navigation is developed, and then a navigation error model is derived through linear perturbation. The indirect Kalman filter is realized by applying Kalman filter to the navigation error model. Finally, the indirect Kalman filter feeds back the error estimates to the main navigation algorithm. A further extension of this work is proposed in [Park et al., 1997] which is based on a differential encoder and a gyroscope, where an AUTOGYRO (Fiber Optical Gyroscope) was used to measure the angular velocity. In this research, encoder errors are calibrated using a gyroscope. The indirect Kalman filter was adopted for the estimation of encoder systematic errors and gyroscope errors.

The problem of navigation of an autonomous guided vehicle (AGV) was addressed in [Adam et al., 1999]. The approach proposed in this navigation system again relies on fusing the odometry and the information provided by a vision system. A point-able camera was utilized to focus on a point in the environment while the AGV is navigating. The coordinates of the landmark are not assumed to be a priori known. The camera is responsible for finding the coordinates of the landmark. The difference of [Adam et al., 1999] over previous navigation methods is that any point in the observed scene by the camera can be selected as a landmark and not just pre-measured point. In this method, the fusion of odometry and camera measurements is also performed using a Kalman filter for better estimation capabilities.

A more precise navigation system for a 4WS mobile robot is proposed in [He et al., 2002], where the position and orientation estimation is based on GPS and encoders. The estimator consists of two Extended Kalman Filters, Runga-Kutta-based dead-reckoning unit and an Arbitrator, to estimate the position and orientation of the vehicle. An outdoor mobile robot navigation based on DGPS and odometry data fusion was presented in [Ohno et al., 2003] where the position of the robot is fundamentally determined from odometry information. The DGPS measurement data is used to calibrate the estimated position obtained from odometry information. Authors in [Meng and Bischoff, 2005] have proposed a trigonometric method for the calculation of position and orientation based on odometric information. Their platform focused only on the localization problem for autonomous mobile robot navigation. This model was derived to deliver timely and accurate odometric information from measurements of drive wheels' revolutions and steering angles. Recently, in addition to dead-reckoning and GPS, a map matching technique is also incorporated for providing improved localization accuracy for mobile robots operating in urban environments [Lee et al., 2011]. In this case, GPS measurements are validated according to the map. Based on the assumption that the robot always operates on the road, the GPS measurement which fall outside the road can be judged as the incorrect

measurement. Therefore, that erroneous measurement will be discarded and will not be used in the EKF. Furthermore, the map matching method is combined with the Mahanolobis distance approach to increase the robustness of the localization system.

It is clear that the dead-reckoning-based navigation systems are primarily focused on estimating robot's positions rather than navigating itself in an environment. These techniques can only provide good short-term accuracy, since it accumulates position and orientation errors if navigation is attempted over long distances. Despite these limitations, most of the researchers agree that odometry is an important part of mobile robot navigation due to its simplicity. Due to the inevitable accumulating error associated to dead-reckoning, additional correcting/calibrating methods are quite necessary.

2.3 Landmark-based Navigation

Basically, landmarks are distinct features that a robot can recognize in its workspace through its sensors. Landmarks can have geometric shapes such as rectangles, lines and circles, and may even include additional information (*e.g.*, in the form of a bar-code). In general, landmarks have fixed and known positions, relative to which a robot can localize itself. Landmarks are carefully chosen to be easily identifiable. For example, there must be sufficient contrast with the background. Before a robot can use landmarks for navigation, the characteristics of the landmark must be known and stored in the robot's memory. The main task in localization is then to recognize the landmarks and to calculate the robot's position with respect to them.

Landmark-based navigation is intended to greatly improve robot position estimation over dead-reckoning by tracking visual features in the environment and using them as landmarks. This measurement returns bearing to the visual feature only. Landmark-based navigation can be classified in the following two main sub-categories:

- Artificial landmark-based navigation and
- Natural landmark-based navigation.

The main contributions in these categories are highlighted in the following subsections.

2.3.1 Artificial Landmark-based Navigation

Artificial landmarks are specially designed objects or markers that need to be placed in the environment with the sole purpose of enabling robot navigation. Vision plays a crucial role in such navigation systems. Ultimately, interaction with unknown, complex, dynamic environment requires some visual sensory data processing techniques. Hence, artificial landmark-based robot navigations systems often rely on cameras, some image processing techniques, and pattern recognition methodologies. In recent years, a significant research has been conducted in robotics that incorporate several sensors and landmarks (artificial and natural) as navigation media in an operating environment.

2.3.1.1 Navigation based on Landmark's Geometry/Color

Kotani *et al* in [Kotani et al., 1998] described a navigation system for an autonomous mobile robot in outdoor environments which uses vision to detect landmarks and DGPS information to determine the robot's initial position and orientation. The vision system is responsible for detecting landmarks in the environment by referring to an environmental model. As the robot moves, it estimates its position by conventional dead-reckoning, and matches up the landmarks with the environmental model in order to reduce the error in the robot's position estimation. Navigation using single-camera vision and ultrasonic sensors has been studied in [Ohya et al., 1998] where self-localization of the robot is done using a model-based vision system, and nonstop navigation is realized by a retroactive position correction system. For self-localization,

the robot first renders an expectation image using its current best estimate of its present location and then the model edges extracted from the expectation image are compared and matched with the edges extracted from the camera image through an extended Kalman filter. The Kalman filter automatically then yields updated values for the location and the orientation of the robot.

Authors in [Abe et al., 1999] have developed a vision-based navigation system where global matching is defined as the behavior that is able to relocate the position of the robot which is calculated by locating the corners in the workspace. Murray and Little [Murray and Little, 2000] focused on building occupancy grid maps of the environment and presented a method for reducing stereo vision disparity images to two-dimensional map information. The robot can navigate and autonomously explore the environment using this information. However, building a map of the environment is a complex task. To alleviate this complexity, a novel qualitative vision-based navigation algorithm is proposed in [Chen and Birchfield, 2006] where the algorithm is entirely qualitative in nature, requiring no map of the environment, image Jacobian, homography, fundamental matrix, nor assumption about a flat ground plane. In this teach-reply approach, the robot is manually led along a desired path in a teaching phase, then it automatically follows that path in a reply phase. This method requires a single off-the-shelf, forward-looking camera with no calibration.

Hallmann and Siemiatkowska [Hallmann and Siemiatkowska, 2001] used a mobile robot to navigate in a partially known environment. In there, the map of the robot's environment is represented as grids of cells. Circular shaped landmarks, two vertical parallel strips and H-shaped landmarks are considered as the artificial landmarks which help the robot's self-localization in the environment while navigating. The images of these artificial landmarks are taken using a gray-scale camera and then some image processing techniques are applied for the recognition of the landmarks in the environment for accurate positioning. The path of the mobile robot is determined using a diffusion method. The indoor mobile robot navigation

presented in [Yi and Choi, 2004] uses a global ultrasonic system for locating the robot. An extended Kalman filter is opted to process the sensory data to locate the robot.

Extensive research works on vision systems have been conducted by Merhy *et al* and Payeur for different applications including mobile robot navigation [Merhy et al., 2006, 2008, Payeur, 2002]. In some cases, occupancy grid maps are processed as textured images in order to extract meaningful information which is required for mobile robot navigation. A robot localization system using biologically inspired vision system is proposed in [Siagian and Itti, 2009]. Human visual capabilities are modeled through gist features and salient regions which are then further processed using a Monte Carlo localization algorithm to allow the robot to generate its position. Yet, they are based on known noise statistics and constrained on specific lighting conditions. A survey on vision-based navigation systems is presented by Dsouza in [DeSouza and Kak, 2002]. Several methods are suggested to improve the robot position estimation by tracing visual features in the environment and using them as landmarks. These measurements usually return bearing to the visual features only, with no a priori knowledge of the landmark positions. Many vision-based pose-estimation schemes in robot control rely on an extended Kalman lter (EKF) that requires tuning of lter parameters. To obtain satisfactory results, EKF-based techniques rely on "known" noise statistics, initial object pose, and sufciently high sampling rates for good approximation of measurement-function linearization. To alleviate that, iterative adaptive EKF (IAEKF), is proposed by integrating mechanisms for noise adaptation and iterative-measurement linearization [Janabi-Sharifi and Marey, 2010]. This method demonstrated the superiority of IAEKF in dealing with erroneous a priori statistics, poor pose initialization, variations in the sampling rate, and trajectory dynamics. A recent research on monocular vision-based simultaneous localization and mapping (SLAM) of a mobile robot using an upward-looking camera is proposed in [Hwang and Song, 2011]. This approach is opted to use corner, lamp, and door features simultaneously to achieve stable navigation in various environments. Although

a monocular camera looking up toward the ceiling can provide a low-cost solution to indoor SLAM, this strategy is often unable to achieve dependable navigation due to a lack of reliable visual features on the ceiling.

Based on the aforementioned contributions of artificial landmark-based navigation methods, it can clearly be noticed that vision sensors are of a paramount importance for such navigation systems. The features extracted from vision systems provide significant improvement on estimating robot's state. In addition, in the case of noisy information from vision systems, they can easily be fed through conventional filtering algorithms for the robot to estimate it's postures with a reasonably high precision in a particular operating environment. As such, vision-based mobile robot navigation methods have received a considerable amount of attention to the robotic community. However, it is important to articulate the fact that vision-based navigation techniques often suffer from some significant disadvantages, such as lack of information depth, complex image processing algorithms with high computational complexity, and its dependence on the working environment. Such navigation schemes are very sensitive to the lighting conditions of the robot's workspace. Moreover, a dedicated hardware is often required for processing vision features which are extracted for navigating the robot. Having said that, much attention is given to RFID-based robot navigation systems where RFID systems are used as artificial landmarks. A non-comprehensive record of mobile robot navigation techniques using RFID technology is given below.

2.3.1.2 RFID-based Robot Localization/Navigation

RFID technology drew the attention of a large body of research on mobile robot localization owing to its wide availability, affordability, non-touch recognition system that transmits and processes the information on events and environments using a wireless frequency and small chips [Hahnel et al., 2004]. Since an RFID system can recognize signals at high-speed and send

data within various distances, the application of RFID technology has become popular in the robot technology recently [Kulyukin et al., 2004]. In general, this technology is now capable of solving the localization problem efficiently.

Khubitz *et al* presented a navigation system that uses RFID tags as artificial landmarks [Kubitz et al., 1997]. The tags' global position, environment class, environment position, and further optional data are pre-stored in the tags' memory. The system also employs a behavior-based control architecture which enables the robot to reach any landmark within its working environment through a topological robot positioning approach. The behavior-based control architecture is specially designed to be able to integrate several positioning sensors with different accuracies and error categories while enabling the robot to navigate. A robot navigation system in man-made environments, such as hallways, was developed in [Tsukiyama, 2001], where RFID tags are used as artificial landmarks and the mobile robot is equipped with an on-board laptop computer, an RFID tag sensor, and a vision system. The RFID reader is mounted on the robot itself while the tags are pasted at particular locations on walls. At the junction of two passages, the RFID tag sensor reads the unique tag identification numbers and infers the necessary actions (turn left, right, or remain straight) to reach the desired positions. Authors in [Kantor and Singh, 2002, Kurth et al., 2003] have investigated the application of RFID systems for mapping relying on active beacons which provide distance information based on the time required to receive the response from an RFID tag. Besides that, RFID tags have also been used for mobile robot navigation for the visually impaired [Kulyukin et al., 2004]. In this case, passive RFID tags deployed in the environment act as reliable stimuli that trigger local navigation behavior to achieve global navigation objectives.

In [Lionel et al., 2004], the authors developed an indoor location sensing prototype which can be used for various mobile commercial applications. This suggested prototype uses RFID technology for locating objects inside buildings. Hahnel et al. tried to improve the localization

with a pair of RFID antennas [Hahnel et al., 2004]. They presented a probabilistic measurement model for RFID readers that allow them to accurately localize the RFID tags in the environment. Another technique was proposed by Tsukiyama [Tsukiyama, 2005], where the robot tries to build a topological map of its surrounding environment to be used in path planning and navigation. Each node in the topological map is the intersection point of two passages. At these points, the robot has to decide on the next action according to a plan stored in the robot's memory to reach the target position. The robot then follows certain paths using an ultrasonic range finder until a tag is found. However, such a methodology is specific to a particular workspace and requires a substantial amount of customization for it to operate in a new environment. Chae et al. proposed a mobile robot localization method with the help of a combination of RFID and vision [Heesung and Kyuseo, 2005]. The global localization of the robot is performed by incorporating signal detection from artificial landmarks represented by RFID tags. The tags are assigned different weights which are determined by the RFID reader mounted on the robot. The algorithm takes advantage of a vision system incorporating a feature descriptor derived from a scene view of the robot environment, which provides the fine position and orientation of the robot. Although this algorithm offers an efficient localization method, it naturally inherits the typical shortcomings of vision-based techniques in general.

It is clear from the aforementioned methods that the advent of RFID and RFID systems [Nasri et al., 2008, Peris-Lopez et al., 2006], and their applications in the field of robotics [Milella et al., 2007], positioning systems have been used to deliver location information in indoor and/or outdoor environments. The primary role of such localization systems is to estimate and report geographical information pertaining to the data processing unit associated with a mobile robot for the purpose of management, enhancement, and personalization services. While most researchers employed RFID systems for solving localization problem of mobile robots in a particular environment [Han et al., 2007, Jing and Yang, 2007, Kodaka et al., 2008,

Luo et al., 2007]; in 2007, Miah and Gueaieb have turned the attention to use a customized RFID system to navigate (not just to localize) a indoor mobile robot using phase-difference method [Miah, 2007]. The working principle of such a robot navigation system is that the reader of an RFID system is mounted on the robot and the tags (or transponders) are attached at different locations as artificial landmarks. The robot's actuator signals are generated based on the phase-difference information received through an RFID reader mounted on it. In 2009, Park and Hashimoto proposed an algorithm for estimating pose for mobile robot navigation using a passive RFID system. In there, the robot's pose is basically estimated using trigonometric functions and the IC tags' Cartesian coordinates which are arranged in a regular gridlike pattern in its operating environment. Recently, robot's localization accuracy is further improved by fusing information from ultrasonic sensors and RFID tags which are distributed in space [Choi et al., 2011]. For that, a global position estimation process is defined using an RFID system and a local environment cognition process is defined using ultrasonic sensors. Then, a hierarchical localization algorithm is proposed to estimate the position of the mobile robot using these processes.

In addition to conventional RFID-based position estimation systems reported above, they can also be classified as range-based and bearing-based [Kim and Chong, 2009]. The main idea behind the range-based technique is to trilaterate the robot's position using some known reference points and the estimated distances at those points in the environment. Distances can be estimated from either received signal strength (RSS) measurements or time-based methods. Although a small subset of such works have explored the use of Time of Flight (ToF) [Lanzisera et al., 2006] or Time Difference of Arrival (TDoA) measurements [Ni et al., 2003], RSS is generally the feature of choice for indoor positioning. This is due to the fact that RSS measurements can be obtained relatively effortlessly and inexpensively. In addition, no extra hardware (e.g., ultrasonic or infra-red) is needed for network-centric localization [Youssef,

2004]. On the other hand, bearing-based schemes use the direction of arrival (DoA) of a target. However, these schemes require multiple range sensors in order to be better suited for mobile robot applications [Kim and Chong, 2009].

2.3.2 Natural Landmark-based Navigation

In addition to artificial landmarks, natural landmarks have also been exploited in a number of robot navigation algorithms. Borenstein et al. in [Borenstein et al., 1996] defined natural landmarks as those objects or features that are already in the environment and have a function other than robot navigation. There are so many examples of natural landmarks: trees, lampposts, furniture, lamps and so on. These landmarks are usually chosen depending on the environment (e.g. indoor or outdoor). The main challenge becomes to extract those landmarks from the entire scene in a robust way in order to generate a spatial representation. In the context of natural landmark-based mobile robot navigation, the representation of the landmarks in the environment is a crucial point. Geometric modeling of landmarks can be used to deal with two fundamental tasks: landmark extraction, and recognition for sensor-based motion control or robot localization. However, geometrical representations can lead to a bulky model and after some iteration, to a combinatorial explosion.

For instance, Betge-Brezetz et al. [Betge-Brezetz et al., 1994] focused on the high level representation of the natural scene to guide a mobile robot in a priori unknown environment. The landmarks in this case are defined as natural objects extracted from perceptual data. The scene is structured into elements corresponding to its main entities and only the parametric description is employed to characterize the shape of every entity. A segmentation algorithm has been adopted to distinguish different components in the 3-dimensional scene. After that, the object models are built using a quadratic representation. Finally, the objects and the topological models are merged to construct the scene model which is ultimately used for the navigation

control. Wijk and Christensen developed a similar algorithm for natural landmark extraction from sonar data streamed from a mobile platform [Wijk and Christensen, 2000]. In this work, the robot's absolute position is determined through a matching procedure between the recently collected landmarks and the reference map. The natural point landmark extraction method adopted consists of a double-fold filtering process of sonar data: a triangulation-based fusion and a completion of the landmark hypothesis. In the first layer, two-dimensional data points are filtered out and the best triangulation points from the first filtering stage are considered in the second layer. Then, in the second layer, these extracted landmark points are used to match the reference map in order to localize the robot in its working environment.

Some researchers proposed mobile robot navigation methods using lines marked on the floor [Drake et al., 1988, McVey et al., 1986]. A continuous, straight-edged line is employed for the visual navigation of an autonomous mobile robot. An intelligent line-tracking navigation and a policy control method assist a mobile robot rolling along an indicated line on the ground, to ensure it automatically navigates home after leaving the line and finishing its tasks. Line navigation can be thought of as a continuous landmark, although in most cases the sensor used in this system needs to be very close to the line, so that the range of the vehicle is limited to the immediate vicinity of the line. These techniques have been used for many years in industrial automation tasks and vehicles. However, the techniques are not discussed in detail here since they do not allow the vehicle to move freely - the main feature that sets mobile robots apart from AGVs.

The recognition process of features or objects in landmark-based navigation system might be too complex in dynamically varying environments. A simple solution to this issue is addressed in [Hu and Gu, 1999] where the mobile robot is utilized to operate in the manufacturing industry. In this research, the navigation algorithm is capable of locating the robot and updating the landmarks in a dynamic manufacturing environment. An Extended Kalman Filter (EKF)

algorithm is adopted in the navigation system to improve the localization accuracy of the mobile robot. Both artificial and natural landmarks are deployed in order to provide useful solution toward real-world applications. The position of the mobile robot is initialized autonomously and recalibrated using a Kohonen neural network. This navigation algorithm can be applied in a mobile robot to be applicable at home, office and hospitals where many features are available.

While landmark-based mobile robot navigation systems offer significant advantages over odometry-based techniques, such as improved global navigation accuracy, no data-association problem [Choset et al., 2004], etc., they are quite challenging when the robot is deployed in a reverberant environment with high signal distortions.

2.4 Behavior/Soft Computing-based Navigation

As the development of different autonomous robot navigation techniques in real-world environments constitutes one of the major trends in current research in robotics, an important problem is the need to cope with the large amount of uncertainty inherited from natural environments. As such, soft computing techniques have received a considerable attention due to their relative robustness to uncertainties. Numerous navigation techniques have been suggested in the state of the art using some tools of computational intelligence such as fuzzy logic, neural network, neuro-fuzzy system, genetic algorithm, and several combinations of them.

The natural appeal of fuzzy logic to robot navigation has motivated several researchers in this area [Lee et al., 2005, Saffiotti, 1997]. An alternate direction of work is presented in [Parhi, 2005, Peterson, 1981] where fuzzy logic and Petri Net models were employed for the navigation control of multiple mobile robots in a cluttered environment. In there, a fuzzy logic system was embedded in the controller of the robot to avoid static obstacles other than the robots. The inputs of the fuzzy logic system were the line-of-sight distances of obstacles around the robot and the bearing of the robot's target. These inputs are, however, are modeled using the

data dispatched from ultrasonic sensors and infrared sensors, respectively. The output was the steering angle that needs to be applied on the wheels to guide itself in the proper direction. The function of the Petri Net model is to avoid collision among robots operating in the environment. Although, fuzzy logic proved to be effective in coping with uncertain environments, it has some limitations. In a purely fuzzy system, the parameters cannot be optimized in an analytical way. Therefore, they cannot be applied for learning and tuning fuzzy rules. On the other hand, neural networks can easily map empirical training set through learning. As a result, they have been successfully applied in several research bodies in robotics [Lagoudakis and Maida, 1999, Pham and Parhi, 2003]. Nevertheless, the mapping rules in the network are not visible and are difficult to understand. A control law illustrated in [Li and Chang, 2003] uses the fuzzy logic for a car-like mobile robot which has autonomous garage-parking and parallel-parking capability by using real-time image processing technique. Sheng *et al* proposed a controller with two parameters to stabilize a nonholonomic mobile robot with two driving wheels, and find out its optimal parameters using the genetic algorithm to minimize the given performance criterion function [Sheng et al., 2004]. Authors in [Hsu and Liu, 2007] designed a flexible architecture for a mobile robot, called Virtual Operator MultiAgent System, in order to satisfy the rapidly changing missions by dynamic task switch or dynamic role switch. More specifically, a virtual operator and a robot agent work together to control a robot to fulfill a specific task. Each task is represented by a virtual operator and the robot agent handles reactive control. Based on this architecture, multiagent system can perform the dynamic task switch to handle missions, which are not initially given.

In addition to fuzzy logic controllers, neural network (NN)-based control architectures have also been popular for mobile robot navigation in indoor/outdoor environment [Das et al., 2006, Jolly et al., 2009, Ye, 2008]. This is due to the fact that NN-based methods have their inherent parallelism and learning capabilities for solving highly nonlinear systems. Lately, Mo-

hareri *et al* proposed a novel adaptive trajectory tracking controller for a nonholonomic mobile robot with unknown parameters and uncertain dynamics [Mohareri et al., 2012]. Kinematic controller gains are tuned on-line to minimize the velocity error and improve the trajectory tracking characteristics. Navigation problem of mobile robot has also been tackled by combining both fuzzy logic and neural network systems in a neuro-fuzzy system [Rusu et al., 2003]. For example, the control scheme proposed in [Su et al., 2010/08/] employs fuzzy inference as a main controller and the neural network as an auxiliary controller. In the fuzzy controller, the translation width and total sliding surface are adopted to reduce the chattering phenomena. The neural uncertainty observer is added in the balance, speed and synchronous controllers to reduce the accumulated error and ascend the stability.

An important observation that can be made in this category of mobile robot navigation methods is that the precise mathematical model of the robot itself does not have to be known a-priori. However, a significant amount of parameters need to be tuned off-line (in most cases) in order to achieve an acceptable level of navigation accuracy.

2.5 Conventional Nonlinear Control Schemes

This section illustrates some existing techniques for the problem of mobile robot navigation using conventional nonlinear control schemes. In this context, conventional techniques refer to mathematical methods which do not use any sort of computational intelligence tools, such as fuzzy logic, neural netwok, genetic algorithms, and swarm intelligence.

Feedback control design problems for tracking a pre-defined trajectory or stabilizing to a fixed point using a nonholonomic robot are quite challenging tasks. The tracking problem is divided into two sub-problems: state tracking and output tracking, that deal with the stabilization of system outputs or states to any reference output or desired state (especially an equilibrium point) [Cervantes and Alvarez-Ramirez, 2001, Kim et al., 2005, Luo et al., 2005, Wei et al., 2010].

The stabilization problem has been extensively studied for both linear and nonlinear systems due to its simplicity. It is worth noting that, the complexities of stabilization and tracking problems of nonholonomic robot are inversely proportional, which means that stabilizing a wheeled robot to a fixed configurations is more difficult than path following or trajectory tracking [Luca et al., 2000]. In particular, Brockett's theorem [Brockett, 1983] proves the nonexistence of smooth state-feedbacks for the asymptotic stabilization of fixed configurations. The detailed description of motion control issues of nonholonomic robots can be sought in [de Wit et al., 1993, Luca et al., 2000].

Kanayama *et al* pioneered a stable tracking control rule for non-holonomic vehicles. Stability of the rule is proved through the use of a Lyapunov function [Kanayama et al., 1990]. The robot model is linearized to decide parameters for critical dumping for a small disturbance. In order to avoid any slippage, a velocity/acceleration limitation scheme is also introduced. Some nonlinear control laws have been proposed in [Fontes, 2003, Keerthi and Gilbert, 1988], where basic optimal control theory has been used to solve control problems of a general class of nonlinear systems. Jiang *et al* addressed the tracking problem for a class of nonholonomic chained form control systems [Jiang and Nijmeijer, 1999]. A recursive technique is proposed which appears to be an extension of the currently popular integrator backstepping idea to the tracking of nonholonomic control systems. Conditions are given under which the problems of semiglobal tracking and global path-following are solved for a nonholonomic system in chained form and its dynamic extension. Authors in [Aguiar and Hespanha, 2007] addressed the problem of position trajectory-tracking and path-following control design for underactuated autonomous vehicles in the presence of possibly large modeling parametric uncertainty. They demonstrated how adaptive switching supervisory control can be combined with a nonlinear Lyapunov-based tracking control law to solve the problem of global boundedness and convergence of the position tracking error to a neighborhood of the origin that can be made arbitrarily small.

A simple adaptive control approach for path tracking of uncertain nonholonomic mobile robots incorporating actuator dynamics is proposed in [Park et al., 2010]. All parameters of robot kinematics, robot dynamics, and actuator dynamics are assumed to be uncertain. For the simple controller design, the dynamic surface control methodology is applied and extended to mobile robots that the number of inputs and outputs is different.

Authors in [Tarin et al., 2000] illustrated an optimal feedback position control law which consists of controlling the posture of an autonomous mobile robot in order to reach a specified goal posture using the velocities of translation and rotation as inputs. The contribution claimed by the authors is to propose a time varying feedback control law that minimizes the input energy for solving the optimal feedback control problem for high precision positioning. The solution to the problem of combined trajectory tracking and path following for marine craft is presented in [Encarnacao and Pascoal, 2001]. The methodology is opted to maneuver trajectory tracking for vehicles with non negligible dynamics and is also applied to a fully actuated underwater vehicle. However, the model parametric uncertainty and external disturbances are not taken intro consideration for devising the control mechanism.

Unfortunately, there are not many optimal control strategies that have been proposed in the literature to solve the stabilization and tracking problems of nonholonomic mobile robots in a unified manner. Even though, some research works have been conducted to propose nonlinear control laws for solving only tracking problems, but less attention is paid towards solving such problems since the tracking problem is simpler than point stabilization problem for nonholonomic systems [Luca et al., 2000]. Authors in [Gu and Hu, 2006] proposed a Receding Horizon (RH) open-loop optimization control method to tackle these control tasks for mobile robots. Their tracking performance is smooth and satisfactory for complex reference trajectories (circle, eight-shaped, for example). However, it suffers from choosing the appropriate control time horizon that is dependent on the complexity of the reference trajectories.

Some nonlinear control strategies have specifically been designed to tackle the problem of point stabilization only [Aicardi et al., 1995, Cao et al., 2010, Pomet, 1992]. Authors in [Jiang et al., 2001, Jiang and Nijmeijer, 1997] achieved a global tracking control with respect to arbitrary initial tracking errors using backstepping methods. The desired linear and angular velocities of the robot were constrained in those techniques. These constraints were eliminated in [Lee et al., 2001], which, nevertheless, requires the adjustment of a control parameter that depends on the reference trajectories, and a persistently excited control input. Motion control problem of wheeled mobile robots in obstacle-free environments is detailed in [Oriolo et al., 2002]. In there, a dynamic feedback linearization tool is used which led to a solution that is simultaneously valid for both trajectory tracking and setpoint regulation problems. It is worth noting that a high resolution robot hardware is required to achieve a desired performance. In addition, the controller's performance might degrade if the robot is deployed for tracking a longer trajectory. These shortcomings are strongly tackled by Chwa in [Chwa, 2004], which is mainly based on two sliding surfaces in polar coordinates in terms of tracking errors in both position and heading direction. This technique is also dependent on the model parameters and comes short of considering actuator constraints which are required for the actual implementation. Authors in [Chen et al., 2009] provide a solution to the wheeled mobile robot tracking problem in the simultaneous presence of control constraints and external disturbances. An $\mathcal{H}_\infty$ tracking performance for disturbance attenuation is achieved despite the actuator saturations. This work is, however, based on linearizing the robot model around the reference trajectory. In [Chwa, 2010], Chwa again proposed an excellent tracking control method for differential-drive wheeled mobile robots by using a backstepping-like feedback linearization, which might not work for some highly nonlinear systems. This is due to the fact that the full-state or input-output linearizations may not exist for such systems. The trajectory tracking problem for nonholonomic mobile robots has recently been tackled by the transverse function approach [Morin and Samson,

2009]. A salient feature of this approach is the obtention of feedback laws that unconditionally achieve the practical stabilization of arbitrary reference trajectories, including fixed points and nonadmissible trajectories. However, this approach requires comprehensive tuning of transverse function parameters. Authors in [Sun et al., 2009] present a synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations. The main idea is to control each robot to track its desired trajectory while synchronizing its motion with those of other robots to keep relative kinematics relationships, as required by the formation. An excellent vector-field-orientation feedback control method for a differentially driven wheeled vehicle has been demonstrated in [Michalek and Kozowski, 2010]. This technique solves both trajectory tracking and point stabilization problem as in our current work. The dynamic effects of the vehicle and the noisy feedback signal may affect the vehicle to stabilize on a fixed configuration. Note that Lizarrage in [Lizarrage, 2004] states that for systems with nonholonomic constraints frequently found in control applications, if the control inputs contain reference functions that are not persistently excited, then the "universal stabilization" problem cannot be solved, even if time-varying feedback is used.

A large body of research work has been conducted on model predictive control that relies on the solution of an open-loop optimal control problem to predict the system behavior over a time horizon [Allgower et al., 1999, de Oliveira Kothare and Morari, 2000, Diehl et al., 2000, Graichen and Kugi, 2010, Mayne et al., 2000, Rawlings and Mayne, 2009]. A sliding mode controller in cooperation with RFID system is proposed in [Lee et al., 2009] to track a desired trajectory, where RFID tags are placed on the floor in a triangular pattern to estimate the position of the mobile robot. This technique, however, is not suitable if the operating environment is dynamically changed. Recently, some researchers in control theory have focused on design and analysis of controllers for Lipschitzian nonlinear systems and a strategy for observer design has been developed for such systems [Rajamani, 1998]. In order to address

stabilization performance, robustness, actuator fault tolerance and disturbance attenuation of dynamic systems, LMI-based linear state feedback has been used extensively [Sun and Liu, 2007]. Such techniques are based on a nonlinearity assumption, such that $f(x) = 0$ at $x = 0$, for example. Although this makes problem handling easier, the issue of stabilizing of classes of systems not satisfying this assumption still stands as a challenge [Rehan et al., 2011].

2.6 Summary

In this chapter we presented relevant background of our work. Several advantages and disadvantages of each mobile robot navigation technique described above are also illustrated. Dead-reckoning navigation techniques are simple but accumulate error over time during the navigation of a mobile robot in its working environment. Landmarks and vision systems are appropriate for robot localization. However, they are heavily dependent on efficient image processing, feature extraction, and data interpretation algorithms. In most cases, these tasks require additional dedicated hardware (sophisticated processor, memory card, data acquisition card, etc.) Furthermore, image occlusion, low resolution, and unfavorable lighting conditions, may significantly limit the robots maneuverability. Behavior-based navigation techniques require extensive tuning of controllers' parameters if the robot's operating environment is dynamically changing. The conventional control techniques for guiding a mobile robot provide satisfactory performance for small variations of external disturbances. Nevertheless, they may not be applicable for highly unstructured and dynamic environments due to lack of proper modelling of control laws in such environments.

Chapter 3

Kinematics of a Differential Drive Mobile Robot and Problem Formulation

3.1 Introduction

Theorem Kinematics is the basic study of the mechanical behaviour of a mobile robot. In recent years, the robotics community has achieved a fairly complete understanding of the kinematics and even the dynamics (*i.e.*, relating to forces and inertia) of robot manipulators [Craig, 1989, Sciavicco and Siciliano, 1996]. In general, the mobile robotics community poses many of the same kinematic questions as the robot manipulator community. However, a key factor that mainly distinguishes between a mobile robot and a manipulator arm is their *position estimation*. Measuring the position of an arm's end effector is simply a matter of understanding the kinematics of the robot and measuring the position of all intermediate joints. On the other hand, a mobile robot is a self-contained automation that can wholly move with respect to its operating environment [Siegwart and Nourbakhsh, 2002].

This chapter outlines some theoretical models that may be employed to represent a mobile robot and its environment in order to solve the problem of mobile robot navigation. Some

preliminary mathematical conventions and properties are provided in section 3.2. It then begins with the process of understanding the kinematics of a unicycle robot system followed by a differential drive wheeled mobile robot (DDMR) by describing the contribution of each wheel for motion. Each wheel of a DDMR has a role in enabling the whole robot to move forward or backward subject to its nonholonomic constraints. These are illustrated in detail in section 3.3. The general navigation problem for a DDMR is formulated in section 3.4 followed by the summary in section 3.5.

Throughout this thesis, small and capital bold letters will be used to denote vectors and matrices, respectively. Scalars are denoted by non-bold letters.

3.2 Preliminaries

The 2-norm and scalar product are defined by

$$\|\mathbf{x}\| \equiv \left[\sum_{i=1}^{n} |x_i|^2\right]^{1/2} \text{ and } (\mathbf{x}\cdot\mathbf{y}) \equiv \mathbf{x}^T\mathbf{y} \equiv \sum_{i=1}^{n} x_i y_i,$$

respectively, for vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and positive n. For matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m\times n}$, these quatities are given by

$$\|\mathbf{X}\| \equiv \left[\sum_{i=1}^{m}\sum_{j=1}^{n} |x_{i,j}|^2\right]^{1/2} \text{ and } (\mathbf{X}\cdot\mathbf{Y}) \equiv Tr\left[\mathbf{X}^T\mathbf{Y}\right] \equiv Tr\left[\mathbf{X}\mathbf{Y}^T\right],$$

respectively, where $Tr[\cdot]$ denotes the trace of matrix $[\cdot]$. Clearly, $Tr\left[\mathbf{X}^T\mathbf{X}\right] = \|\mathbf{X}\|^2$.

If the function $J : \mathbb{R}^n \to \mathbb{R}$ is differentiable at $\mathbf{x} \in \mathbb{R}^n$, then for any $\boldsymbol{\nu} \in \mathbb{R}^n$, $\boldsymbol{\nu}^T dJ(\mathbf{x})$ denotes the Gateaux (directional) derivative in the direction of $\boldsymbol{\nu}$, which is given by

$$dJ(\mathbf{x};\boldsymbol{\nu}) = \lim_{\varepsilon\to 0} \frac{J(\mathbf{x}+\varepsilon\boldsymbol{\nu}) - J(\mathbf{x})}{\varepsilon}.$$

However, if $J : \mathbb{R}^{m\times n} \to \mathbb{R}$, then for any $\mathbf{X}, \mathbf{V} \in \mathbb{R}^{m\times n}$, the directional derivative in the direction of $\mathbf{V}$ is defined by

$$dJ(\mathbf{X};\mathbf{V}) \equiv Tr[\mathbf{V}^T dJ(\mathbf{X})] = \lim_{\varepsilon\to 0} \frac{J(\mathbf{X}+\varepsilon\mathbf{V}) - J(\mathbf{X})}{\varepsilon}.$$

For any bounded interval $I \equiv [t_0, t_f]$, $C(I, \mathbb{R}^n)$ denotes the class of all continuous functions on I taking values in $\mathbb{R}^n$. Let $p \in [1, \infty)$ and any finite time interval I, we use $\mathcal{L}_p(I, \mathbb{R}^n)$ to denote the set of Lebesgue measurable functions $\{\mathbf{f}\}$ defined on the measurable set I and taking values in $\mathbb{R}^n$ whose norms are p-th power integrable [Khalil, 2002, Royden and Fitzpatrick, 2010] *i.e.*,

$$L_p(\mathbf{f}) = \left(\int_{t_0}^{t_f} \|\mathbf{f}\|^p dt\right)^{1/p} < \infty,$$

where $L_p(\mathbf{f})$ denotes the p-th norm of the function $\mathbf{f}$. For $p = \infty$, $\mathcal{L}_\infty(I, R^n)$ denotes the space of Lebesgue measurable functions $\{\mathbf{f}\}$ defined on I and taking values in $\mathbb{R}^n$ satisfying

$$\text{ess-sup}\{\|\mathbf{f}(t)\|, t \in I\} < \infty.$$

Similarly, $\mathcal{L}_p^{\ell oc}([0, \infty), \mathbb{R}^n)$ are locally convex topological vector spaces of p-th power locally integrable functions containing the spaces $\mathcal{L}_p(I, \mathbb{R}^n)$. We use $diag(a_1, \ldots, a_n)$ to represent a diagonal matrix whose entries are $a_1, \ldots, a_n$ starting from the upper left corner.

3.3 Robot's Kinematic Model and Constraints

Deriving a model for the whole robot's motion is a bottom-up process [Siegwart and Nourbakhsh, 2002]. Each individual wheel contributes to the robot's motion and, at the same time imposes contraints on robot motion. Wheels are tied together by the robot chassis geometry, and therefore their contraints combine to form contraints on the overall motion of the robot chassis. This is

particularly important in mobile robotics because of its self-contained and mobile nature. We begin by representing the position of a DDMR, then using the resulting formalism to annotate the kinematics of individual wheels and the whole robot.

3.3.1 Representing the Robot's Pose

The robot's model considered throughout this dissertation is a rigid body operating on wheels in a 2-D horizontal plane. The total dimensionality of this robot chassis on the plane is three, two for position in the horizontal plane and one for orientation along the vertical axis, which is orthogonal to the plane. For simplicity, the robot's pose (position and orientation) is represented in global reference frame $X - Y$ as shown in Figure 3.1. The robot's position is the midpoint of the wheelbase of length l connecting the two lateral wheels along their axis. Let (x, y) be the coordinate of the position taking values from $\mathbb{R}^2$ and the orientation (angular position) is denoted by θ. Note that the robot's orientation is defined on the unit circle $\mathbb{S}^1$, *i.e.*, $\theta \in [-\pi, \pi) \equiv \mathbb{S}^1$. In this dissertation, $\mathbf{q} \in \mathcal{Q}$ denotes the pose (or configuration) of the robot in the configuration space $\mathcal{Q} \subset \mathbb{R}^2 \times \mathbb{S}^1$. Therefore, the robot's pose can now be described as a vector of three elements *i.e.*,

$$\mathbf{q} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}.$$

3.3.2 Contraints of a DDMR

Consider a unicycle (human-powered, single-track vehicle with one wheel) whose contact pose on a horizontal plane is given by $\mathbf{q} = [x \;\; y \;\; \theta]^T$. If the linear velocity of the unicycle is denoted by ν, then the velocity components along the $X - Y$ frame are given by

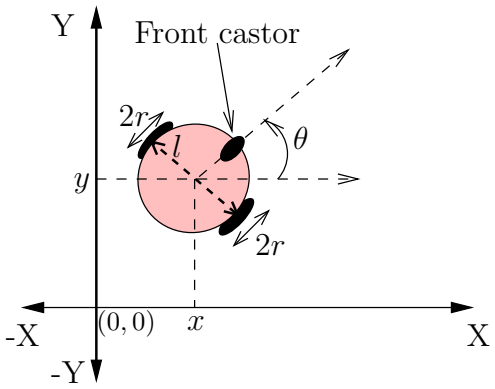


Figure 3.1: Robot's position in global reference frame.

$$\dot{x} = \frac{dx}{dt} = \nu \cos\theta$$
$$\dot{y} = \frac{dy}{dt} = \nu \sin\theta.$$

Clearly, these expressions assume that the unicycle's sideway translation is prevented by the no-slip constraint (nonholonomic) imposed by the wheel, which is given by

$$\frac{\dot{x}}{\dot{y}} = \frac{\cos\theta}{\sin\theta} \Rightarrow -\dot{x}\sin\theta + \dot{y}\cos\theta = 0. \tag{3.1}$$

In other words, the velocity vector of the unicycle is orthogonal to the vector $[-\sin\theta \quad \cos\theta]^T$ as shown in Figure 3.2. The nonholonomic constraints of a DDMR can now be derived using the unicycle's constraint which is given by the model (3.1). This is illustrated using Lemma 3.1.

Lemma 3.1 (nonholonomic constraints of DDMRs)**.** *Consider* $\mathbf{q} = [x \quad y \quad \theta]$ *to be the pose of a DDMR, where* (x, y) *is the midpoint of the line connecting the two lateral wheels of the robot and* θ *is its orientation with respect to the* X*-axis in global reference frame. Then the nonholonomic constraint can be written in the form*

$$\mathbf{w}(\mathbf{q})\dot{\mathbf{q}} = 0, \tag{3.2}$$

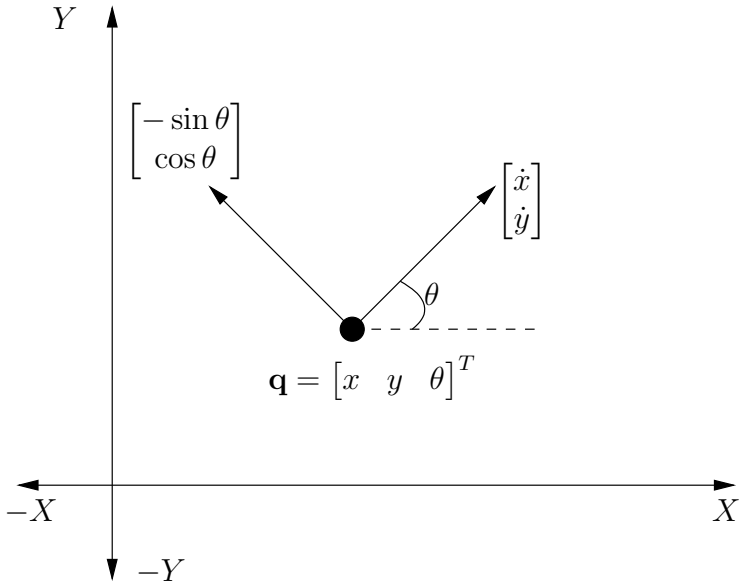


Figure 3.2: The unicycle robot system.

where

$$\mathbf{w}(\mathbf{q}) = [\sin\theta \quad -\cos\theta \quad 0] \qquad \textit{and}$$
$$\dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt} = [\dot{x} \;\; \dot{y} \;\; \dot{\theta}]^T.$$

The detail proof of this Lemma is given in Appendix A.2. The nonholonomic constraint (3.2) will be used to derive the kinematic model of a DDMR.

3.3.3 Kinematic Model of a DDMR

The kinematic model of a DDMR will be derived based on the general formulation of a unicycle kinematic model given in [Choset et al., 2004, Chapter 12]. Let $\mathcal{G}$ be a set of vector fields, and let $span(\mathcal{G})$ be the linear span of vector fields in $\mathcal{G}$, given by all linear combinations of vector fields in $\mathcal{G}$. For $\mathbf{q} \in \mathcal{Q}$, $\mathcal{D}(\mathbf{q})$ is the distribution which assigns a subspace of the tangent space $\mathcal{T}_q\mathcal{Q}$ *i.e.*,

$$\mathcal{D}(\mathbf{q}) = \{\dot{q} \in \mathcal{T}_\mathbf{q}\mathcal{Q} \,|\mathbf{w}(\mathbf{q})\dot{\mathbf{q}} = 0\}.$$

Note that the row vector $\mathbf{w}(\mathbf{q})$ is called a *co-vector* and lives in the co-tangent space $\mathcal{T}^*_{\mathbf{q}}\mathcal{Q} = \mathbb{R}^n$, where n is the dimension of the configuration space ($n = 3$ in the case of a mobile robot). If $\mathcal{W}(\mathbf{q})$ is the co-distribution that assigns the subspace of the co-tangent space $\mathcal{T}^*_{\mathbf{q}}\mathcal{Q}$, then a *co-vector field* $\mathbf{w}(\mathbf{q}) \in \mathcal{W}(\mathbf{q})$ pairs with a vector field $\hat{\mathbf{g}}(\mathbf{q}) \in \mathcal{G}$ to yield a real value, $\mathbf{w}(\mathbf{q})\hat{\mathbf{g}}(\mathbf{q}) \in \mathbb{R}$. In the case of a mobile robot, $\mathcal{W}(\mathbf{q})$ and $\mathcal{D}(\mathbf{q})$ must be orthogonal to each other, $\mathcal{W}(\mathbf{q}) \perp \mathcal{D}(\mathbf{q})$. Hence,

$$\mathbf{w}(\mathbf{q})\hat{\mathbf{g}}(\mathbf{q}) = 0. \tag{3.3}$$

Since a DDMR has one nonholonomic constraint given in (3.2), the dimension of the distribution $\mathcal{D}(\mathbf{q})$ is $3 - 1 = 2$. The distribution $\mathcal{D}(\mathbf{q})$ can now be expressed as

$$\mathcal{D}(\mathbf{q}) = \dot{\mathbf{q}} = span\left(\hat{\mathbf{g}}_{\mathbf{1}}(\mathbf{q}), \hat{\mathbf{g}}_{\mathbf{2}}(\mathbf{q})\right) = \hat{\mathbf{g}}_{\mathbf{1}}(\mathbf{q})\nu_1 + \hat{\mathbf{g}}_{\mathbf{2}}(\mathbf{q})\nu_2, \tag{3.4}$$

which is the kinematic equation of motion of a DDMR with $\nu_1, \nu_2 \in \mathbb{R}$ being the linear and angular velocities of the robot, and $\|\hat{\mathbf{g}}_m\| \neq 0$, for $m = 1, 2$. The unknown vector fields $\hat{\mathbf{g}}_{\mathbf{1}}(\mathbf{q})$ and $\hat{\mathbf{g}}_{\mathbf{2}}(\mathbf{q})$ of (3.4) can be determined by the model (3.3). Clearly, $\hat{\mathbf{g}}_{\mathbf{1}}(\mathbf{q}) = [\cos\theta \;\; \sin\theta \;\; 0]^T$ and $\hat{\mathbf{g}}_{\mathbf{2}}(\mathbf{q}) = [0 \;\; 0 \;\; \alpha]^T$, $\alpha \in \mathbb{R}$, satisfy Equation (3.3) yielding the kinematic model (3.4) of the form

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} \nu_1 + \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix} \nu_2 = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix}.$$

If ν_R and ν_L denote the left and right wheels' linear velocities, then

$$\dot{x}_R = \nu_R \cos\theta,$$
$$\dot{y}_R = \nu_R \sin\theta,$$
$$\dot{x}_L = \nu_L \cos\theta,$$
$$\dot{y}_L = \nu_L \sin\theta.$$

Substituting these expressions in the expressions of the velocity components given in (A.2) yields the solutions for $\dot{x},\ \dot{y}$, and $\dot{\theta}$ as

$$\begin{aligned} \dot{x} &= \frac{1}{2}(\nu_R + \nu_L)\cos\theta \\ \dot{y} &= \frac{1}{2}(\nu_R + \nu_L)\sin\theta, \qquad \text{and} \\ \dot{\theta} &= \frac{1}{l}(\nu_R - \nu_L). \end{aligned}$$

In matrix form, the above equations are used to describe the kinematic equations of motion of a DDMR as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\cos\theta & \frac{1}{2}\cos\theta \\ \frac{1}{2}\sin\theta & \frac{1}{2}\sin\theta \\ \frac{1}{l} & -\frac{1}{l} \end{bmatrix} \begin{bmatrix} \nu_R \\ \nu_L \end{bmatrix}. \tag{3.5}$$

If r is the radius of each wheel, then robot's left and right wheels' angular velocities are $u_R = \nu_R/r$ and $u_L = \nu_L/r$, respectively, yielding

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \frac{r}{2} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \frac{2}{l} & -\frac{2}{l} \end{bmatrix} \begin{bmatrix} u_R \\ u_L \end{bmatrix}. \tag{3.6}$$

A compact form of (3.6) can be written as

$$\dot{\mathbf{q}}(t) = \mathbf{f}[\mathbf{q}(t), \mathbf{u}(t)] = \frac{r}{2}\mathbf{B}[\mathbf{q}(t)]\mathbf{u}(t) \ , \tag{3.7}$$

where the robot's configuration $\mathbf{q}(t) \equiv [x(t) \ \ y(t) \ \ \theta(t)]^T \in \mathbb{R}^2 \times \mathbb{S}^1$, its control input vector $\mathbf{u}(t) \equiv [u_R(t) \ u_L(t)]^T \in \mathbb{R}^2$,

$$\mathbf{B}[\mathbf{q}(t)] = \begin{bmatrix} \cos\theta(t) & \cos\theta(t) \\ \sin\theta(t) & \sin\theta(t) \\ \frac{2}{l} & -\frac{2}{l} \end{bmatrix},$$

and $\mathbf{f} : \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2 \times \mathbb{S}^1$. Due to actuators physical limitations, u_R and u_L are constrained as

$$|u_L| \leq u_L^{max} \text{ and } |u_R| \leq u_R^{max}, \tag{3.8}$$

where u_R^{max} and u_L^{max} are the maximum possible speeds of the right and left wheels, respectively. Hence, $\mathbf{u}(t)$ is employed as the control input vector to the robot's actuators and it must be chosen from an admissible set $\mathcal{U}_{ad}$. It is clear from (3.7) that this DDMR is an underactuated system in which the number of control inputs is less than the number of degrees of freedom. That is, there are two control inputs (u_R, u_L) and three degrees of freedom (x, y, θ) to be controlled.

In order to test and validate the motion control strategies of this thesis, the kinematic model (3.7) is realized by the Scout robot platform as pictured in 3.3, which is commercialized by Dr Robot Inc.[1]. Scout is a two-wheel differential drive mobile robot whose kinematic model is derived by the conventional geometric model of a unicycle robot. The details of the low-level components, such as amount of torque-speed characteristics, interfacing mechanisms, etc; are proprietary information of the Dr. Robot Inc. and are essentially unknown to us. The Scout

[1] http://www.drrobot.com/

Figure 3.3: Scout mobile robot from Dr Robot Inc.

robot has the following mechanical and control features which are taken from Dr. Robot Inc.:

- Two 12V motors with over 300oz.-inch(22kg.cm) torque each
- 7 inch driving wheel
- Max speed of 0.75 m/sec
- Dimensions: 38.0 cm (15 in.) diameter min 30cm (12 in.) max 50 (20 in.)
- Weight: 7 kg
- Additional carrying payload: 6 kg
- Pre-programmed fine speed and position control achieved by an integrated PMS5005 module.

Throughout this thesis, simulations are based on the kinematic parameters of the Scout robot. Hence, the chasis diameter $l = 30\,\mathrm{cm}$ and the wheel radius $r = 8.25\,\mathrm{cm}$.

3.4 General Problem Formulation

Let $\mathbf{q^d}(t)$ be the desired continuous path that the robot is supposed to follow at time $t \in [t_0, t_f] \equiv I$, where t_0 and t_f are the initial and final time to travel the whole path, respectively, and I denotes the time interval from t_0 to t_f. The corresponding actual pose of the robot is $\mathbf{q}(t)$ for $t \in I$. Suppose that the control actions applied by the robot's actuator are the speeds $\mathbf{u}(t) \in \mathcal{U}_{\mathbf{ad}}$, which can be generated by its actual pose $\mathbf{q}(t)$, estimated pose $\hat{\mathbf{q}}(t)$, or some sensory measurements $\mathbf{z}(t) \in \mathbb{R}^s$ with s being the number of sensors. In general, this control action can be defined mathematically as

$$\mathbf{u}(t) = \begin{cases} \chi[\mathbf{q(t)}, \hat{\mathbf{q}}(\mathbf{t}), \mathbf{z(t)}] & \text{if } \mathbf{q}(t) \neq \mathbf{q}^d(t_f) \text{ and} \\ 0 & \text{if } \mathbf{q}(t_f) \approx \mathbf{q}^d(t_f), \end{cases} \tag{3.9}$$

where $\chi[\cdot]$ is a linear or nonlinear functional, $\mathbf{q}(t_f)$ and $\mathbf{q}^d(t_f)$ are the actual and desired final poses at time t_f, respectively.

The action $\mathbf{u}(t) = 0$ reflects the fact that the robot has approximately reached its final target point at time $t = t_f$. This is when $\|\mathbf{q}_f(t_f) - \mathbf{q}_f^d(t_f)\| \leq \epsilon$ for a small threshold number ϵ. From model (3.6), the robot's actual trajectory can be described by

$$\mathbf{q}(t) = \mathbf{q}(t_0) + \int_{t_0}^{t} \mathbf{f}[\mathbf{q}(\tau), \mathbf{u}(\tau)]d\tau, \tag{3.10}$$

where $\mathbf{q}(t_0)$ is the actual initial pose of the robot and $t \in I$. If the robot's position tracking error is defined as $e(t) = \sqrt{[x^d(t) - x(t)]^2 + [y^d(t) - y(t)]^2}$ with $(x^d(t), y^d(t))$ being the desired 2-D position trajectory, then the objective is to find the control input $\mathbf{u}(t) \in \mathcal{U}_{ad}$ that generates the trajectory $\mathbf{q}(t)$ as in (3.10) while minimizing the total position tracking error, $\mathscr{E}$, given by

$$\mathscr{E} = \int_{t_0}^{t_f} e(t)dt. \tag{3.11}$$

Given the robot's kinematic model (3.7) and its nonholonimic constraint (3.2), the problem can be stated as follows:

$$\mathbf{u}(t) = \underset{\mathbf{u}(t)\in\mathcal{U}_{ad}}{\arg\min}\ [\mathscr{E}]. \tag{3.12}$$

These control inputs $\mathbf{u}(t)$ generate the robot's actual trajectory $\mathbf{q}(t) \in \mathcal{Q}$, for $t \in I$.

3.5 Summary

In this chapter, the kinematic model of a DDMR system is derived from the nonholonomic constraints acting on it. The nonholonomic constraints of a DDMR ensure that there is no slipping and movement in its lateral direction. In other words, these constraints prevents the robot's velocity orthogonal to its orientation. The robot's actuator control inputs can be generated based on some sensory information. The control inputs are, however, determined taking into account the actuators' physical limitations. For the robot to navigate along a geometric path with an associated timing law, its main objective is to determine suitable control inputs based on some sensory information available to it. In this thesis, a customized RFID system is considered to be a significant source of sensory information available to the robot during its mission.

Chapter 4

Overview of RFID Systems

4.1 Introduction

Radio Frequency IDentification (RFID) is simply a technology that uses radio frequency (RF) electromagnetic fields to transfer (wireless, non-contact) data between a group of devices called tags (or transponders) and other devices called readers (or interrogators). Because of the simplicity, an RFID system has been used for many years in various remote sensing applications, specifically in access control and animal tracking applications. Nowadays, in order to track animals, RFID chips are extremely small devices injected via syringe under skin. RFID systems are being used in some hospitals to track a patient's location, and to provide real-time tracking of the location of doctors and nurses in the hospital. In retail stores, RFID offers a real-time inventory tracking and allows companies to monitor and control inventory supply at all the time. It is worth noting that the RFID technology is recently being applied in service robots [Kampke et al., 2012, Peng and Dong, 2012]. However, some salient features of RFID systems may lead to promising alternatives to existing RFID-based robot navigation strategies. The primary focus of this thesis is to use a set of customized features of existing RFID systems for solving the navigation problem of mobile robots in indoor environments.

This chapter provides background information of an RFID system which is aimed to understand its applications in mobile robot navigation throughout this thesis. It begins with presenting the benefits of RFID systems relative to other automatic identification and data capture (AIDC) technologies in section 4.2. Section 4.3 introduces the operating frequencies of a typical RFID system followed by its brief operating principle in Section 4.4. Section 4.5 gives the short description of the basic components of RFID systems. A customized RFID tag architecture which is employed in most parts of the thesis is illustrated in section 4.6.2. This is followed by a chapter summary in section 4.7.

4.2 Advantages of RFID Systems

RFID represents a technological advancement in AIDC (Automatic Identification and Data Capture) because it offers advantages that are not available in other AIDC systems such as bar code reading. RFID offers these advantages because it relies on radio frequencies to transmit information rather than light, which is required in AIDC systems. The use of radio frequencies means that RFID can communicate:

- without line of sight, as radio waves can penetrate many materials,
- at greater speed, as a result, many tags can be read quickly, whereas optical technology, for example, often requires time to manually reposition objects so as to make bar codes readable,
- over greater distances, since many radio technologies can transmit and receive signals more effectively than optical technology under most operating conditions.

The ability of the RFID technology to communicate without optical line-of-sight and over greater distances than other AIDC technology further reduces the need for human involvement in

the identification process. The RFID technology often supports other features that bar codes or AIDC technology do not have, such as rewritable memory, security features, and environmental sensors that enable the RFID system to record history of events.

4.3 Operating Frequencies of an RFID System

The frequency at which the tag transmits and receives signals is termed as the operating frequency of an RFID system. An RFID system can operate in several operating frequencies which range from low frequency (LF) to super high frequency (SHF) depending on its applications in an environment. RFID systems operating in the frequency range of 125-134 KHz are classified as LF RFID systems; 13.56 MHz as high frequency (HF) RFID systems; 303 MHz to 2.4 GHz as ultra high frequency (UHF) RFID systems; finally, RFID systems which operate above 2.4 GHz are classified as super high frequency (SHF) RFID systems [Finkenzeller, 2003, Li, 2011]. SHF RFID systems are, however, still under development [Li, 2011]. The rationale behind several operating frequency bands of RFID systems is that one frequency band cannot provide all that is required in terms of read range, environmental factors, data rate, etc. Each of the operating frequency bands provides unique performance characteristics for various usage scenarios [Li, 2011]. In general, as the tag's operating frequency increases, its signals are able to carry more data. As a result, higher frequency readers are also able to read more tags in a given period of time. In addition, an RFID system that operates at an ultra high frequency or microwave frequency is designed to have a longer operating range than low or high frequency range systems.

4.4 Operating Principle

Normally, a commercially available RFID system is composed of three main parts: a reader, a tag, and a host computer. The tag is composed of an antenna coil and a silicon chip that includes a basic modulation circuitry and a non-volatile memory. It is energized by a time-varying electromagnetic RF signal that is transmitted by the reader. When the RF field passes through an antenna coil, there is an AC voltage generated across the coil. This voltage is rectified to result in DC voltage for the device operation. The device becomes functional when the DC voltage reaches a certain threshold. The information stored in the device is then backscattered to the reader. A reader is mainly composed of an RF transceiver and decoding sections. In addition, the reader also includes a RS-232 or USB interface in order to perform serial communication with a host computer attached to it. The RF transceiver section is responsible for transmitting and receiving the RF signal. Data decoding for the received signal is accomplished through a microcontroller. A firmware algorithm is written in such a way to transmit the RF signal, decode the incoming data, and communicate with the host computer.

4.5 Components of an RFID System

RFID systems can be very complex, and their implementations vary greatly across industries and sectors. For discussion purposes in this thesis, we will divide an RFID system into two primary subsystems:

- An RF subsystem, which performs identification and related transactions through wireless communication,
- An enterprise subsystem, which contains computers (or processing elements) running specialized software that can store, process, and analyze data acquired from RF subsystem transactions to make the data useful to a supported business process, and

The following is a description of each of these subsystems. In this thesis, we are mainly interested in illustrating the RF subsystem only.

4.5.1 The RF Subsystem

There are two major components of an RF subsystem:

- RFID tags (also referred to as transponders), which are small electronic devices that can be attached to or incorporated into a product, animal, person, or 3-dimensional places, for the purpose of identification using radio waves. The RFID tag can automatically be read from several meters away and does not have to be in the line of sight of the reader. Each tag has a unique ID number known as the identifier and may also have other features such as memory to store additional data, environmental sensors, and security mechanisms.

- RFID readers, which are devices used to interrogate RFID tags and read radio frequency waves emitted by the tags. These devices are responsible for communicating with the tags wirelessly to identify the item connected with each tag.

Figure 4.1 depicts a simple configuration of an RF subsystem which is basically composed of three simple blocks, a reader, a tag, and a wireless channel [Finkenzeller, 2003]. Both the reader and the tag are two-way radios. Each block is equipped with an antenna and is capable of modulating and demodulating RF signals.

4.5.1.1 RFID Tags

The primary function of an RFID tag is to provide its identifier to a reader, but many types of tags support additional capabilities, such as memory, integration of environmental sensors, security, privacy protection, and so on, which can be valuable for certain business purposes or specific applications.

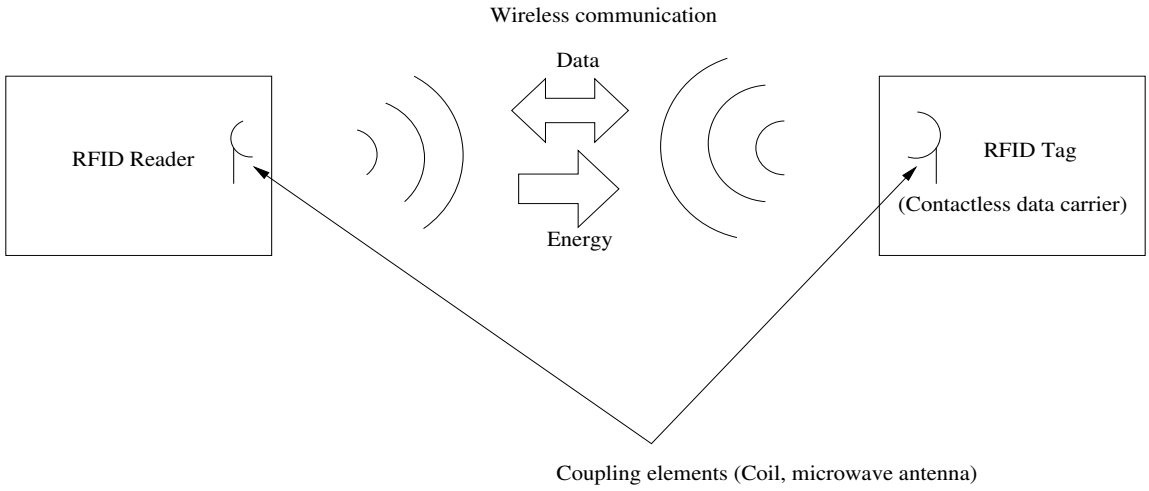


Figure 4.1: A simple RF subsystem configuration.

Tags need power to perform functions such as sending radio signals to a reader, storing and retrieving data, and performing other computations. Tags can obtain this power from a battery or from electromagnetic waves emitted by readers that induce electric current in the tags. The power requirements of a tag depend on several factors including the operating distance between the reader and the tag, the radio frequency being used, and the functionality of the tag. In general, the more complex functions the tag supports, the greater its power requirements. The three main categories of RFID tags which are based on the power source for communication and other functionality are *active* tags, *passive* tags, and *semi-passive*.

Active tags are full-fledged radios, integrating a RF transmitter, battery, receiver and control circuitry [Dobkin and Weigand, 2005]. It does not use backscatter modulation, but generates its own signal and transmits the signal to reader autonomously. On the other hand, passive tags operates without battery. These tags, however, extracts power from the RF signal transmitted by the reader. The semi-passive tag incorporates a battery-assisted-circuit (BAC) to power the tag IC. However the battery is not used to broadcast or amplify a signal to the reader. Like passive tags, it also extracts power of the RF signal from the reader and uses all this power to operate chipset and modulate reader continuous wave signals to backscatter to Reader [Liu et al., 2009].

The communication among passive or semi-passive UHF tags and RFID readers is performed through backscatter modulation [Rao, 1999].

4.5.1.2 RFID Readers

An RFID reader is a device used to interrogate tags. As such, the tag and the reader must comply with the same standards in order to communicate. The characteristics of a reader that are independent of the tag characteristics include: power output and duty cycle, enterprise subsystem interface, mobility, and antenna design and placement. In the following, we provide a short description of these characteristics.

In most cases, standards and regulations determine the power output and duty cycle of the reader. A reader's duty cycle is the percentage of time that the device is emitting energy over a specified amount of time. Readers that communicate with passive tags need greater power output than those that communicate with active tags because the signals must be strong enough to reach the tag and enable the backscatter to return back to the reader. In general, readers with greater power output and duty cycle can read tags quicker more accurately, and within longer distances, but a greater power output also increases the risk of eavesdropping.

In most of the readers, there are two main interfaces, one is the RF interface used to communicate with the tags, and the other is the enterprise subsystem interface which is used to communicate with enterprise subsystems. The enterprise subsystem interface supports the transfer of RFID data from the reader to the enterprise subsystem's computers (or other processing elements) for further processing of data. Usually, the enterprise subsystem interface is employed for a remote management of the RFID reader. This interface may be wired (e.g. Ethernet) or wireless (e.g. Wi-Fi) depending on the application.

As mentioned above, a reader's interface with an enterprise subsystem may be either wired or wireless. Most wired readers are in fixed locations and support applications in which the tags

approach the reader. Some wired readers offer limited mobility using cables. In some cases, the readers are attached to a particular device and the device itself can move around to perform specific tasks. In contrast, wireless readers support applications in which personnel must move around to read tags. A mobile reader usually uses different communication protocols on its RF and enterprise subsystem interfaces, even though both interfaces are wireless.

Readers use a wide variety of antenna types. Each type has a different coverage pattern. To reduce the likelihood of eavesdropping and minimize the interference with other radios, the coverage should only encompass a range that is just sufficient to communicate with the intended tags. Antennas may be integrated into the device or can be detachable. Readers that support detachable antennas are better suited for applications that require specific coverage areas since an antenna can be easily selected or customized to meet those requirements.

The details of communication between RFID tags and readers can be sought in [Miah, 2007].

4.6 RFID System Customization

It is important to note that most of the existing RFID systems available in the market provide only static information which limits its applicability in many real-world proximity-based RFID applications. As such, this thesis provides navigation frameworks for indoor mobile robots based on the following customized architectures of an RFID system:

1. Customized RFID reader architecture

2. Customized RFID tag architecture

Throughout this thesis, an RFID reader is considered to be interfaced with the robot's central processing unit to perform further processing of tags' information.

4.6.1 Customized RFID Reader Architecture

Unlike commercial RFID readers which are bearing insensitive, this reader is equipped with multiple receiving antennas mounted on a circular configuration to enhance the reader's sensitivity to the RF signal's arrival direction. Figure 4.2 shows how this bearing sensitive RFID can be used for mobile robot navigation which will be detailed in Chapter 5.

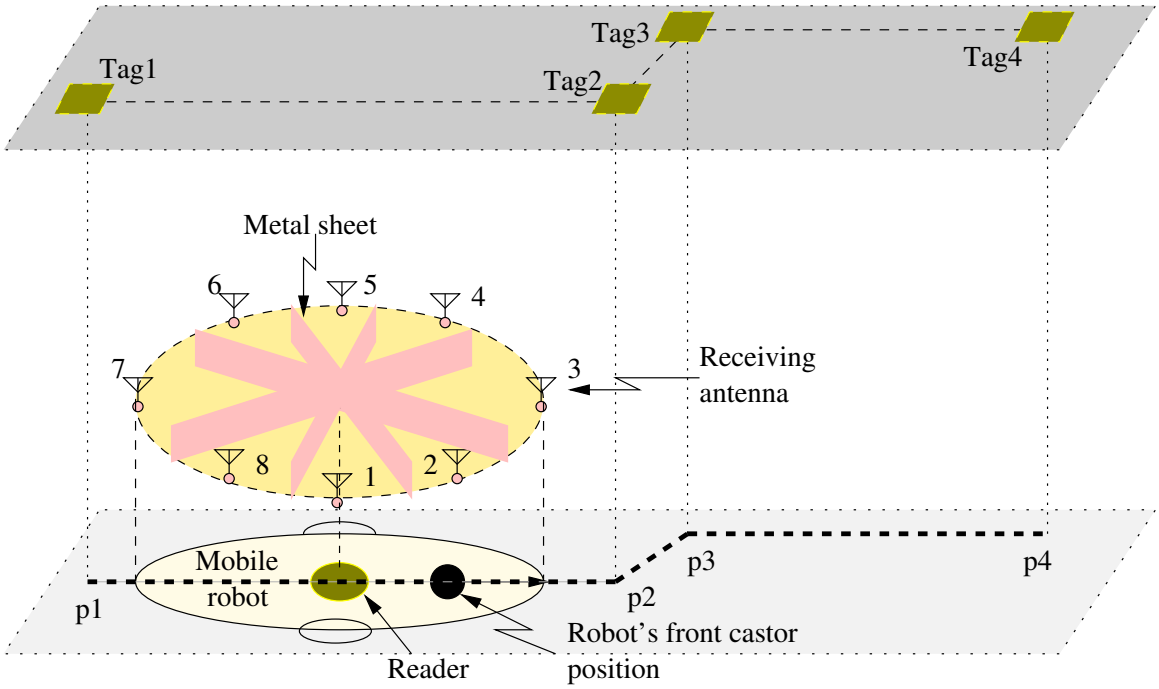


Figure 4.2: High level system architecture of the robot navigation system using bearing sensitive RFID reader.

As can be seen from the high level architecture in Figure 4.2, the proposed RFID reader is comprised of an antenna array where eight antennas are uniformly distributed with angular positions δ_i, $1 \leq i \leq 8$ around a circle of radius r_a centered on a horizontal rectangular metal sheet. Note that this metal sheet (known as ground plane) of dimension $l \times l$ acts as an RF front-end. The relative position of each antenna in the robot's local coordinate system X'-Y' is $(r_a \cos \delta_i, r_a \sin \delta_i)$.

The receiving antennas are interfaced with the reader's back-end through an 8-to-1 mul-

tiplexer and an RSS power detector in addition to its existing analog and digital front-end interface circuits for receiving static binary IDs from the RFID tags, as shown in Figure 4.3. The function of the RSS power detector is to compute the RSS values of the signal received from the tags. The reader can extract RSS values from eight receiving antennas through the multiplexer using the time-division multiple-access (TDMA) technique. The eight RSS values along with the current tag ID are then simply passed to the robot's central processing unit to generate the adequate control action to be dispatched to the wheel actuators.

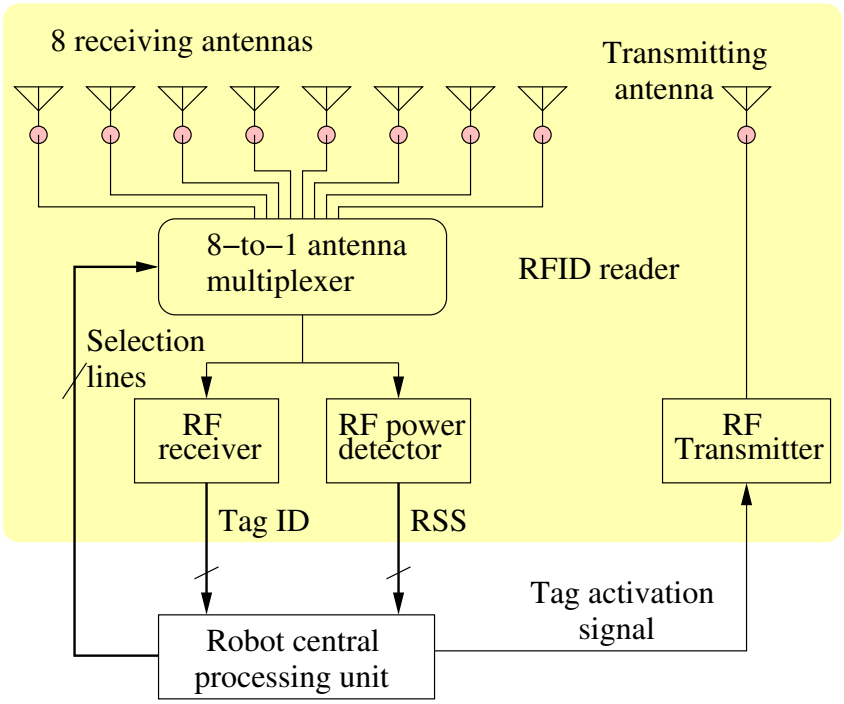


Figure 4.3: Modified RFID reader architecture.

4.6.2 Customized RFID Tag Architecture

As illustrated in section 4.5, an RFID reader is generally composed of an RF module, a control unit, and a coupling element to interrogate electronic tags via RF communications. As a result, the modified tag architecture used in this thesis allows to encode some dynamic information

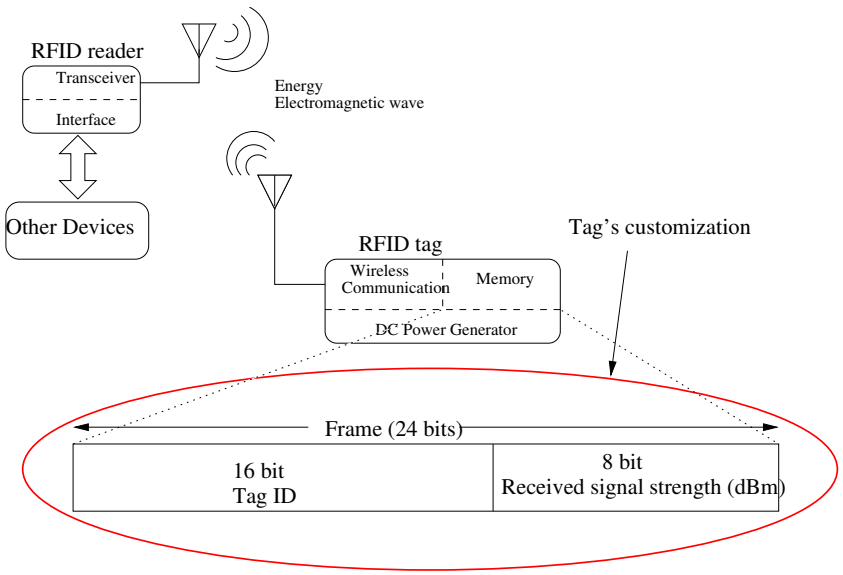
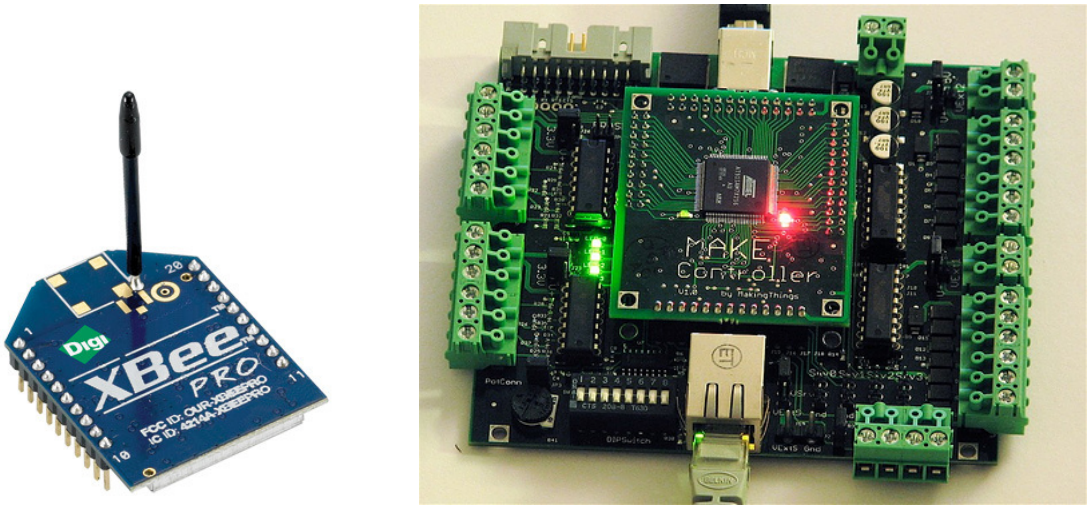


Figure 4.4: Proposed RFID tag architecture.

along with the tag's existing static ID. Figure 4.4 depicts a customized model of an RFID tag. The tag receives an RF signal transmitted by the reader which is then rectified to get its RSS value. In the present RFID system, the tag has some processing capability to convert the RSS value into an 8-bit binary code. As can be seen in Figure 4.4, the RSS measurement of the RFID reader query is embedded with the tag's existing static binary ID (16-bit in this case) which is then backscattered to the RFID reader. It is important to articulate the fact that the reader architecture of the proposed RFID system requires no customization as it would read the 24-bit (16-bit tag-ID + 8-bit RSS) frame in exactly the same way it normally reads tag-IDs. The RFID reader extracts the frame backscattered by the tag which is then passed to the processing element on the robot's board to decode it into a tag-ID and an RSS value. The RSS values are used to described the robot's state with respect to its workspace as will be detailed in the coming chapters.



(a) (b)

Figure 4.5: RFID system emulation using XBee RF module and Makecontroller board (a) RF module used to emulate an RFID tag[1] and (b) Make controller board used to emulate an RFID reader[2].

4.6.2.1 RFID System Emulation

The proposed RFID system with modified tag architecture is emulated using the XBee Pro Modules[1] shown in Figure 4.5(a) as an integrated RF solution. The modules include MC13193 RF chip by freescale, which is compliant to the IEEE 802.15.4 norm [Graefenstein and Bouzouraa, 2008]. One of the XBee Pro modules is attached to the Make Controller (MC) board[2] (Figure 4.5(b)) to emulate a commercial RFID reader. The MC board secures the communication between the emulated RFID reader and the robot.

In order to obtain an RSS value from the tag i, $1 \leq i \leq s$, the reader broadcasts a message with its own static address. The tags are simply configured to reply to the reader's query with their individual binary frames. As mentioned, each tag's frame consists of its 16-bit static address and 8-bit RSS value. The reader simply extracts and decodes the frames in order to get the tag's ID and the corresponding RSS value and then passes them along to the mobile robot for navigation in its operating environment.

[1] http://www.digi.com
[2] http://www.makingthings.com

4.7 Summary

In this chapter, some background information on RFID systems while illustrating their various advantages relative to other similar technologies is presented. A basic configuration of an RFID system consists of two essential components: the tag and the reader, both of which function as two-way radios. In the case of a passive transponder, which is identified by a unique ID, it receives energy from the reader in order to power up its IC memory. It then backscatters the ID to the reader. An important feature that can be taken into consideration is the received signal strength at the tag end which can be embedded with its binary ID. This embedded tag ID can then be backscattered to the reader. In addition to the modified tag architecture, the RFID reader can also be customized for solving mobile robot navigation problems. A bearing sensitive RFID reader architecture is detailed in chapter 5. The rest of the thesis illustrates the application of this customized features of an RFID systems in solving mobile robot navigation problems in indoor environments.

Chapter 5

Mobile Robot Navigation Using Bearing-sensitive RFID Reader

5.1 Introduction

This chapter outlines a novel robot navigation system using a bearing-sensitive RFID reader which is specifically designed to operate in indoor reverberant environments. The operating principle of this bearing-sensitive RFID reader is similar to commercially available RFID readers except that the signal transmitted by the RFID tag is now received by multiple receiving antennas which are mounted on the reader. In addition to the tag ID, the Received Signal Strength (RSS) is also recorded by the RFID reader. This is due to the fact that sufficiently strong RSS measurements may lead to the direction of the transmitting tag. This design does not depend on an accurate mathematical model to map the RSS to an Line-Of-Sight (LOS) distance. The proposed navigation system is evaluated through a series of computer simulations where a commercial electromagnetic simulation software (FEKO) is adopted to realistically assess the reverberations effect on the developed architecture.

Despite the contributions of RFID systems and RSS measurements, the navigation problem

of a mobile robot still poses several significant technical challenges yet to be overcome [Graefenstein and Bouzouraa, 2008]. For instance, the existing antenna configurations in most RFID systems are mostly unable to circumvent the signal reverberation effects inherent in indoor environments. This chapter is devoted to the development of an indoor navigation system where the methods described in the literature might not be suitable. The work described herein is pioneered by mounting a multiple-antenna RFID reader on a mobile robot and placing RFID tags in the 3-dimensional workspace where the tags' orthogonal projection points on the ground define the desired positions that the robot is supposed to reach. As such, the main contributions of this chapter are twofold. First, the direction of the current target point is estimated by the modified RFID reader where RSS measurements along with the current tag ID are extracted from multiple antennas which are mounted on the reader. Second, the problem of true real-time indoor navigation is addressed using a fuzzy logic controller which will be detailed in section 5.2.2.2.

The rest of the chapter is outlined as follows. The different components of the proposed bearing sensitive robot navigation system is described in section 5.2. Section 5.3 details the customization of the RFID reader's architecture and the design of its antenna system to suit the suggested navigation method. A thorough evaluation of the proposed robot control scheme with some numerical simulation results is presented in section 5.4. Finally, the chapter summary is given in section 5.5.

5.2 Proposed Navigation Architecture

In this technique, a mobile robot is pre-programed with an ordered sequence of RFID tag IDs which are attached at unknown 3-D locations in the robot's workspace. The tags' orthogonal projections on the ground then define the ordered sequence of the 2-D points forming the robot's target trajectory. For instance, if the mobile robot is provided with a sequence of four 16-bit

tag IDs, 0xFFF9, 0xFFF2, 0xFFF5, and 0xFFF4, then it is supposed to pass through the points on the ground corresponding to those tag IDs respecting that order. For it to do so, the robot continuously reads the IDs of the tags within its communication range using an RFID reader mounted on it, before extracting and processing the RSS values of the signal coming from the tag corresponding to the robot's destination point at that particular instant. The communication between the tags and the reader is performed through an RFID reader which we specifically designed for this application (see section 4.6.1 for details). A high level setup of the proposed navigation system with four RFID tags attached to the ceiling of an indoor environment is depicted in Figure 5.1. In this configuration, the robot's mission is to visit points p1, p2, p3, and p4, in that order. In fact, this proposed navigation scheme is applicable for any number of tags defining intermediate points in the robot's path. Such a simple navigation scheme may be useful for indoor service robots like in museums and hospitals, for instance. One of its appealing features, in addition to its simplicity, is that changing the robot's desired path is only a matter of changing the orders of the target RFID tags in the robot's memory. In the following, a detailed description of the proposed navigation system is provided.

5.2.1 Robot Model in Discrete-time

Let us sample the continuous-time robot pose $\mathbf{q}(t)$ at discrete time $t = kT$, where k is the discrete time index, $k = 0, \ldots, k_f$, and T is the sample period (in seconds) with $T > 0$. Note that the time index $k = 0$ corresponds to the initial time t_0 and $k = k_f$ corresponds to the final time t_f. As such, the discrete-time equivalent of the model (3.6) can be derived as

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + T\frac{r}{2} \begin{bmatrix} \cos\theta_{k-1} & \cos\theta_{k-1} \\ \sin\theta_{k-1} & \sin\theta_{k-1} \\ \frac{2}{l} & -\frac{2}{l} \end{bmatrix} \begin{bmatrix} u_{R,k-1} \\ u_{L,k-1} \end{bmatrix}, \tag{5.1}$$

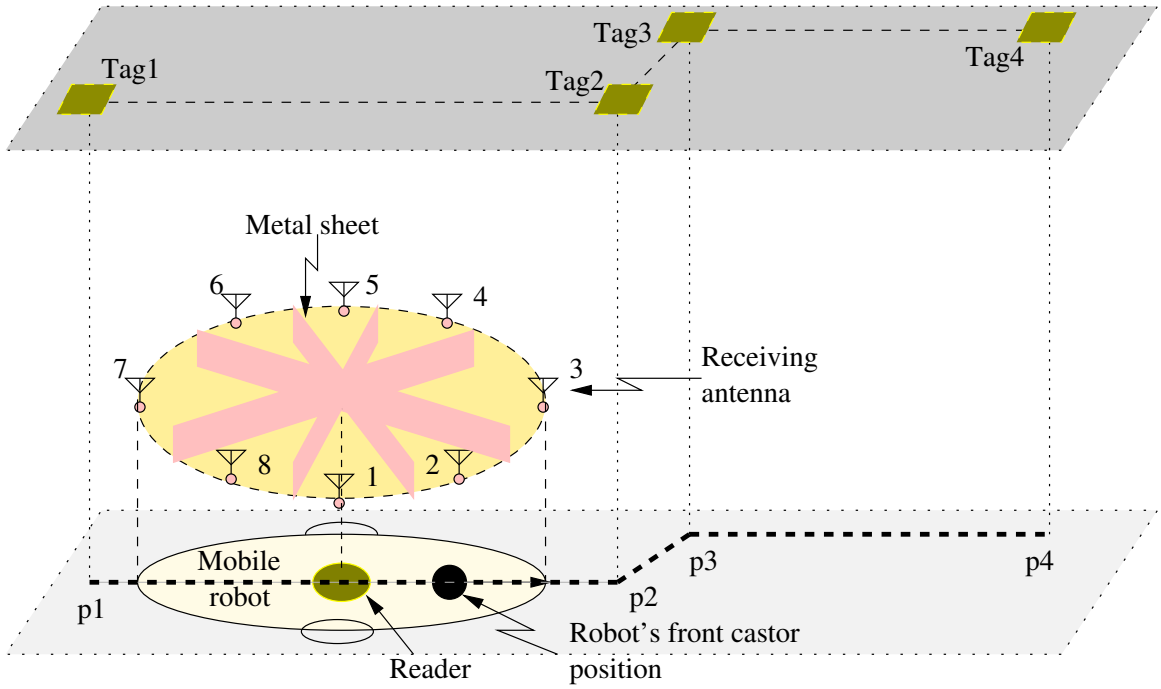


Figure 5.1: High level system architecture of the robot navigation system using bearing sensitive RFID reader.

and its compact form can be written as

$$\mathbf{q}_k = \mathbf{f}(\mathbf{q}_{k-1}, \mathbf{u}_{k-1}),$$

where $\mathbf{q}_k = [x_k \; y_k \; \theta_k]^T \in \mathcal{Q}$ represents the robot's pose and $u_k = [u_{R,k} \; u_{L,k}]^T$ is the control input for its actuator at time step k. It is worth to mention, howerver, that the low sampling frequency and high velocities of the robot can be a significant source of odometric error.

5.2.2 Guiding Principle

Assume $(\cdot)_l$ denotes the quantity $(\cdot)$ in the robot's local coordinate system. The guiding principle of the robot consists of two main stages: (i) approximating the direction of the current target RFID tag (DOCT), $\hat{\gamma}_l$, and (ii) reaching the desired target. A fuzzy logic controller is employed to implement the function $\chi[\cdot]$ in (3.9) for generating the necessary actions to guide the robot

towards the target point. In the following, the two main stages of the proposed navigation strategy are discussed.

5.2.2.1 Stage 1: Tuning the Robot's Direction

Suppose that z_i is the (noisy) RSS value received by antenna i at time t, $i = 1, \ldots, 8$, where z_i^d is the ideal value of that signal (without reverberation nor noise). In other words, $z_i = z_i^d \Gamma(\delta_i) + \zeta_i$, where ζ_i is the noise associated with the RSS value received at antenna i and $\Gamma(\delta_i)$ is the gain of the antenna in the direction of δ_i.

The DOCT at time t is approximated by identifying the antenna with the maximum RSS value. That is,

$$\hat{\gamma}_l = \underset{\delta_i, 1 \leq i \leq 8}{\arg\max}[z_i]. \tag{5.2}$$

Note that the $\hat{\gamma}_l$ can be approximated more precisely by comparing the differences between the RSS values of various antenna combinations, *i.e.*, $z_i - z_j$, with $i, j = 1$ to 8 and $i \neq j$. Algorithm 1 shows how the robot uses this technique to control its orientation.

In real world scenarios, by the time the robot estimates the DOCT, it would have moved from where it was when it just started the estimation. This leads to an additional DOCT error which depends on the geometric relations between the robot's current position and its estimated DOCT. To assess the effect of this error, let us assume that the LOS distance $\hat{d}_i$ between receiving antenna i and the robot's target tag at time t, is related to z_i using a "distance-RSS map model". Figure 5.2 depicts a geometric configuration of a robot's orientation and the target tag which is attached at a 3-D point of height h. The target point on the ground indicates the desired point to be reached by the robot. When the robot moves in the direction of DOCT, the following expression holds:

Algorithm 1: Robot orientation update algorithm.

Input: Robot's current orientation θ^{old} and $\hat{\gamma}_l$.
Output: Robot's updated orientation θ^{new}.
begin
  $\Delta\delta = \hat{\gamma}_l - 90°$
  if $\Delta\delta > 180°$ **then**
    Rotate clockwise direction by
    $\Delta\delta = -(360° - \Delta\delta)$
    Update robot orientation using
    $\theta^{new} = \theta^{old} + \Delta\delta$
  if $\Delta\delta < 0$ **then**
    Rotate clockwise direction by $\Delta\delta$
    Update robot orientation using
    $\theta^{new} = \theta^{old} + (360° - |\Delta\delta|)$
  else
    Rotate counter-clockwise direction by $\Delta\delta$
    Update robot orientation by
    $\theta^{new} = \theta^{old} + \Delta\delta$
  if $\theta^{new} \geq 360°$ **then**
    $\theta^{new} = \theta^{new} - 360°$
  Return θ^{new}

$$\frac{\sqrt{\left(\hat{d}_i^2\right)^{new} - h^2}}{\sin\left(\left(\theta_l^d\right)^{old} - \left[\hat{\theta}_l\right]^{old}\right)} = \frac{\sqrt{\left(\hat{d}_i^2\right)^{old} - h^2}}{\sin\left[\pi - \left(\theta_l^d\right)^{new}\right]}$$

which yields,

$$\frac{\sqrt{\left(\hat{d}_i^2\right)^{new} - h^2}}{\sin\left[\left(\tilde{\theta}_l\right)^{old}\right]} = \frac{\sqrt{\left(\hat{d}_i^2\right)^{old} - h^2}}{\sin\left[\left(\theta_l^d\right)^{new}\right]}.$$

As a result, the previous orientation error can be computed as

$$\left(\tilde{\theta}_l\right)^{old} = \sin^{-1}\left\{\sqrt{\frac{\left(\hat{d}_i^2\right)^{new} - h^2}{\left(\hat{d}_i^2\right)^{old} - h^2}}\sin\left[\left(\theta_l^d\right)^{new}\right]\right\}.$$

This shows that the current orientation error can be propagated back to adjust that at the previous instant of time, provided that the "distance-RSS map model" is accurate. Nevertheless, RSS-based proximity determination remains a significant challenge to date due to the stochastic nature of RF signals, especially in the presence of large metal objects in the workspace. Furthermore, it is quite challenging to quantify the true orientation error, $\tilde{\theta}_l$, in real life. The proposed approach circumvents this problem by employing relative RSS measurements at the receiving antennas of the RFID reader for reaching target points in the workspace where modeling an accurate "distance-RSS map" is not necessary. This is simply because it has been experimentally shown that modeling the accurate "distance-RSS map" is impossible even in outdoor environment [Graefenstein and Bouzouraa, 2008].

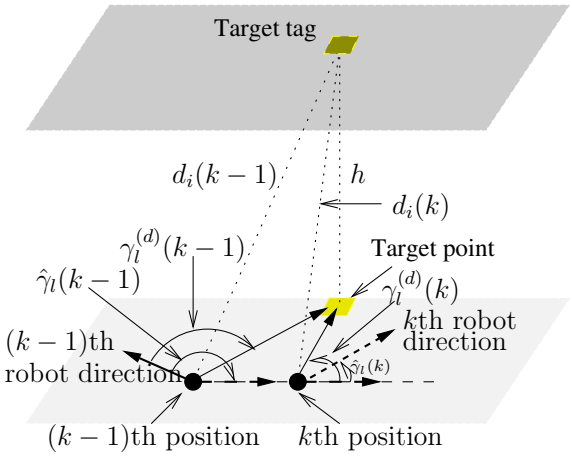


Figure 5.2: Geometric configuration of the orientation error model.

5.2.2.2 Stage 2: Tracking of Desired Path

Without loss of generality, let us assume that antenna "3" is the antenna mounted at the front of the robot. Updating the robot's heading in the previous stage, ensures that the robot points more or less to the direction of its target tag. Hence, the highest RSS values received by the eight antennas would normally come from antennas "2", "3", and "4", as depicted in Figure 5.3. This enables the robot to pass on to the tracking stage in which it navigates to its target point. A fuzzy logic controller (FLC) is used for this purpose. It operates on the RSS values received at antennas "2" and "4" to generate the adequate control actions to be passed to the robot's driving wheels. Opting for an FLC was primarily based on the FLCs' relatively higher tolerance of noise and model imperfections compared to other types of controllers which depend on precise mathematical models of the system in hand [Karray and de Silva., 2004].

A block diagram of the FLC is presented in Figure 5.4. The perception block of the controller aggregates the RSS values from antennas "2" and "4"; *i.e.*, z_2 and z_4. The input to the FLC is $\Delta z_{24} = z_2 - z_4$, which is normalized to $[-1200, 1200]$nW before being passed to the fuzzy inference engine, which in turn computes normalized values of the robot's left and right wheel speeds, u_{Ln} and u_{Rn}, respectively. The post-processing block is used to match these speeds

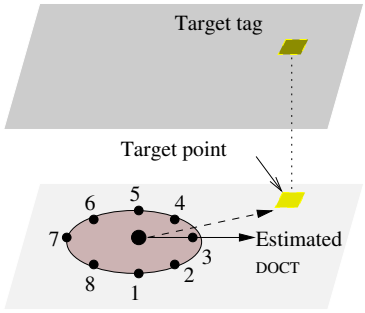


Figure 5.3: Estimating the target direction.

Table 5.1: Fuzzy rules for controlling robot's speed.

	Input	Output	
Rule #	Δz_{24}	u_{Rn}	u_{Ln}
1	NH	VH	VL
2	NL	H	L
3	Z	M	M
4	PL	L	H
5	PH	VL	VH

to the real ones depending on the hardware used. Figure 5.5 depicts the FLC's membership functions. They were tuned based on an empirical analysis to optimize the FLC's performance. The *if-then* rules which are the possible pathological situations reported in Table 5.1. The centroid defuzzification method was used to compute the crisp FLC's outputs (see [Karray and de Silva., 2004] for more details on defuzzification methods). After each inference cycle, the robot updates its pose according to its kinematic model (5.1). Note that the FLC does not depend on the robot's kinematic model in real-time.

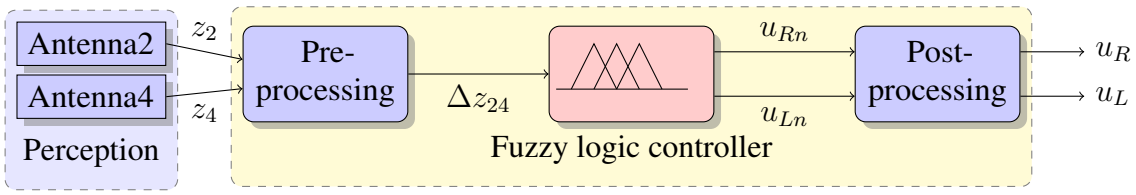
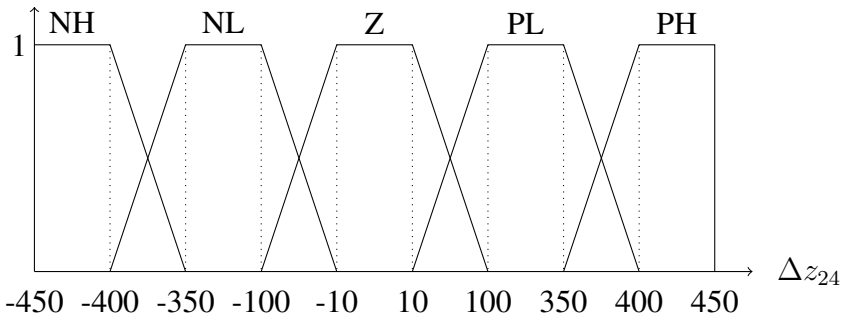
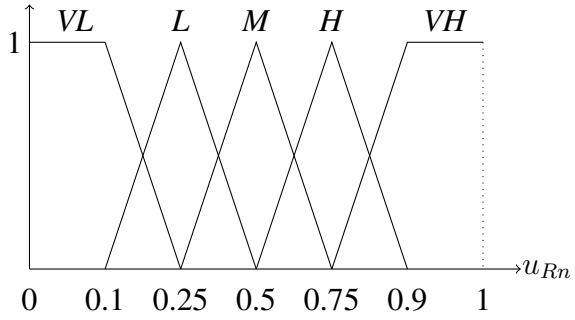


Figure 5.4: Robot's speed control mechanism.

The following is a summary of the key steps of the proposed navigation system integrating



(a)



(b)

Figure 5.5: (a) input, and (b) output membership functions.

the fundamental components discussed so far. The main operational flow of the algorithm is provided in Figure 5.6.

Key steps of bearing-sensitive robot navigation scheme:

Step 1: Pre-program the robot with the list of tag IDs defining the desired path.

Step 2: Record the RSS values received by the eight receiving antennas as sent by the current target tag.

Step 3: Check if the robot reached the current target by computing the differences between the RSS values across different antenna pairs, as discussed in Section 5.2.2.1. If not, then proceed to **step 6**.

Step 4: Check if the current tag is the final tag on the robot's desired path. If so, then stop the robot.

Step 5: The robot sets the next tag ID in the list as its current target tag and passes to the *rotation* mode before going back to **Step 2**.

Step 6: Check the robot's mode (*rotation* or *navigation*) to decide on the next move.

Step 7 to 8: If the robot is in the *rotation* mode, it computes the DOCT using (5.2), rotates itself towards the target tag, and updates its current state to *navigation*.

Step 9: If the robot is in the *navigation* mode, it passes the control to the FLC to generate the left and right wheel speeds which are applied to their respective actuators for the duration of one cycle period T before looping back to **Step 2**.

5.3 Antenna Design for Customized RFID Reader

This section deals with the in-depth antenna design of the customized bearing sensitive RFID reader, which is detailed in section 4.6.1, for mobile robot navigation in an indoor environment. It is well known that RF signals can be easily distorted by reverberant objects, and therefore the error in RSS measurements drastically increases in a highly reverberant environment. To

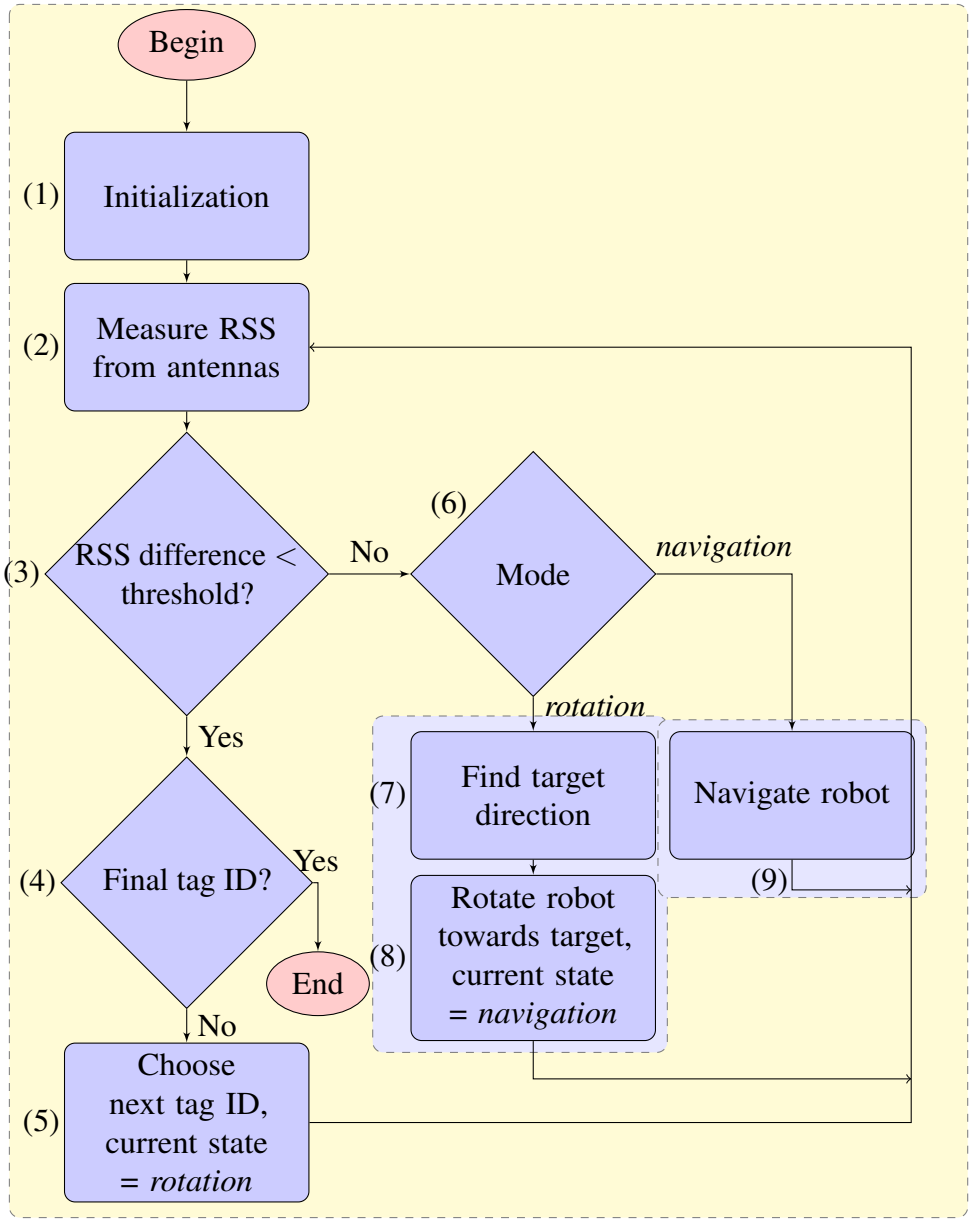


Figure 5.6: Flowchart of the proposed navigation system.

remedy this problem, each receiving antenna on this RFID reader is spatially separated from the other antennas by vertical rectangular metal sheets of dimension $l_p \times h_p$ (see Figure 5.1). Together with the $l_p \times l_p$ metal ground plane, these sheets attenuate the effect of the noise due to signal reverberations, and hence, make it easier to distinguish the different RSS measurements.

In order to optimize the receiving antennas' performance we studied the effect of two key parameters, the vertical metal shields' height, h_p, and the radial placement, r_a, of the antennas from the center of the ground plane whose dimension was fixed to $0.2 \times 0.2\mathrm{m}^2$. The values used for h_p and r_a are $\{0, 8, 15\}\mathrm{cm}$ and $\{5, 10\}\mathrm{cm}$, respectively.

The antenna design analysis is carried out using the commercial simulation software FEKO[1]. This software follows industrial standards to faithfully reproduce what would happen in a real-world scenario. It does so by performing a complex computational process to compute the RSS values at each point in the environment taking into account the effect of reverberating signals bouncing off the walls and other obstacles if any. The configuration setup in FEKO is shown in Figure 5.7. The RFID tag was simulated by an omnidirectional transmitting antenna with an operating frequency of $2.4\,\mathrm{GHz}$ and mounted at $(5, 0, 3)\mathrm{m}$ of the workspace (pointing vertically to the ground). Its total transmitting power was set to $1\,\mathrm{W}$. The eight receiving antennas were set up such that antenna "1" is aligned with the x-axis of the world coordinate system. To decide on the type of antenna to use, both, monopole and dipole antennas were tested.

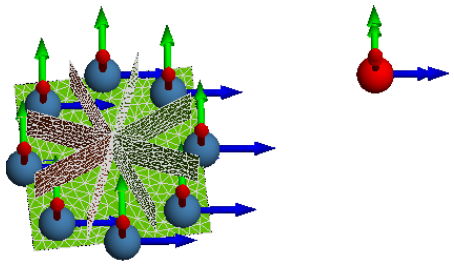


Figure 5.7: Antenna setup in FEKO.

[1] www.feko.info

5.3.1 Performance Using Dipole Antennas

The first set of simulations are conducted with dipole antennas (Figure 5.7). The length of the antenna is chosen to be $\lambda/2 = 6.25\,\mathrm{cm}$, where λ is the RF wavelength. With a matched load of $50\,\Omega$, the input impedance of the antenna is $76.64 + j27.79\Omega$. A 3-D gain plot of how much power is transmitted with respect to the elevation angle θ (with respect to the antenna's axis) is shown in Figure 5.8. This pattern is the same in the transmitting and receiving modes due to the reciprocity property [Balanis, 2005]. Note how very little power is transmitted along the z-axis (the antenna's axis). The radiation is maximum along the x-y plane, i.e., $\theta = \pm 90°$, where the gain is ≈ 1.65 dBi.

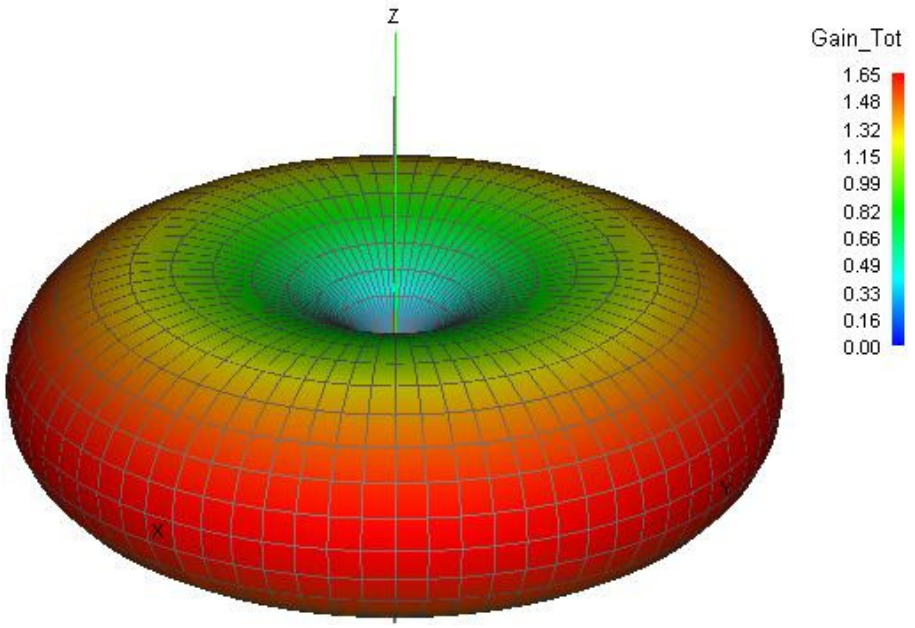
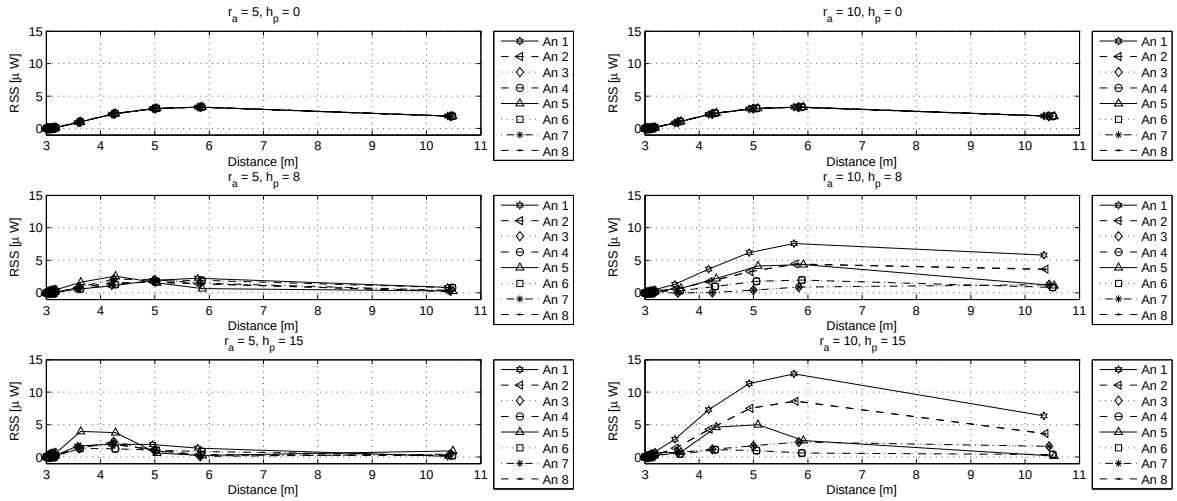


Figure 5.8: Radiation pattern of a dipole antenna of length 6.25 cm.

Initially, the reader is placed right under the tag antenna (line-of-sight distance $\approx 3\,\mathrm{m}$). As expected, the RSS values recorded at the eight antennas were closed to $0\,\mu\mathrm{W}$. This is simply because of the negligible amount of power transmitted along the antenna's axis. Figure 5.9 demonstrates how these RSS values are affected when the robot travels along the direction of antenna 1. The x-axis represents the line-of-sight distance between the receiving and the

tag antennas. Note how when $r_a = 5\,\mathrm{cm}$ and $h_p = 0\,\mathrm{cm}$, the eight RF signals cannot be discriminated, which makes them useless in this application. When r_a is varied to 8 and $15\,\mathrm{cm}$, the RF signals become more distinguishable. Yet, the front receiving antenna's RSS value is not always higher than those of the other antennas. This problem does not occur when $r_a = 10\,\mathrm{cm}$ as shown in Figure 5.9(b). This set of simulations show that the optimal values of r_a and h_p are $10\,\mathrm{cm}$ and $8\,\mathrm{cm}$, respectively.



(a) (b)

Figure 5.9: RSS values for dipole antennas (a) $r_a = 5\,\mathrm{cm}$, and (b) $r_a = 10\,\mathrm{cm}$.

5.3.2 Performance Using Monopole Antennas

The same experiment is repeated replacing the dipole antennas with monopole ones of length $\lambda/4 = 3.12\,\mathrm{cm}$. Unlike the dipole antennas, the monopole antennas have a ground plane whose dimension was set to $4\,\mathrm{cm}^2 \times 4\,\mathrm{cm}^2$. The input impedance of the monopole antenna is chosen as $32.12 + j22.76\Omega$ and the far-field 3-D radiation pattern is obtained as with a dipole antenna. This type of antenna has an omnidirectional radiation pattern with a lower gain (≈ 1.22 dBi) compared to the dipole antenna (Figure 5.10).

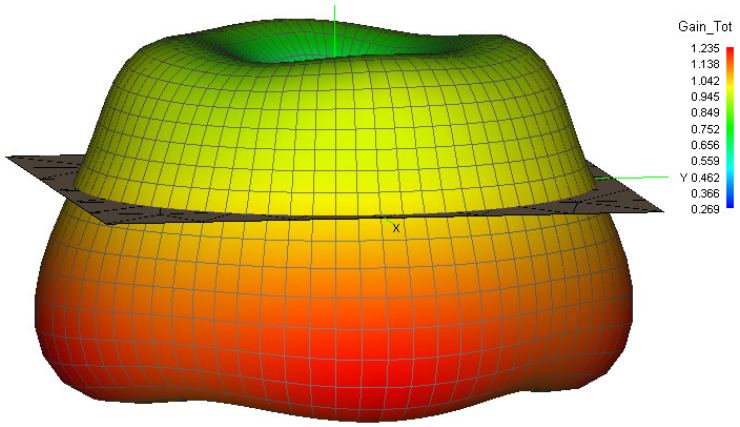
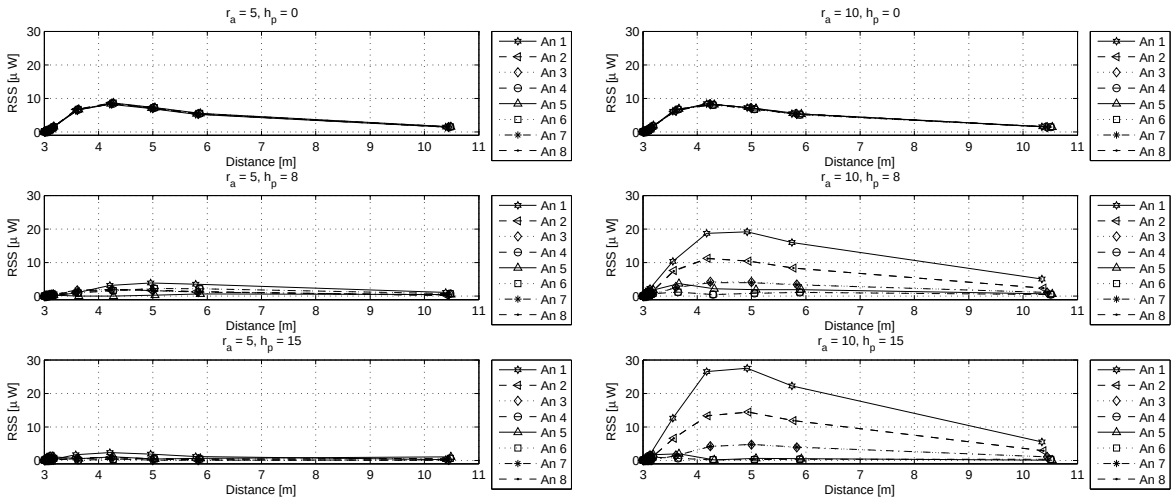


Figure 5.10: Radiation pattern of a monopole antenna of length 3.12 cm.

The same FEKO simulations conducted with a dipole antena were repeated. The results are revealed in Figure 5.11. As can be observed, although the RSS values are sensitive to the line-of-sight distance between the receiving and the transmitting antennas when $h_p = 0$ and $r_a = 5$ or $10\,\mathrm{cm}$, they are indistinguishable. As with dipole antennas, these signals are better segregated when $r_a = 10\,\mathrm{cm}$ and $h_p = 8$ or $15\,\mathrm{cm}$.



(a) (b)

Figure 5.11: RSS values for monopole antennas (a) $r_a = 5\,\mathrm{cm}$, and (b) $r_a = 10\,\mathrm{cm}$.

Among the two types of antennas, it was found that the dipole antenna performed better due

to its smooth radiation pattern with respect to the monopole counterpart. The best performance is attained with dipole antennas when $r_a = 10\,\mathrm{cm}$ and $h_p = 8\,\mathrm{cm}$. This configuration allows for the best discrimination of, both, the line-of-sight distance and the the direction of the target tag (DOCT). This is not the case though when the receiving antenna is about $40\,\mathrm{cm}$ or less from the target point on the ground (right under the tag) due to the very low power radiated vertically (see Figures 5.8 and 5.10).

5.4 Robot Navigation Performance

The purpose of this section is to study the performance of the proposed navigation system using the virtual differential drive mobile robot model defined in section 3.3.3. The performance metric adopted to assess the performance of the proposed approach is the pose error at time t defined by $\tilde{\mathbf{q}}_l(t) = \left[\tilde{x}_l(t)\ \tilde{y}_l(t)\ \tilde{\theta}_l(t)\right]^T$ in the robot's local coordinate system which is defined as:

$$\tilde{\mathbf{q}}_l(t) = \begin{bmatrix} \cos(\hat{\theta}(t)) & \sin(\hat{\theta}(t)) & 0 \\ -\sin(\hat{\theta}(t)) & \cos(\hat{\theta}(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{q}}(t)\,, \tag{5.3}$$

where $\tilde{\mathbf{q}}(t) = \mathbf{q}_d(t) - \hat{\mathbf{q}}(t)$. The robot's position error is defined as $e(t) = \sqrt{\tilde{x}_l^2(t) + \tilde{y}_l^2(t)} = \sqrt{\tilde{x}^2(t) + \tilde{y}^2(t)}$ [Kanayama et al., 1990]. In order to quantify the robot's navigation accuracy, the root mean squared error (RMSE) is used as a performance measure which is defined as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{k=1}^{k_f} e^2(k)}{k_f}}, \tag{5.4}$$

where $t = kT$, for $k = 0, 1, \ldots, k_f$. The sampling time period considered for the simulations is $T = 1\,\mathrm{s}$. The dimension of the virtual test area is $\approx 5 \times 5 \times 3\,\mathrm{m}^3$. The robot is programmed to pass by an ordered sequence of points on the ground $p_1 = (2, 0.5, 0)\mathrm{m}$, $p_2 = (3.5, 2, 0)\mathrm{m}$,

$p_3 = (2, 3.5, 0)\text{m}$, and $p_4 = (4, 3.5, 0)\text{m}$. To achieve this goal, four tags were mounted on the ceiling at locations $(2, 0.5, 3)\text{m}$, $(3.5, 2, 3)\text{m}$, $(2, 3.5, 3)\text{m}$, and $(4, 3.5, 3)\text{m}$. The robot is initially placed at the origin of the world coordinate system with an orientation of $90°$. The customized RFID reader with the antenna system shown in Figure 5.7 is mounted on top of the robot. Only dipole antennas are considered in these simulations for their simple design and better performance compared to monopole antennas. Three scenarios are considered to study different aspects of the proposed navigation system, as detailed below.

5.4.1 Navigation in a Non-reverberant Environment

The first scenario simulates an ideal case where the workspace contains no reverberant objects. Figure 5.12(a) reveals the actual and the desired trajectories of the mobile robot in such a scenario. The four RFID tag locations are denoted by crosses ('X'). In this case, the robot's path can be regarded as a four-phase mission. In the first phase, the robot reaches out for point p_1 as its momentary target. Note that a desired target direction of $\approx 14°$ has to be achieved to accomplish this mission. The RFID reader receives the highest RSS value ($\approx 1.25\ \mu$W) from antenna 1. This is shown in Figure 5.12(c). The direction of this antenna with respect to the robot's local coordinate system is 0^o, as defined by (5.2). Hence, the robot is in the *rotation* state and updates its heading direction towards the target point p_1. Notice that the robot has detected its destination point p_1 with an error of $\approx 14\,\text{cm}$. This is due to the fact that it has received approximately equal RSS values across all of its receiving antennas. In other words, the robot has met its stopping criteria for it to reach the current target point.

When the robot is in the vicinity of point p_1, it starts the second phase of navigation where p_2 becomes the next target point. As can be noted from Figure 5.12(c), at the beginning of this phase, it is antenna 4 that is found to provide the highest RSS value and the robot's updated direction is then estimated as $\approx 60°$. The third and fourth phase of navigation to reach points p_3

and p_4 are conducted in the same manner.

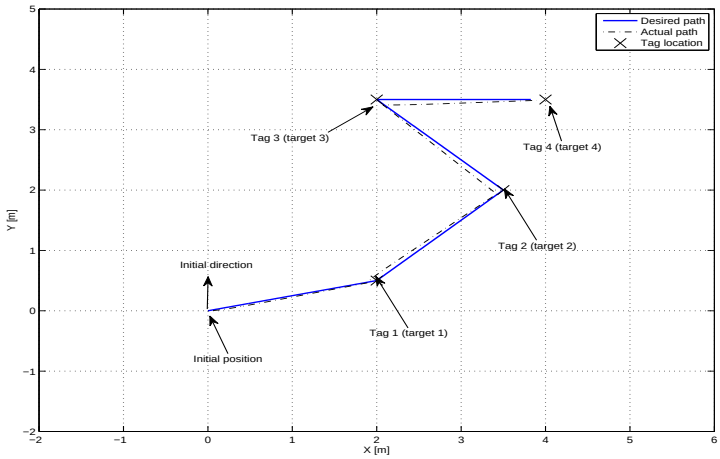
As shown in Figure 5.12(a), the orientation errors increase around the corners of the desired path. This is natural since the robot needs time to go back on track after each corner. Figure 5.12(b) shows that for 90% of the simulation time, the position and orientation errors are less than $0.12\,\mathrm{m}$ and $4°$, respectively. The RMSE in this experiment is about $9\,\mathrm{cm}$.

5.4.2 Navigation in a Reverberant Environment with Metal File Cabinets

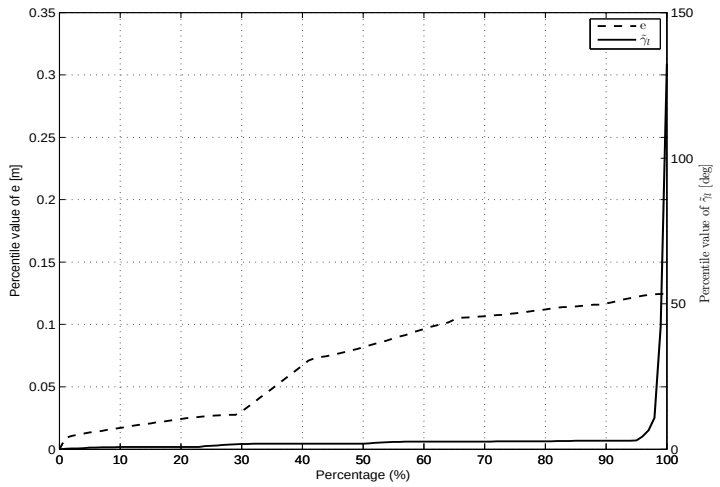
To better illustrate the performance of the proposed navigation system, the above simulation is repeated with a large $1.5 \times 1 \times 0.35\mathrm{m}^3$ metal file cabinet added to the environment. The results are reported in Figure 5.13. The navigation performance has not been affected much by the reverberation from the metal file cabinet as the robot could still reach the desired target points with a satisfactory pose error. It can be seen from the percentile error plot in Figure 5.13(b) that for 90% for the robot's travel time, the position and direction errors are less than $27\,\mathrm{cm}$ and $8°$, respectively. The RMSE recorded for this simulation is $19\,\mathrm{cm}$.

5.4.3 Navigation in a Reverberant Environment with Metal Objects of Arbitrary Shapes

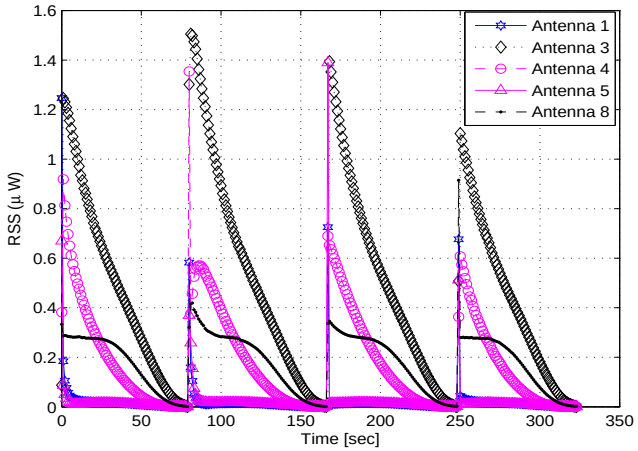
The purpose of this scenario is to study the navigation system's ability to sustain various magnitudes of RF noise and reverberations induced by different types of metal objects in the robot's environment. The results are shown in Figure 5.14. Three metal objects are placed close to the robot's desired path as shown in Figure 5.14(a). The performance was not much different from that of the second simulation. Figure 5.14(b) shows that the error values lie within more or less the same range as in the previous experiment. The RMSE for this scenario is $12\,\mathrm{cm}$.



(a)

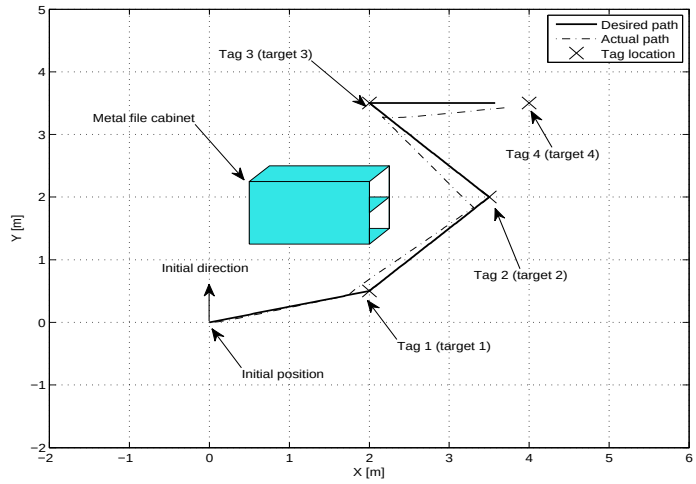


(b)

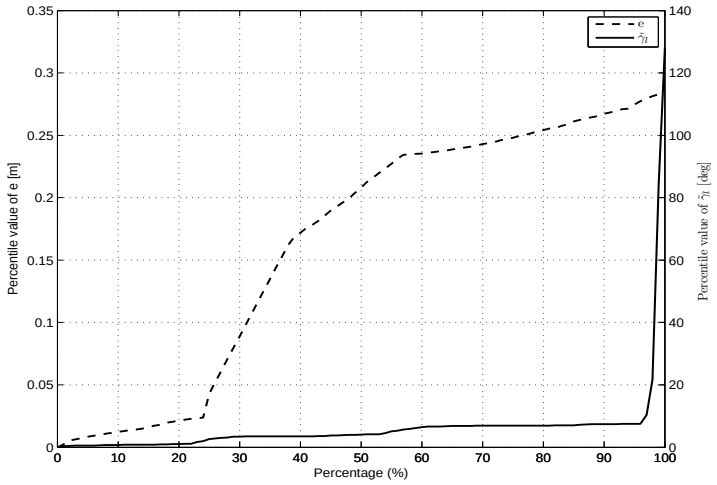


(c)

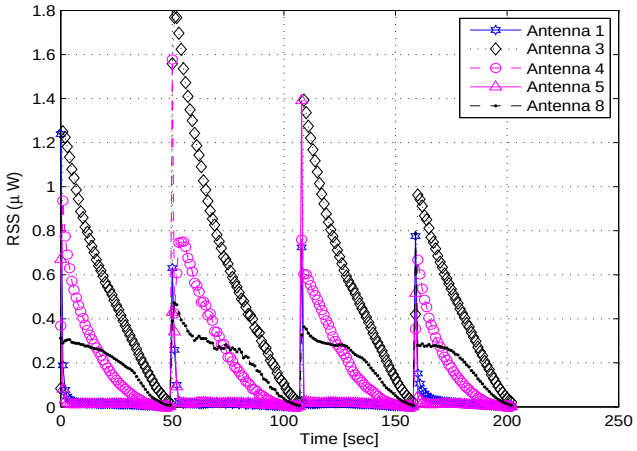
Figure 5.12: Navigation performance in a non-reverberant environment, (a) trajectory, (b) percentile error for position and orientation, and (c) RSS measurements from different antennas.



(a)

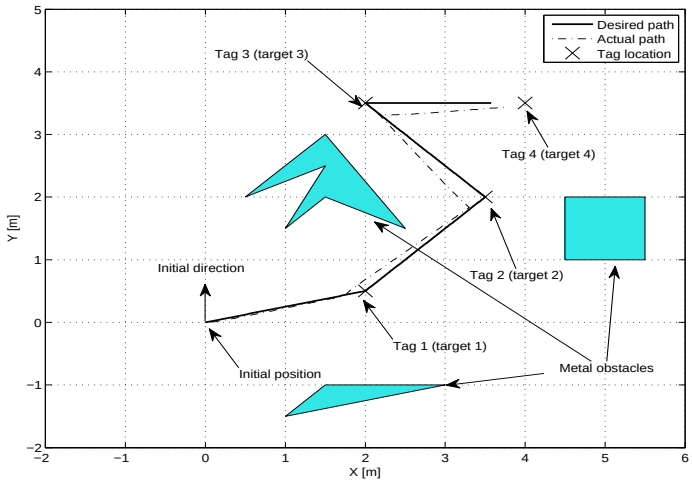


(b)

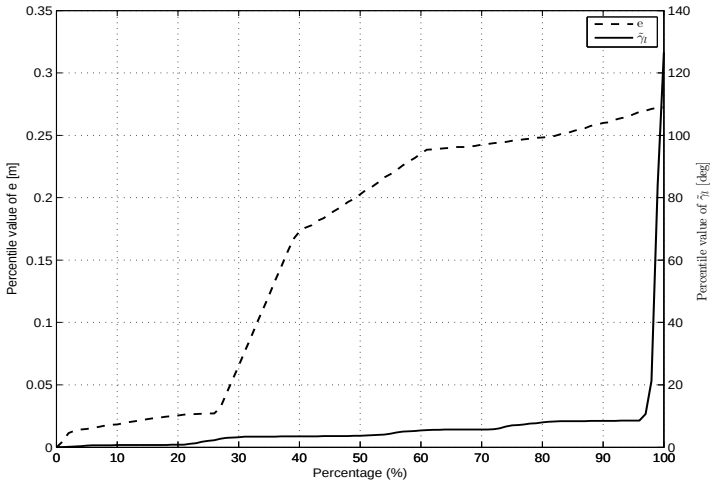


(c)

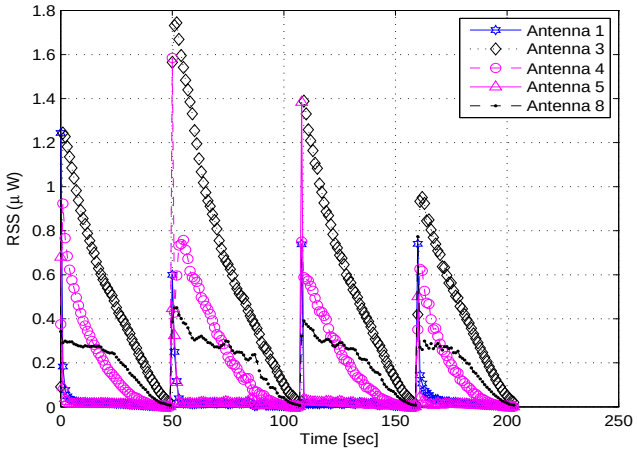
Figure 5.13: Navigation performance in an environment with a metal file cabinet, (a) trajectory, (b) percentile error for position and orientation, and (c) RSS measurements from different antennas.



(a)



(b)



(c)

Figure 5.14: Navigation performance in a reverberant (metal) environment, (a) trajectory, (b) percentile error for position and orientation, and (c) RSS measurements from different antennas.

The above simulations ensure that the proposed navigation system is capable of guiding a mobile robot in reaching a set of target points in an indoor environment with a satisfactory tracking error using a modified RFID architecture regardless of noisy signals reverberated from metal objects. It is worth noting that an RSS-based robot localization algorithm with a custom-made receiving antenna was presented by Graefenstein *et al* in [Graefenstein and Bouzouraa, 2008], which is basically similar to what is presented in this chapter except that RSS measurements are used here for the purpose of robot navigation as opposed to localization. The localization error reported in [Graefenstein and Bouzouraa, 2008] is in the range of 30-40 cm as opposed to 9-20 cm here. This is despite the fact that not much RF reverberations were conducted in [Graefenstein and Bouzouraa, 2008] since it mostly deals with outdoor robotic applications.

5.5 Summary

The navigation method suggested in this chapter is based on an RFID system whose reader is redesigned to be most suitable for this application. The RFID reader with eight receiving antennas is employed to estimate the direction of RF signal coming from the desired target position. Once the direction of the desired target is determined, the robot simply navigates towards that target using its actuators whose actions are mainly generated by an FLC that performs intelligent processing of tags RSS measurements received at the RFID reader mounted on the robot. Note that the antenna array of the reader is designed in such a way as to attenuate RF noise and signal reverberations even in unfavorable environments. The effectiveness of this approach was demonstrated in three scenarios with different environment complexities of various noise and reverberation amplitudes. For it to be as realistic as possible, the navigation system was simulated using FEKO, a comprehensive industrial electromagnetic software that takes the finest details into account for computing the RSS values at each point in the workspace.

It is clear that the simple bearing sensitive control scheme presented in this chapter has the ability to navigate a mobile robot in an indoor environment with a satisfactory position errors in terms of RMSEs ($\approx 20\mathrm{cm}$). In addition, this control scheme also shows its power to tackle signal reverberations in an environment populated with metal obstacles.

Chapter 6

Optimal Measurement Feedback Control For Trajectory Tracking

6.1 Introduction

This chapter and the following chapter contribute to the development of mobile robot navigation systems (geometric path following with an associated timing law) based on the modified RFID tag architecture as opposed to reader architecture discussed in Chapter 5. The modified RFID tag architecture is detailed in section 4.6.2. Note that RSS measurements from modified RFID tags placed in the robot's operating environment are used as the crucial information for navigation. Most of the existing mobile robot navigation techniques that depend on reference RF beacons rely on approximating line-of-sight (LOS) distance between these beacons and the robot. The approximation of LOS is mostly performed by using RSS measurements of signals propagating between the robot and RF beacons. However, to date, relying on RSS measurements for approximating LOS distance remains a very significant challenging task as it is highly dependent on environment [Kim and Chong, 2009, Youssef, 2004]. Especially, mapping between RSS measurements and LOS distance is almost impossible in an indoor

reverberant environment. It is worth pointing out that most of the RFID-based navigation systems suggested in the literature are tailored along with the localization systems where the central task of an RFID system is to estimate the position of a mobile robot rather than generating it's actuator control inputs, see [Graefenstein and Bouzouraa, 2008, Liu et al., 2011], for example.

Having said that, a novel optimal output feedback control law for the problem of mobile robot trajectory tracking is presented in this chapter. The feedback control is still based on noisy RSS measurements. This feedback controller does not require accurate mapping of LOS distance and RSS measurements from RFID tags. Unlike other feedback controllers that perform dynamic linearization of DDMR model around certain operating points [Chen et al., 2009, Chwa, 2010], the proposed controller provides the measurement (output) feedback control to the original nonlinear DDMR's model (i.e., without the linearization of the model's dynamics). Moreover, this controller takes advantage of the general optimal control theory that is applicable to any dynamical system to find the optimal control inputs for an optimal state trajectory. Note that the feedback control design problem for tracking a pre-defined trajectory or reaching a desired target using a nonholonomic mobile robot remains quite challenging to date. This is due to a well-known Brockett's theorem [Brockett, 1983] which proves the non-existence of the smooth state-feedbacks for the asymptotic stabilization of fixed configurations. It implies that controlling nonholonomic systems cannot be applied to methods of linear control theory, and they are not transformable into linear control problems. These limitations lead us to focusing on designing a feedback control law based on an optimization control technique. The feedback control law described in this chapter generates optimal control inputs which lead the robot to track a pre-defined trajectory in an indoor environment.

The rest of the chapter is outlined as follows. It begins with describing a high level architecture of the robot navigation system in section 6.2. Section 6.3 gives the mathematical

formulation of the tracking problem for a DDMR. This problem is then sloved by designing an optimal measurement feedback control law, which is the main contribution of this chapter. This is illustrated in section 6.4. A thorough evaluation of the proposed feedback control law with some numerical simulation results is presented in section 6.5. Finally, conclusions with some limitations are drawn in section 6.6.

6.2 Navigation System Architecture

A mobile robot is pre-programed with an ordered sequence of RFID tag IDs which are attached at unknown 3-D locations in the robot's workspace. For instance, if the mobile robot is provided with a sequence of four 16-bit tag IDs, 0xFFF9, 0xFFF2, 0xFFF5, and 0xFFF4, then it is supposed to receive RSS measurements from the tags within the operating range of the RFID reader which is mounted on the robot. These RSS measurements are then used to generate the appropriate control actions for the robot's actuators in order to track a pre-defined path in an indoor environment. In this case, the robot's desired trajectory is independent of the tags orthogonal positions on the ground. A high level setup of the proposed navigation system with four RFID tags attached to the ceiling of an indoor environment is depicted in Figure 6.1. In this configuration, the robot's mission is to track the desired geometric path defined by $\mathbf{q}^d(t)$, for $t \in [t_0, t_f]$, with an associated timing law. Before detailing the proposed navigation strategy (output feedback control law), a mathematical formulation of the tracking problem using RSS measurements is necessary which is given in the following section.

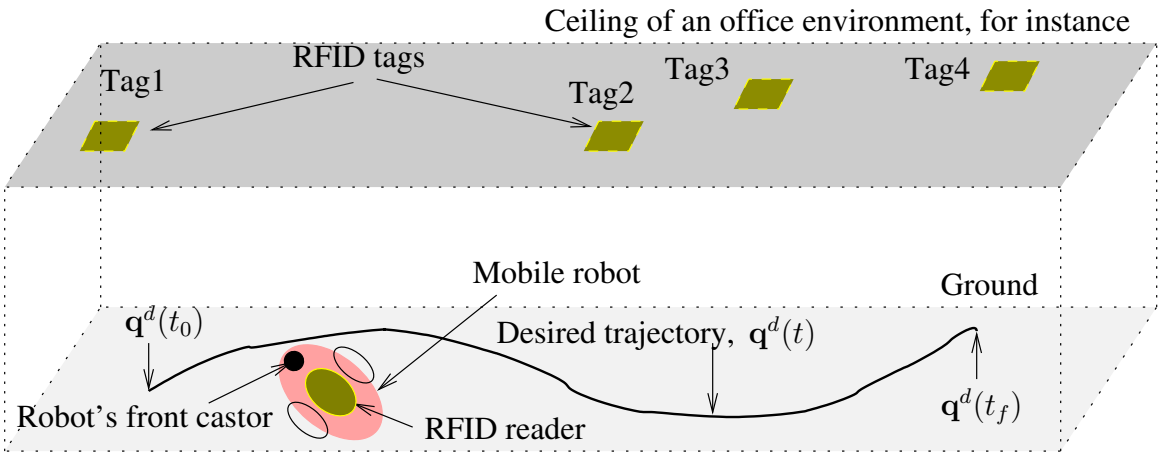


Figure 6.1: High level system architecture using modified RFID tag architecture.

6.3 Problem Formulation

In order to design the output feedback control law for solving the problem (3.12), consider the RSS measurement model

$$\mathbf{z}(t) = \hat{\mathbf{h}}[\hat{\mathbf{q}}(t)] + \boldsymbol{\zeta}(t), \tag{6.1}$$

where $\mathbf{z}(t) \in \mathbb{R}^s$ is the RSS measurement vector from s RFID tags in the environment and $\boldsymbol{\zeta}(t) \in \mathbb{R}^s$ is the noise associated with the RSS measurements. The nonlinear measurement function $\hat{\mathbf{h}}[\hat{\mathbf{q}}(t)] = \left[\hat{h}_1[\hat{\mathbf{q}}(t)] \ \ldots \ \hat{h}_s[\hat{\mathbf{q}}(t)]\right]^T$ of (6.1) is given by $\mathbf{h} : \mathbb{R}^2 \times \mathbb{S}^1 \to \mathbb{R}^s$ with

$$\hat{h}_j[\mathbf{q}(t)] = \alpha e^{\beta \hat{d}_j}, \text{ for } j = 1, \ldots, s, \tag{6.2}$$

where α and β are the parameters which are obviously dependent on the operating environment. These parameters will be later optimized on-line using the nonlinear least square method. The estimated distance $\hat{d}_j$ of (6.2) is simply the Euclidean distance between the robot's current estimated position $(\hat{x}, \hat{y})$ and j-th RFID tag position $\mathbf{q}_t^j$. If $\mathbf{q}_t^j = [x_t^j \ y_t^j \ z_t^j]^T$ represents the 3D

position of the j-th tag, then

$$\hat{d}_j = \sqrt{(\hat{x} - x_t^j)^2 + (\hat{y} - y_t^j)^2 + (z_t^j)^2}, \text{ for } j = 1, \ldots, s.$$

The control defined earlier by (3.9) is now modeled as the output feedback control law

$$\mathbf{u}(t) = \mathbf{K}(t)\mathbf{z}(t) \tag{6.3}$$

subject to (3.8), where $\mathbf{K}(t) \in \mathcal{K}_{ad} \subset \mathbb{R}^{2\times s}$, $t \in I$, is the feedback control gain for the system (3.7). Substituting (6.3) in (3.7) and using the measurement model (6.1), the robot's feedback system can be formulated as

$$\dot{\mathbf{q}}(t) = \frac{r}{2}\mathbf{B}[\mathbf{q}(t)]\mathbf{K}(t)\mathbf{h}[\mathbf{q}(t)] + \frac{r}{2}\mathbf{B}[\mathbf{q}(t)]\mathbf{K}(t)\boldsymbol{\zeta}(t). \tag{6.4}$$

For simplicity, the feedback model (6.4) is rewritten in the form

$$\dot{\mathbf{q}}(t) = \hat{\mathbf{f}}[\mathbf{q}(t), \mathbf{K}(t)] + \hat{\mathbf{g}}[\mathbf{q}(t), \mathbf{K}(t)],$$

where

$$\hat{\mathbf{f}}[\mathbf{q}(t), \mathbf{K}(t)] \equiv \frac{r}{2}\mathbf{B}[\mathbf{q}(t)]\mathbf{K}(t)\mathbf{h}[\mathbf{q}(t)] \text{ and}$$
$$\hat{\mathbf{g}}[\mathbf{q}(t), \mathbf{K}(t)] \equiv \frac{r}{2}\mathbf{B}(\mathbf{q}(t))\mathbf{K}(t)\boldsymbol{\zeta}(t).$$

The block diagram of the proposed feedback system is shown in Figure 6.2. The reference trajectory block is to synthesize feasible nominal paths, $\mathbf{q}^d(t),\ t \in I$. The optimal controller block takes the tracking error and the noisy tags' RSS measurements into account in order to generate optimal control inputs $\mathbf{u}(t)$ which are passed to the robot's actuator. Since the feedback

control gain $\mathbf{K}(t)$ in (6.3) needs to be optimized in order to generate optimal $\mathbf{u}(t)$, the current feedback control problem is reduced to an optimization problem. This is the main contribution of this chapter that will be detailed in section 6.4.

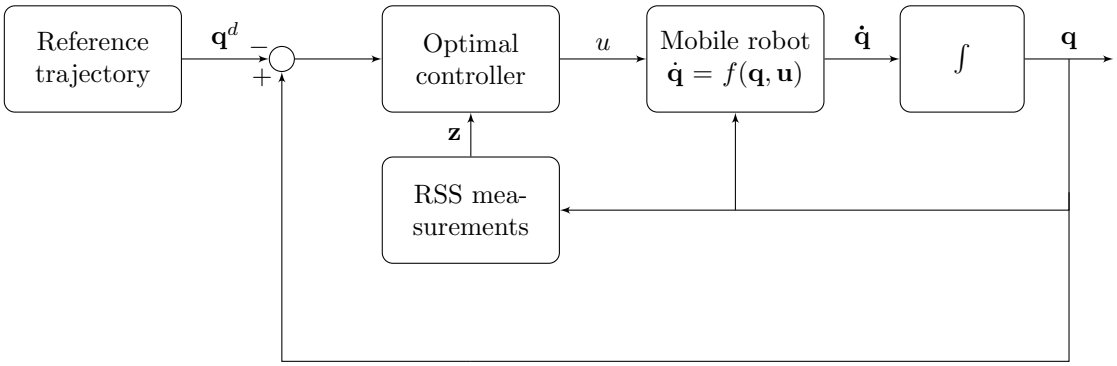


Figure 6.2: Feedback structure of the proposed trajectory tracking method.

For the robot to find the optimal trajectory $\mathbf{q}^o(t)$, define the cost functional as

$$J(\mathbf{K},\boldsymbol{\zeta}) = \frac{1}{2}\left[(\mathbf{q}(t_f) - \mathbf{q}^d(t_f))^T \mathbf{P}(t_f)(\mathbf{q}(t_f) - \mathbf{q}^d(t_f))\right] + \frac{1}{2}\int_{t_0}^{t_f} \left[(\mathbf{q}(t) - \mathbf{q}^d(t))^T \mathbf{Q}(t)(\mathbf{q}(t) - \mathbf{q}^d(t))\right] dt \,, \tag{6.5}$$

where $\mathbf{P}(t_f), \mathbf{Q}(t) \in \mathbb{R}^{3\times 3}$ are symmetric positive definite matrices that indicate the relative importance of the error components along $\mathbb{R}^2 \times \mathbb{S}^1$. If the robot's purpose is to reach a desired target point in its environment, then the weight matrix $\mathbf{P}(t_f)$ must be larger than $\mathbf{Q}(t)$. The opposte is true for the robot to track a desired trajectory. The performance index $J(\mathbf{K}, \boldsymbol{\zeta})$ in (6.5) depends on the feedback control gain matrix $\mathbf{K}(t)$ through the state variable $\mathbf{q}(t)$ as it is clear from the feedback system (6.4). A compact form of (6.5) can be written as

$$J(\mathbf{K}, \boldsymbol{\zeta}) = \phi[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \ell[t, \mathbf{q}(t)]dt. \tag{6.6}$$

Note that the cost function (6.6) is similar to the performance measure defined in Bolza type problem, where $\phi[t_f, \mathbf{q}(t_f)] = \frac{1}{2}\left[(\mathbf{q}(t_f) - \mathbf{q}^d(t_f))^T \mathbf{P}(t_f)(\mathbf{q}(t_f) - \mathbf{q}(t_f)^d)\right]$ represents the

terminal cost and $\ell[t, \mathbf{q}(t)] = \frac{1}{2}\left[(\mathbf{q}(t) - \mathbf{q}^d(t))^T \mathbf{Q}(t)(\mathbf{q}(t) - \mathbf{q}^d(t))\right]$ represents the running cost [Ahmed, 1988]. Given the kinematic model of the robot in (3.7) and its nonholonimic constraint (3.2), the objective is now to

find $\mathbf{u}(t)$ and $\mathbf{q}(t),\ t \in I$,

minimizing $J(\mathbf{K}, \boldsymbol{\zeta})$,

subject to (3.7), (3.8), and (3.2).

6.4 Optimal Feedback Control Law

An important remark on RSS-based robot localization systems suggested in the literature is that simple RSS measurements from RFID tags can not make the robot decide on it's accurate position with respect to the map of the operating environment. This problem has been traditionally solved by several approaches suggested in the literature [Miah and Gueaieb, 2009b, Wu et al., 2009]. Among the most common shortcomings of those approaches are the use of excessive number of sensors or multiple reference RF stations for the robot to estimate its location in an indoor/outdoor environment.

The purpose of this section is to study the proposed optimal feedback control law for determining the control inputs (*i.e.*, speeds) where noisy RSS measurements from RFID tags are employed as the feedback information. These control inputs are applied to the robot's actuators for tracking a desired trajectory in its workspace. For that, let us introduce some fundamental assumptions:

(A1): since the input $\mathbf{u}(t) \in \mathcal{U}_{ad}$ is a piecewise constant bounded function; that is, $\mathbf{u}(t) \in \mathcal{L}_\infty([0, \infty), \mathbb{R}^2)$, the set $\mathcal{K} \subset \mathbb{R}^{2\times s}$ must be a closed bounded convex set and

$$\mathcal{K}_{ad} \equiv \left\{\mathbf{K}(t) \in \mathcal{L}_\infty([0, \infty), \mathbb{R}^{2\times s}) \ : \ \mathbf{K}(t) \in \mathcal{K}\right\}.$$

(A2): the noise vector of the measurement model (6.1), $\boldsymbol{\zeta} : [0, \infty) \to \mathbb{R}^s$, is a measurable

stochastic process taking values from the closed (Euclidean) ball $\mathcal{B}(\bar{\boldsymbol{\zeta}}, r^{'})$ defined by

$$\mathcal{B}(\bar{\boldsymbol{\zeta}}, r^{'}) = \left\{ \boldsymbol{\zeta}(t) \in \mathbb{R}^{s} \ : \ \|\boldsymbol{\zeta}(t) - \bar{\boldsymbol{\zeta}}\| \leq r^{'} \right\},$$

where $r^{'} > 0$ is the radius of the noise associated with the RSS measurements and $\bar{\boldsymbol{\zeta}}$ is the mean of $\boldsymbol{\zeta}(t)$, for $t \in I$.

To solve for the optimal trajectory using the feedback system (6.4) that minimizes the objective functional (6.6), we need to derive the necessary conditions of optimality. These necessary conditions are most readily found if the integrand of the cost functional (6.6) is recast in terms of Hamiltonian

$$\mathcal{H} : I \times \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^3 \times \mathbb{R}^{2\times s} \longrightarrow \mathbb{R},$$

which is given by

$$\mathcal{H}[\mathbf{q}(t), \boldsymbol{\psi}(t), \mathbf{K}(t)] = \boldsymbol{\psi}^{T} \left[\hat{\mathbf{f}}[\mathbf{q}(t), \mathbf{K}(t)] + \hat{\mathbf{g}}[\mathbf{q}(t), \mathbf{K}(t)] \right] + \ell(t, \mathbf{q}(t)), \tag{6.7}$$

where $\boldsymbol{\psi}(t) \in \mathbb{R}^3, \ t \in I$, is a vector of Lagrange multipliers whose elements are called the costates of the system [Ahmed, 1988]. Before deriving the necessary conditions of optimality, we present the following Lemma pertaining to the robot's feedback model (6.4).

Lemma 6.1 (Solutions of feedback system)**.** *Given the assumptions* **(A1)**-**(A2)**, *consider the feedback system* (6.4)*, which is defined over the finite time horizon of* I*, then for every initial condition* $\mathbf{q}(0) \in \mathbb{R}^2 \times \mathbb{S}^1$*, and feedback control gain* $\mathbf{K}(t) \in \mathcal{K}_{ad}$*, the system* (6.4) *has a unique absolutely continuous solution* $\mathbf{q}(t) \in C(I, \mathbb{R}^2 \times \mathbb{S}^1)$*. Furthermore, the solution set* $\mathcal{Q} \equiv \{\mathbf{q}(t) \equiv \mathbf{q}(t, \mathbf{K}(t), \boldsymbol{\zeta}(t)) \in C(I, \mathbb{R}^2 \times \mathbb{S}^1) : \mathbf{K}(t) \in \mathcal{K}_{ad}\}$ *is a bounded subset of* $C(I, \mathbb{R}^2 \times \mathbb{S}^1)$.

Proof. The proof is classical and follows from the similar technique as given in [Ahmed, 2006,

Theorem 3.5.1, page 89]. □

According to Lemma 6.1, if $\mathbf{q_0} \equiv \mathbf{q}(t_0)$ denotes the robot's pose at time t_0, the actual trajectory of the robot can be described by

$$\mathbf{q}(t) = \mathbf{q}_0 + \int_{t_0}^{t} \left\{ \hat{\mathbf{f}}[\mathbf{q}(\tau), \mathbf{K}(\tau)] + \hat{\mathbf{g}}[\mathbf{q}(\tau), \mathbf{K}(\tau)] \right\} d\tau, \tag{6.8}$$

where $\mathbf{q}(t) \in \mathcal{Q}$, for $t \in I$.

Theorem 6.1 (Necessary conditions of optimality)**.** *Consider the feedback system* (6.4) *satisfying Lemma 6.1. The optimal trajectory* $\mathbf{q}^o(t),\ t \in I$ *can be obtained if there exists an optimal feedback control gain* $\mathbf{K}^o(t) \in \mathcal{K}_{ad}$ *and an optimal multiplier* $\boldsymbol{\psi}^o(t) \in C(I, \mathbb{R}^3)$ *such that the triple* $\{\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)\}$ *satisfies the following necessary conditions:*

$$\mathcal{H}[\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}(t)] \geq \mathcal{H}[\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)],\ \ \forall\, \mathbf{K}(t) \in \mathcal{K}, t \in I, \tag{6.9}$$

$$\dot{\mathbf{q}}^o = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\psi}}[\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K^o}(t)],\ \mathbf{q^o}(t_0) = \mathbf{q_0},\ t \in I, \tag{6.10}$$

$$\dot{\boldsymbol{\psi}}^o = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}[\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)],\ \boldsymbol{\psi}^o(t_f) = \frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)]. \tag{6.11}$$

Proof. Let $\mathbf{q}(t) \equiv \mathbf{q}[t, \mathbf{K}(t), \boldsymbol{\zeta}(t)]$ be the solution of the feedback system (6.4), with the cost functional (6.6) for any choice of $\mathbf{K}(t) \in \mathcal{K}_{ad}$. For simplicity of the proof, and without loss of generality, assume that the noise vector $\boldsymbol{\zeta}(t) \equiv \boldsymbol{\zeta}_c$ is fixed. Since $\mathbf{K}^o(t)$ is optimal with the associated trajectory $\mathbf{q}^o(t)$, it is clear that

$$J[\mathbf{K}^o(t), \boldsymbol{\zeta}_c] \leq J[\mathbf{K}(t), \boldsymbol{\zeta}_c],\ \ \forall\, \mathbf{K}(t) \in \mathcal{K}_{ad},\ t \in I.$$

Suppressing the variable t for clarity and for any $\varepsilon \in [0, 1]$, we define $\mathbf{K}^\varepsilon = \mathbf{K}^o + \varepsilon(\mathbf{K} - \mathbf{K}^o)$. Since $\mathcal{K}$ is a closed convex set, $\mathcal{K}_{ad}$ is also a closed convex subset of $\mathcal{L}_\infty(I, \mathbb{R}^{2\times s})$ and therefore,

$\mathbf{K}^\varepsilon \in \mathcal{K}_{ad}$. Thus, $J(\mathbf{K}^o, \boldsymbol{\zeta}_c) \leq J(\mathbf{K}^\varepsilon, \boldsymbol{\zeta}_c)$, which yields

$$Tr\left[(\mathbf{K} - \mathbf{K}^o)^T \, dJ(\mathbf{K}^o, \boldsymbol{\zeta}_c)\right] \geq 0, \tag{6.12}$$

where $dJ(\mathbf{K}^o, \boldsymbol{\zeta}_c)$ denotes the Gateaux (directional) derivative of J evaluated at $\mathbf{K} = \mathbf{K}^o$ in the direction of $\mathbf{K} - \mathbf{K}^o$.

Let $\mathbf{q}^\varepsilon$ be the solution of the feedback system (6.4) corresponding to the gain $\mathbf{K}^\varepsilon$ with the same initial state $\mathbf{q}^\varepsilon(t_0) = \mathbf{q}_0$. It is easy to veryify that

$$\lim_{\varepsilon \to 0} \mathbf{K}^\varepsilon(t) = \mathbf{K}^o(t), \text{ and} \qquad\qquad \lim_{\varepsilon \to 0} \mathbf{q}^\varepsilon(t) = \mathbf{q}^o(t).$$

Hence, the state trajectories $\mathbf{q}^\varepsilon(t)$ and $\mathbf{q}^o(t)$ become defined as

$$\begin{aligned}
\dot{\mathbf{q}}^\varepsilon &= \hat{\mathbf{f}}(\mathbf{q}^\varepsilon, \mathbf{K}^\varepsilon) + \hat{\mathbf{g}}(\mathbf{q}^\varepsilon, \mathbf{K}^\varepsilon), \;\; \mathbf{q}^\varepsilon(t_0) = \mathbf{q}_0, \; t \in I, \text{ and} \\
\dot{\mathbf{q}}^o &= \hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}^o) + \hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}^o), \;\; \mathbf{q}^o(t_0) = \mathbf{q}_0, \; t \in I.
\end{aligned}$$

Subtracting one from the other leads to the following relation

$$\begin{aligned}
\dot{\mathbf{q}}^\varepsilon - \dot{\mathbf{q}}^o = \hat{\mathbf{f}}(\mathbf{q}^\varepsilon, \mathbf{K}^o) - \hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^\varepsilon, \mathbf{K}^o) - \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K}^o) \\
+ \varepsilon \left[\hat{\mathbf{f}}(\mathbf{q}^\varepsilon, \mathbf{K} - \mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^\varepsilon, \mathbf{K} - \mathbf{K}^o)\right].
\end{aligned} \tag{6.13}$$

Defining

$$\boldsymbol{\eta}(t) \equiv \lim_{\varepsilon \to 0} \left(\frac{\mathbf{q}^\varepsilon(t) - \mathbf{q}^o(t)}{\varepsilon} \right),$$

it follows from (6.13) that $\boldsymbol{\eta}(t)$ must satisfy the following initial value problem

$$\begin{aligned}
\dot{\boldsymbol{\eta}} = \left[\frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{q}}(\mathbf{q}^o, \mathbf{K}^o) + \frac{\partial \hat{\mathbf{g}}}{\partial \mathbf{q}}(\mathbf{q}^o, \mathbf{K}^o) \right] \boldsymbol{\eta} + \hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K} - \mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K} - \mathbf{K}^o), \text{ and} \\
\boldsymbol{\eta}(t_0) = 0.
\end{aligned} \tag{6.14}$$

Equation (6.14) is linear and non-homogeneous with $\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o)$ being the driving force. As a result, it has a continuous solution $\boldsymbol{\eta}(t) \in C(I, \mathbb{R}^3)$, which is continuously dependent on $\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o)$.

By definition of Gateaux (directional) derivative we can derive the following expression

$$\begin{aligned} Tr\left[(\mathbf{K}-\mathbf{K}^o)^T\, dJ(\mathbf{K}^o, \boldsymbol{\zeta}_c)\right] &= \lim_{\varepsilon \to 0} \frac{J(\mathbf{K}^o + \varepsilon(\mathbf{K}-\mathbf{K}^o), \boldsymbol{\zeta}_c) - J(\mathbf{K}^o, \boldsymbol{\zeta}_c)}{\varepsilon} \\ &= \boldsymbol{\eta}^T(t)\frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t)\frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}(t)]dt. \end{aligned}$$

Hence, inequality (6.12) yields

$$\boldsymbol{\eta}^T(t)\frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t)\frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}(t)]dt \geq 0. \tag{6.15}$$

Since $\boldsymbol{\eta}(t)$ of the variational equation (6.14) is continuously dependent on $\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o)$, the map

$$\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o) \longmapsto \boldsymbol{\eta}(t), \quad t \in I$$

has to be continuous from $\mathcal{L}_1(I, \mathbb{R}^3)$ to $C(I, \mathbb{R}^3)$ [Ahmed, 2006, p. 260]. The map

$$\boldsymbol{\eta}(t) \longmapsto \boldsymbol{\eta}^T(t)\frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t)\frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}(t)]dt$$

is then a continuous linear functional on $C(I, \mathbb{R}^3)$. Thus, the composition map

$$\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o) \longmapsto \boldsymbol{\eta}^T(t)\frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t)\frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}(t)]dt$$

is a continuous linear functional on $\mathcal{L}_1(I, \mathbb{R}^3)$, where $\left[\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K}-\mathbf{K}^o)\right] \in \mathcal{L}_1(I, \mathbb{R}^3)$. Therefore, by the Riesz representation theorem or by the duality [Royden and

Fitzpatrick, 2010] between $\mathcal{L}_1(I,\mathbb{R}^3)$ and $\mathcal{L}_\infty(I,\mathbb{R}^3)$, we may conclude that there exists an element $\boldsymbol{\psi}^o \in \mathcal{L}_\infty(I,\mathbb{R}^3)$ such that

$$\begin{aligned} Tr\left[(\mathbf{K}-\mathbf{K}^o)^T\, dJ(\mathbf{K}^o,\boldsymbol{\zeta}_c)\right] &= \boldsymbol{\eta}^T(t)\frac{\partial\phi}{\partial\mathbf{q}}[t_f,\mathbf{q}(t_f)] + \int_{t_0}^{t_f}\boldsymbol{\eta}^T(t)\frac{\partial\ell}{\partial\mathbf{q}}[t,\mathbf{q}(t)]dt \\ &= \int_{t_0}^{t_f}(\boldsymbol{\psi}^o)^T\left[\hat{\mathbf{f}}(\mathbf{q}^o,\mathbf{K}-\mathbf{K}^o)+\hat{\mathbf{g}}(\mathbf{q}^o,\mathbf{K}-\mathbf{K}^o)\right]dt. \end{aligned} \tag{6.16}$$

It follows from inequality (6.15) that $\forall\ \mathbf{K}\in\mathcal{K}_{ad}$,

$$\int_{t_0}^{t_f}(\boldsymbol{\psi}^o)^T\left[\hat{\mathbf{f}}(\mathbf{q}^o,\mathbf{K}-\mathbf{K}^o)+\hat{\mathbf{g}}(\mathbf{q}^o,\mathbf{K}-\mathbf{K}^o)\right]dt \geq 0\,. \tag{6.17}$$

Using the variational equation (6.14) and (6.16)

$$\begin{aligned} &\boldsymbol{\eta}^T(t)\frac{\partial\phi}{\partial\mathbf{q}}[t_f,\mathbf{q}(t_f)] + \int_{t_0}^{t_f}\boldsymbol{\eta}^T(t)\frac{\partial\ell}{\partial\mathbf{q}}[t,\mathbf{q}(t)]dt = \\ &\int_{t_0}^{t_f}(\boldsymbol{\psi}^o)^T\left[\dot{\boldsymbol{\eta}}-\left\{\frac{\partial\hat{\mathbf{f}}}{\partial\mathbf{q}}(\mathbf{q}^o,\mathbf{K}^o)+\frac{\partial\hat{\mathbf{g}}}{\partial\mathbf{q}}(\mathbf{q}^o,\mathbf{K}^o)\right\}\boldsymbol{\eta}\right]dt. \end{aligned} \tag{6.18}$$

Integrating by parts and since $\boldsymbol{\eta}(t_0)=0$,

$$\begin{aligned} \int_{t_0}^{t_f}(\boldsymbol{\psi}^o)^T\left[\dot{\boldsymbol{\eta}}-\left\{\frac{\partial\hat{\mathbf{f}}}{\partial\mathbf{q}}(\mathbf{q}^o,\mathbf{K}^o)+\frac{\partial\hat{\mathbf{g}}}{\partial\mathbf{q}}(\mathbf{q}^o,\mathbf{K}^o)\right\}\boldsymbol{\eta}\right]dt &= \boldsymbol{\eta}^T(t_f)\boldsymbol{\psi}^o(t_f)+ \\ \int_{t_0}^{t_f}\boldsymbol{\eta}^T\left[-\dot{\boldsymbol{\psi}}^o-\left\{\left(\frac{\partial\hat{\mathbf{f}}}{\partial\mathbf{q}}(\mathbf{q}^o,\mathbf{K}^o)\right)^T+\left(\frac{\partial\hat{\mathbf{g}}}{\partial\mathbf{q}}(\mathbf{q}^o,\mathbf{K}^o)\right)^T\right\}\boldsymbol{\psi}^o\right]dt. \end{aligned}$$

Expression (6.18) can now be written as

$$\begin{aligned} \boldsymbol{\eta}^T(t)\frac{\partial\phi}{\partial\mathbf{q}}[t_f,\mathbf{q}(t_f)] + \int_{t_0}^{t_f}\boldsymbol{\eta}^T(t)\frac{\partial\ell}{\partial\mathbf{q}}[t,\mathbf{q}(t)]dt &= \boldsymbol{\eta}^T(t_f)\boldsymbol{\psi}^o(t_f)+ \\ \int_{t_0}^{t_f}\boldsymbol{\eta}^T\left[-\dot{\boldsymbol{\psi}}^o-\left\{\left(\frac{\partial\hat{\mathbf{f}}}{\partial\mathbf{q}}(\mathbf{q}^o,\mathbf{K}^o)\right)^T+\left(\frac{\partial\hat{\mathbf{g}}}{\partial\mathbf{q}}(\mathbf{q}^o,\mathbf{K}^o)\right)^T\right\}\boldsymbol{\psi}^o\right]dt, \end{aligned} \tag{6.19}$$

and so

$$\begin{aligned}\dot{\boldsymbol{\psi}}^o = &-\left\{\left(\frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{q}}(\mathbf{q}^o, \mathbf{K}^o)\right)^T + \left(\frac{\partial \hat{\mathbf{g}}}{\partial \mathbf{q}}(\mathbf{q}^o, \mathbf{K}^o)\right)^T\right\}\boldsymbol{\psi}^o \\ &- \frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}^o(t)], \quad \boldsymbol{\psi}^o(t_f) = \frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)].\end{aligned} \tag{6.20}$$

This costate dynamics equation is linear along the optimal trajectories. Thus, the necessary conditions of optimality are given by the integral inequality (6.17), the costate dynamics (6.20), and the state equation (6.3). In other words, the choice of $\mathbf{K} \in \mathcal{K}_{ad}$ determines the optimality conditions (6.17), (6.20), and (6.3).

Reformulating the optimality condition (6.17) as

$$\int_{t_0}^{t_f} (\boldsymbol{\psi}^o)^T \left[\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K})\right] dt \geq \int_{t_0}^{t_f} (\boldsymbol{\psi}^o)^T \left[\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K}^o)\right] dt, \ \forall\, \mathbf{K} \in \mathcal{K}_{ad}, \tag{6.21}$$

it becomes easy to derive the following point-wise inequality [Ahmed, 2006]

$$(\boldsymbol{\psi}^o)^T \left[\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K})\right] \geq (\boldsymbol{\psi}^o)^T \left[\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K}^o)\right], \ \forall\, \mathbf{K} \in \mathcal{K}_{ad}. \tag{6.22}$$

Adding the term $\ell[t, \mathbf{q}^o(t)]$ in both sides, yields

$$(\boldsymbol{\psi}^o)^T \left[\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K})\right] + \ell[t, \mathbf{q}^o(t)] \geq (\boldsymbol{\psi}^o)^T \left[\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}^o) + \hat{\mathbf{g}}(\mathbf{q}^o, \mathbf{K}^o)\right] + \ell[t, \mathbf{q}^o(t)],$$

which gives the following Hamiltonian inequality

$$\mathcal{H}[\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}(t)] \geq \mathcal{H}[\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)].$$

This is the same as inequality (6.9) stated in the theorem. Differentiating $\mathcal{H}[\cdot]$ in (6.7) with

respect to the costate variable $\boldsymbol{\psi}$, we get

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{\psi}}[\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)] = \hat{\mathbf{f}}[\mathbf{q}^o(t), \mathbf{K}^o(t)] + \hat{\mathbf{g}}[\mathbf{q}^o(t), \mathbf{K}^o(t)],$$

which leads to the state equation

$$\dot{\mathbf{q}}^o = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\psi}}[\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)], \quad \mathbf{q}^o(t_0) = \mathbf{q}_0,$$

as defined in (6.10).

Differentiating the Hamiltonian with respect to the state variable $\mathbf{q}$ yields

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial \mathbf{q}}[\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^{\mathbf{o}}(t)] &= \left\{ \left(\frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{q}}(\mathbf{q}^o, \mathbf{K}^o) \right)^T + \left(\frac{\partial \hat{\mathbf{g}}}{\partial \mathbf{q}}(\mathbf{q}^o, \mathbf{K}^o) \right)^T \right\} \boldsymbol{\psi}^o + \frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}^o(t)] \\ &= -\dot{\boldsymbol{\psi}}^o.\end{aligned}$$

Hence, the costate dynamics (6.20) can be expressed in terms of the Hamiltonian as

$$\dot{\boldsymbol{\psi}}^o = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}[\mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)], \ \boldsymbol{\psi}^o(t_f) = \frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)],$$

which is condition (6.11). □

Theorem 6.1 states that the feedback control gain $\mathbf{K}^o \in \mathcal{K}_{ad}$ effectively determines optimal control inputs for its actuator. In order to solve for $\mathbf{K}^o$, we express the gradient of the Hamiltonian defined in (6.7) as

$$\frac{\partial \mathcal{H}}{\partial \mathbf{K}} = \frac{r}{2} \left[(\mathbf{B}(\mathbf{q}))^T \boldsymbol{\psi} \hat{\mathbf{h}}^T(\mathbf{q}) + (\mathbf{B}(\mathbf{q}))^T \boldsymbol{\psi} \boldsymbol{\zeta}^T \right] \tag{6.23}$$

The gradient $\mathcal{H}_{\mathbf{K}} \equiv \frac{\partial \mathcal{H}}{\partial \mathbf{K}}$ indicates the direction of convergence for the optimal trajectory of (6.3).

Having defined $\mathbf{q}(t)$ in (6.8) and $\boldsymbol{\psi}(t)$ in (6.11) for $t \in I$, the optimal $\mathbf{K}$ can be found by equating the gradient (6.23) to zero, *i.e.*,

$$\frac{r}{2}\left[(\mathbf{B}(\mathbf{q}))^T \boldsymbol{\psi}\hat{\mathbf{h}}^T(\mathbf{q}) + (\mathbf{B}(\mathbf{q}))^T \boldsymbol{\psi}\boldsymbol{\zeta}^T\right] = 0. \tag{6.24}$$

Note that expression (6.24) is independent of the gain matrix $\mathbf{K}$. Hence, the problem boils down to finding $\mathbf{K}(t),\ t \in I$, such that the robot's equation of motion (6.8) and the costate trajectory from (6.11), satisfy (6.24). The optimal feedback control gain $\mathbf{K}^o$ can be determined by satisfying the Hamiltonian inequality (6.9). In other words, $\mathbf{K}$ is to be adaptively tuned to minimize the robot's tracking error.

Corollary 6.1 (Adapting the gain $\mathbf{K}$)**.** *Consider the robot's feedback system* (6.3) *defined over the time horizon* I*. Adapting the gain* $\mathbf{K}$ *according to the following offline update rule*

$$\mathbf{K}^{new} = \mathbf{K}^{old} - \epsilon\mathcal{H}_{\mathbf{K}},\ \textit{for}\ 0 < \epsilon < 1 \tag{6.25}$$

satisfies the Hamiltonian inequality (6.9) *and, hence, guarantees the converge of the robot's trajectory towards its target.*

Proof. Let

$$\mathbf{K}^* = \mathbf{K} - \epsilon\mathcal{H}_{\mathbf{K}}\,, \tag{6.26}$$

for some $\mathbf{K} \in \mathbb{R}^{2\times s}$ and $0 < \epsilon < 1$. The corresponding Hamiltonian is

$$\mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K}^*) = \mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K} - \epsilon\mathcal{H}_{\mathbf{K}})\,,$$

where $\mathbf{q}^*$ and $\boldsymbol{\psi}^*$ are the robot's pose and the costate variable corresponding to $\mathbf{K}^*$. Taking

Taylor series expansion of the right hand side about $\mathbf{K}$, we get

$$\mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K}^*) = \mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K}) + Tr\left[\mathcal{H}_{\mathbf{K}}^T(\mathbf{K}^* - \mathbf{K})\right] + \mathcal{O}(\varepsilon)\,.$$

Neglecting the higher order terms of the above expression yields

$$\begin{aligned}\mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K}^*) &\equiv \mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K}) - \epsilon Tr\left[\mathcal{H}_{\mathbf{K}}^T \mathcal{H}_{\mathbf{K}}\right] \\ &= \mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K}) - \epsilon \|\mathcal{H}_{\mathbf{K}}\|^2 \\ &< \mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K})\end{aligned}$$

since $\epsilon\|\mathcal{H}_{\mathbf{K}}\|^2 > 0$. Hence, update rule (6.26) guarantees that the Hamiltonian $\mathcal{H}$ is monotonically decreasing, which proves that the adaptation law (6.25) eventually leads to the optimal control gain $\mathbf{K}^o$. □

Remark 6.1. *Theoretically, if* $\mathbf{K}^*$ *is an arbitrary feedback control gain that satisfies the Hamiltonian inequality* (6.9)

$$\mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K}^*) \leq \mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K}),\ \forall\, \mathbf{K} \in \mathcal{K}_{ad},$$

then the update rule

$$\mathbf{K}^{'} = \mathbf{K}^* + \varepsilon(\mathbf{K} - \mathbf{K}^*),\ \forall\, \mathbf{K} \in \mathcal{K}_{ad} \text{ and } \varepsilon \in [0, 1] \tag{6.27}$$

yields the inequality (6.9)

$$\mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K}^*) \leq \mathcal{H}(t, \mathbf{q}^*, \boldsymbol{\psi}^*, \mathbf{K}^* + \varepsilon(\mathbf{K} - \mathbf{K}^*)). \tag{6.28}$$

By taking the Taylor series expansion of the right hand side of (6.28) *about* $\mathbf{K}^*$*, we get*

$$\begin{aligned}\mathcal{H}(t,\mathbf{q}^*,\boldsymbol{\psi}^*,\mathbf{K}^*) \leq \mathcal{H}(t,\mathbf{q}^*,\boldsymbol{\psi}^*,\mathbf{K}^*)+ \\ \varepsilon Tr\left[\mathcal{H}_{\mathbf{K}}^T(\mathbf{K}-\mathbf{K}^*)\right]+\mathcal{O}(\varepsilon),\end{aligned} \tag{6.29}$$

where $\mathcal{O}(\varepsilon)$ is the higher order terms of the Taylor series. Neglecting the higher order terms of (6.29)*, we can write*

$$\begin{aligned}0 \leq \varepsilon Tr\left[\mathcal{H}_{\mathbf{K}}^T(\mathbf{K}-\mathbf{K}^*)\right] = \varepsilon Tr\left[(\mathbf{K}-\mathbf{K}^*)^T\mathcal{H}_{\mathbf{K}}\right], \\ \Rightarrow Tr\left[(\mathbf{K}^*)^T\mathcal{H}_{\mathbf{K}}\right] \leq Tr\left[\mathbf{K}^T\mathcal{H}_{\mathbf{K}}\right],\end{aligned} \tag{6.30}$$

which shows that $\mathbf{K}^*$ in (6.27) *is the optimal feedback gain. However, the problem of the update rule* (6.27) *is to find proper feedback gain $\mathbf{K}$ that satisfies inequality* (6.30)*. Hence, the update rule* (6.25) *in Corollary 6.1 is one way to find the optimal feedback gain $\mathbf{K}^o$.*

In the following, we numerically solve for the gain $\mathbf{K}$ such that (6.24) is satisfied, aggregating the components described earlier.

Output (Measurement) Feedback Control Algorithm:

Let $\mathbf{K}_i \equiv \mathbf{K}_i(t)$, $t \in I$, be the gain at the i-th iteration of the optimization procedure.

Step 0 (initialization): Subdivide the time interval $I \equiv [t_0, t_f]$ into N subintervals. Assume a piecewise-contant $\mathbf{K_i}(t) = \mathbf{K_i}(t_k)$, $t \in [t_k, t_{k+1}]$, for $k = 0, \ldots, k_f - 1$, and choose any $\boldsymbol{\zeta} \in \mathcal{B}(\bar{\boldsymbol{\zeta}}, r')$.

Find the optimal gain $\mathbf{K}^o$ by repeating Steps 1–5 until the stopping criterion in Step 5 is met.

Step 1: Integrate the robot's feedback system (6.3) as in (6.8) with $\mathbf{K} \equiv \mathbf{K_i}(t)$, $t \in I$.

Step 2: Solve costate equation (6.11) backward for $\boldsymbol{\psi}_i$.

Step 3: Define the Hamiltonian $\mathcal{H}(\mathbf{q}_i, \boldsymbol{\psi}_i, \mathbf{K}_i)$ as in (6.7).

Step 4: Compute the cost function $J(\mathbf{K}_i, \boldsymbol{\zeta})$ using (6.5), the gradients of the Hamiltonian $\mathcal{H}_{\mathbf{K}}$ using (6.23), and its corresponding intergrated norm $\int_{t_0}^{t_f} \|\mathcal{H}_{\mathbf{K}}\|^2 dt$.

Step 5: If $J(\mathbf{K}_i, \boldsymbol{\zeta}) \leq \delta_1$ or $\int_{t_0}^{t_f} \|\mathcal{H}_{\mathbf{K}}\|^2 dt \leq \delta_2$, for pre-defined small positive tolerance constants δ_1 and δ_2, then $\mathbf{K_i}$ is regarded close enough to its nominal optimal value, and so the algorithm is halted.

Otherwise, use the following update rule to adjust the piecewise-constant feedback control gain:

$$\mathbf{K}_{i+1}(t_k) = \mathbf{K}_i(t_k) - \epsilon\, \mathcal{H}_{\mathbf{K}(t_k)} + \lambda\, \Delta\mathbf{K}_i(t_k)$$

$$\Delta\mathbf{K}_i(t_k) = \mathbf{K}_i(t_k) - \mathbf{K}_{i-1}(t_k)$$

for $k = 0, \ldots, k_f - 1$, where ϵ and λ are the step size and the momentum constant (for faster convergence), respectively.

6.5 Simulation Results

This section illustrates the performance of the proposed trajectory tracking system using the circular shaped differential drive virtual mobile robot described in section 3.3.3 of Chapter 3.

The rotational speeds of the right and left wheels are constrained by $|u_R| \leq u_R^{max} = 12.1\,\mathrm{rad{\cdot}s^{-1}}$, and $|u_L| \leq u_L^{max} = 12.1\,\mathrm{rad{\cdot}s^{-1}}$. The size of the virtual test environment is $12 \times 12 \times 3$ m high.

A mesh of nine RFID tags is mounted on the ceiling. The orthogonal projection of the tags' locations on the ground are shown in Figures 6.4(a) and 6.5(a). Note that these locations are unknown to the robot. The performance metric adopted in the current work is the tracking error between the robot's desired and actual trajectories over the time period of $I \equiv [0, 60]$ s, as defined in (3.11). Initially, the feedback control gain $\boldsymbol{K}$ is chosen as:

$$\boldsymbol{K}(t) = 10^{-6} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \qquad t \in I.$$

The sampling time period is set to 0.1 s. The weighting matrices of the cost function (6.5) are chosen as $\mathbf{P}(t_f) = diag(0.5, 0.5, 1)$ and $\mathbf{Q}(t) = diag(1, 1, 2)$, $\forall\, t \in I$. Hence, trajectory tracking is regarded twice as important as just reaching the final destination.

6.5.1 Modelling RSS Measurements and Noise

To make the controller's simulation as realistic as possible, the RSS signals were experimentally measured by emulating the RFID system using an XBee Pro RF module and a MakeController board (see Figure 4.5 in Chapter 4). For that, the XBee Pro RF module (emulating the RFID tag) was attached on a 3-m high ceiling in our research laboratory while the MakeController board (emulating the RFID reader) was mounted on the robot. Then, the robot was manually placed in various positions on the ground. For each position, the average RSS measurements transmitted by the XBee module is recorded as received by the MakeController board. Table 6.1 shows some sample RSS measurements logged in this experiment. Since the purpose of this exercise is to use these measurements as semi-ground truth for the noise-free signal model,

Table 6.1: Sample RSS measurements from XBee Pro module.

Distance (d) m	RSS (dBm)
3.00	-47
3.13	-48
3.50	-51
4.04	-62
4.69	-54
5.41	-64
6.18	-69

precautions were taken to reduce the effect of reverberations from the ground and the far-end walls. Fitting the data to model (6.2) yielded the parameter values $\alpha = -35.5$ and $\beta = 0.1071$.

In order to match real-world scenarios where the robot's workspace may be populated with reverberant obstacles, such as walls, metal file cabinets, appliances, etc., this model was superimposed with an exaggerated Gaussian noise $\zeta(t)$ (see (6.1)) defined by

$$\Pr[\zeta(t)] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\left\{\zeta(t) - \bar{\zeta}\right\}^2}{2\sigma^2}\right],$$

with a mean $\bar{\zeta} = -30$ dBm and a standard deviation $\sigma = 50$ dBm. This yielded a signal-to-noise ratio of -179.45 dBm. This weak signal-to-noise ratio was adopted on purpose to better demonstrate later the performance of the proposed controller. Figure 6.3 shows the "noise-free" and noisy RSS signals obtained. Fitting the resultant noisy RSS signal again to model (6.2) led to the parameter values of $\alpha = -60$ and $\beta = 0.2$. This model of the noisy signal was used in the following controller's simulations.

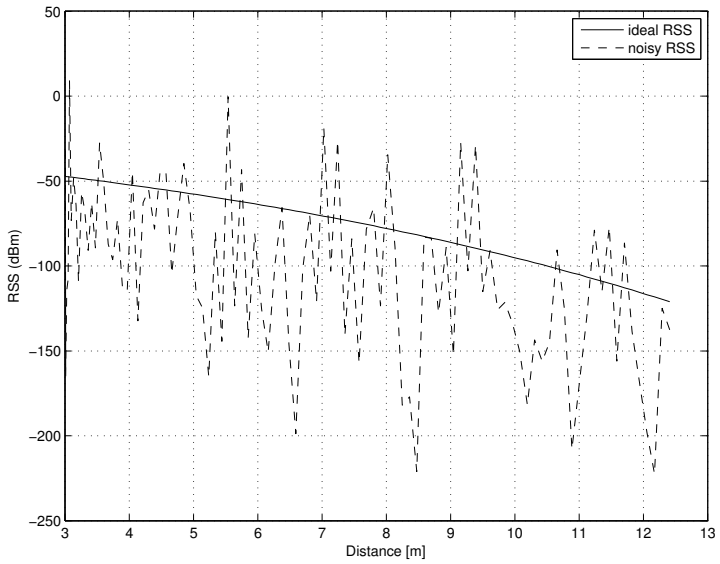


Figure 6.3: Noise model considered for the simulation.

6.5.2 Generating Desired State Trajectory

The desired Cartesian trajectory $(x^d(t), y^d(t))$ is feasible when it satisfies the robot's desired (reference) model from (3.6), *i.e.*,

$$\dot{\mathbf{q}}^d(t) = \mathbf{f}[\mathbf{q}^d(t), \mathbf{u}^d(t)], \tag{6.31}$$

where $\mathbf{q}^d(t) = [x^d(t)\ y^d(t)\ \theta^d(t)]^T$ is the desired state of the robot with the suitable initial condition $\mathbf{x}^d(0) = [x^d(0)\ y^d(0)\ \theta^d(0)]^T$. We first solve for the robot's linear and angular velocities $(\nu^d(t),\ \omega^d(t))$ by following the procedure illustrated in [Luca et al., 2000] which is repeated herein for convenience. The desired trajectory $(x^d(t), y^d(t))$ must satisfy the following kinematic equations:

$$\begin{bmatrix} \dot{x}^d(t) \\ \dot{y}^d(t) \\ \dot{\theta}^d(t) \end{bmatrix} = \begin{bmatrix} \cos\theta^d(t) & 0 \\ \sin\theta^d(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu^d(t) \\ \omega^d(t) \end{bmatrix},$$

Solving for $\nu^d(t)$ and $\omega^d(t)$ from the above expressions yields $\nu^d(t) = \pm\sqrt{[\dot{x}^d(t)]^2 + [\dot{y}^d(t)]^2}$,

$$\theta^d(t) = \text{ATAN2}\left\{\frac{\dot{y}^d(t)}{\nu^d(t)}, \frac{\dot{x}^d(t)}{\nu^d(t)}\right\}, \text{ and}$$
$$\omega^d(t) = \dot{\theta}^d(t) = \left\{\frac{[\ddot{y}^d(t)\dot{x}^d(t) - \ddot{x}^d(t)\dot{y}^d(t)]}{[\nu^d(t)]^2}\right\}$$

Note that the robot's desired orientation $\theta^d(t) \in (-\pi, \pi]$. Hence, $(x^d(t), y^d(t), \theta^d(t))$, for $t \in I$, is the desired state trajectory of the robot, which is the solution of the desired model (6.31).

6.5.3 Tracking a Rectilinear Trajectory

The first test was aimed to assess the performance of the proposed tracking system to guide a robot along a linear trajectory defined by $x^d(t) = [8t/60] - 4$, $y^d(t) = [\tan(\theta^d)]x^d(t)$, and $\theta^d(t) = \pi/4$ for $t \in [0, 60]$ s. The robot is initially placed at position $(-3.5, -4)$ m with an orientation of $0°$. The results are summarized in Figure 6.4. As expected, the controller tributes a higher priority to converging to the desired trajectory before heading to the target point. This can be clearly seen in Figure 6.4(a), where the hollow and solid arrows indicate the robot's initial and final poses, respectively. Despite the excessively noisy RSS signals received from the tags, the proposed control scheme could effectively track the desired trajectory with an insignificant tracking error Figure 6.4(b). Figure 6.4(c) reveals that once the robot converged to its target trajectory, it maintained a tracking error of less than 0.05 m until the end of the simulation.

The percentile plot in the same figure shows that the tracking error was less than 0.05 m for about 75% of the navigation time and less than 0.2 m of about 90% of the time. It can be clearly seen from this figure that the higher end of the percentile curve is mainly due to the relatively large initial error of 0.5 m. The average tracking error throughout the whole trajectory ($\mathscr{E}/60$ of (3.11)) is 0.07 m.

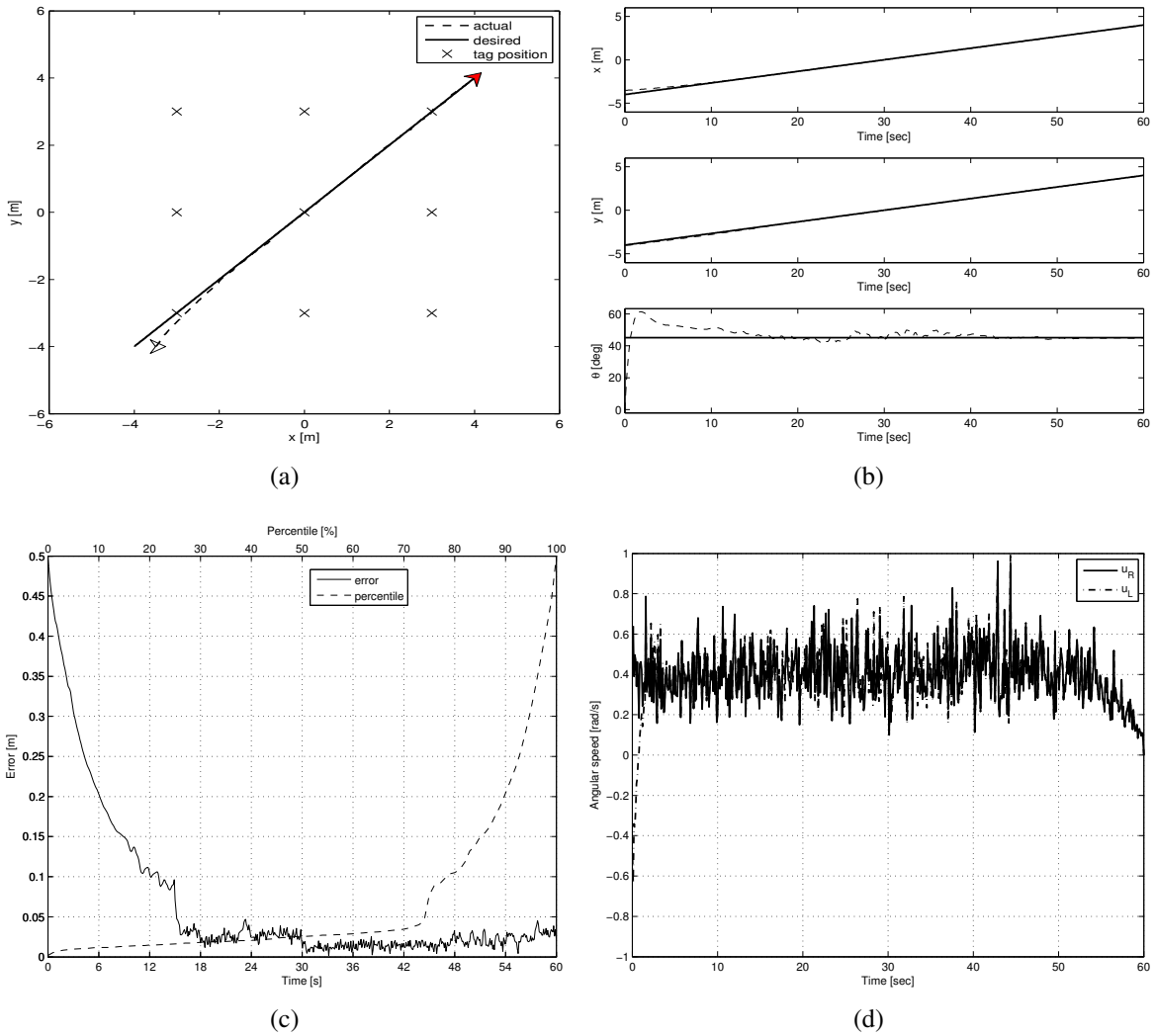


Figure 6.4: Controller's performance in following a rectilinear trajectory: (a) robot's trajectory, (b) pose, (c) error, and (d) control inputs.

6.5.4 Tracking a Curvilinear Trajectory

The purpose of this test scenario is to study the robot's tracking ability along a more complex path. To do that, we define a curvilinear desired trajectory as $x^d(t) = (6/60)t - 3$, $y^d(t) = 3\sin[1.2x^d(t)]$, and $\theta^d(t) = \text{ATAN2}\left\{\frac{\dot{y}^d(t)}{\nu^d(t)}, \frac{\dot{x}^d(t)}{\nu^d(t)}\right\}$. The robot is initially placed at $(-2.5, 1.5)\,\text{m}$ with an initial orientation of $0°$. The results are demonstrated in Figure 6.5. Despite the large initial error of $0.53\,\text{m}$, the robot converged to within $0.1\,\text{m}$ of the target trajectory in about $6\,\text{s}$ (Figure 6.5(c)). Since then, it maintained a tracking error of less than $0.15\,\text{m}$ till the end of the simulation. The percentile plot in Figure 6.5(c) shows that the tracking error was less than $0.1\,\text{m}$ for 80% of the time. This could have been better if not for the large initial error, which was set so on purpose to test the robot's convergence speed. The total distance traveled by the robot in this simulation is $16.4\,\text{m}$. The average tracking error throughout the whole trajectory was recorded to be $0.08\,\text{m}$. It is important to point out here that one of the salient features of the proposed navigation system is that its tracking error is non-cumulative (independent of the total traveled distance). This can be clearly noticed from the tracking error plot in Figure 6.5(c).

6.5.5 Analysis

Both simulations demonstrate a satisfactory accuracy level of the suggested control scheme despite the weak signal-to-noise ratio characterizing the RSS signals received by the RFID reader. It is worth mentioning that this controller is also able to stabilize the robot at a fixed configuration by simply tuning the weight matrices ($\mathbf{P}(t_f)$ and $\mathbf{Q}(t)$) of the cost function (6.5). This is in contrast to many recent RFID-based techniques which usually tackle the localization problem only [Bekkali and Matsumoto, 2009, Di Giampaolo and Martinelli, 2012, Park and Hashimoto, 2009]. The localization accuracy reported therein is in the range of 0.1–$0.5\,\text{m}$, despite neglecting the effect of reverberations and low signal-to-noise ratios.

The grid of RFID tags attached to the ceiling with one tag every $3\,\text{m}$, ensured that the robot

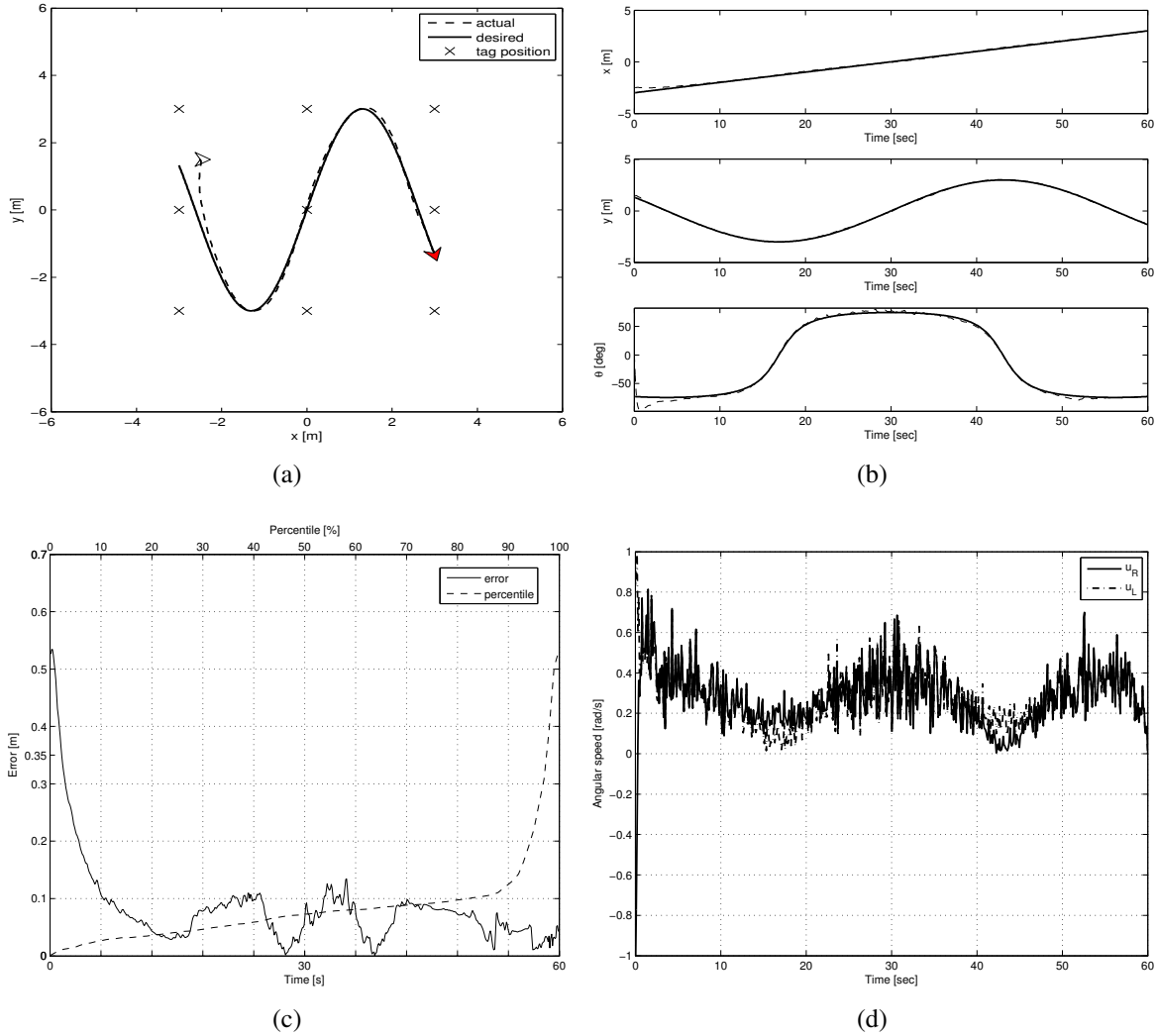


Figure 6.5: Controller's performance in following a curvilinear trajectory: (a) robot's trajectory, (b) pose, (c) error, and (d) control inputs.

always receives RF signals whose RSS values are within the range of the sampled data (see Figure 6.3). The size of the grid (i.e., number of tags) and the inter-tag distance depend on the size of the robot's workspace and the communication range between the tags and the reader. In practical applications, these parameters should be chosen as to guarantee that, at any given time, the robot can receive "meaningful" RSS values from a minimum number of tags (9 in this application). In this context, a "meaningful" RSS value is one that is within the range used in the calibration process of the RF signal (Figure 6.3 in this case). This is not to be confused with trilateration-based localization methods, because as explained in the previous sections, they are not used in this controller.

6.6 Summary

This chapter describes a novel trajectory tracking system for differential drive mobile robots using an optimal output feedback control technique. The feedback control is generated using exaggerated noisy RSS measurements from RFID tags placed in the robot's operating environment. The controller is, however, a generalized output feedback controller in the sense that it opens the door for solving problems of a general class of nonlinear dynamic systems. It is important to mention that the mobile robot model is not required to be linearized to follow a certain desired path as most existing techniques do [Chen et al., 2009, Chwa, 2010]. The robot's control actions is obtained by optimizing the feedback control gain using the gradient descent technique, where no dynamic linearization is performed. The performance of the proposed controller is clearly demonstrated in numerical simulations. A rectilinear trajectory, and a sinusoidal trajectory are followed by the mobile robot with a sufficiently small tracking error. An important point that can be made from the simulation results is that the robot's tracking performance depends on the number of RFID tags deployed in it's operating environment. Increasing the number of RFID tags would certainly improve the tracking performance because redundant measurements would

decrease the effect of noise and so describe the robots state more accurately. In this navigation scheme, RFID tag locations are not known to the robot, however, these tags must be placed within the operating range of the RFID reader which is mounted on the robot. As a matter of fact, a mesh of nine tags are deployed in the environment in order to reflect a more realistic approach of how the proposed technology would be used for navigating a mobile robot in a real-world application. The proposed optimal controller is modular in the sense that it can converge any dynamic system to a desired response. A practical realization of this output feedback controller would be possible if the feedback control gain is optimized locally which basically suffers from high computational complexity. The high computation overhead in this controller stems from the fact that more measurements increase the dimension of the measurement model which lead the measurement feedback model quite challenging to solve in terms of computational power.

Chapter 7

On-line Neighboring Optimal Control for Mobile Robot Trajectory Tracking

7.1 Introduction

Recall that Chapter 5 presented a robot navigation technique where the measurement noise was taken into consideration for navigating through pre-specified points on the ground defined by the RFID tags in the workspace. At any time instant, the RSS measurements were simply passed to the FLC for generating control inputs to the robot's actuators. These inputs, however, were not explicitly constrained on time to navigate along the desired points. In addition, the robot can not follow continuous nonlinear trajectories as it depends on the RFID tag locations. The previous chapter resolves this issue by designing an output feedback control law where control inputs are generated based on the RSS measurements from customized RFID tags. However, since the robot itself is subject to external disturbances (*i.e.*, process noise) that can not be specified ahead of time, generating feedback control based on RSS measurements as described in Chapter 6 can still be further improved for more satisfactory tracking performance. Moreover, high computational resources are required for the robot to track a pre-defined trajectory in

real-life.

This chapter resolves the problems addressed in the previous two chapters. For that, the solution of the robot's tracking problem is partitioned into two parts: the nominal (deterministic) solution and the neighboring solution. The former represents an off-line (prior) solution and the latter represents an on-line (real-time) solution [Stengel, 1994]. Hence, the resulting optimal trajectory for the mobile robot could be well approximated by the sum of the nominal optimal trajectory, computed with no process noise and full-state feedback (*i.e.*, perfect measurement), plus a neighboring optimal trajectory.

The rest of the chapter is organized as follows. Section 7.2 illustrates how to determine the nominal optimal trajectory and its corresponding nominal inputs which are applied to the robot's actuators. The principles of neighboring optimal control and that of optimal state estimation of a mobile robot are concatenated to determine the robot's actual optimal trajectory. This is discussed in detail in section 7.3. The formulation of a conventional Simultaneous Localization and Mapping (SLAM) method for estimating the robot's pose is given in section 7.4. Section 7.5 shows some numerical results that demonstrate the performance of the robot's nominal and neighboring optimal controller. Two experiments showing the robot's performance in real-time environments are illustrated in section 7.6. In order to compare the robot's pose estimation (not navigation) performance of the proposed neighboring optimal controller, section 7.7 presents the pose estimation results using the SLAM method. Finally, section 7.8 gives the summary of this chapter.

7.2 Nominal Pose and Control Generation

In order to determine the robot's nominal optimal pose and its corresponding control input, assume that the process noise $\boldsymbol{\xi}(t) = \mathbf{0}$ and also define the robot's control input (3.9) as the

full-state feedback control:

$$\mathbf{u}(t) = \mathbf{K}(t)\mathbf{q}(t), \tag{7.1}$$

subject to (3.8), where $\mathbf{K}(t) \neq \mathbf{0}$ is the feedback control gain for the DDMR model (3.7). Since the sets $\mathcal{U}$ and $\mathcal{Q}$ are convex, $\mathbf{K}(t)$ must be chosen from a convex set $\mathcal{K} \subset \mathbb{R}^{2\times 3}$. Furthermore, due to the constraint on the wheel speeds, the gain $\mathbf{K}(t)$ has to be chosen from the admissible matrix space $\mathcal{K}_{ad} \subset \mathcal{K}$.

Substituting (7.1) in (3.7), yields the following full-state feedback system:

$$\dot{\mathbf{q}}(t) = \frac{r}{2}\mathbf{B}[\mathbf{q}(t)]\mathbf{K}(t)\mathbf{q}(t) = \hat{\mathbf{f}}[\mathbf{q}(t), \mathbf{K}(t)],\ \mathbf{q}(t_0) = \mathbf{q}_0, \tag{7.2}$$

where $\mathbf{q}_0 \neq \mathbf{0}$ since model (7.2) is a nonlinear homogeneous equation.

The block diagram of the proposed feedback system is shown in Figure 7.1. The optimal controller block takes the tracking error and the feedback control gain into account in order to generate optimal control inputs $\mathbf{u}(t)$ which are passed to the robot's actuator. Since feedback control gain $\mathbf{K}(t)$ in (7.1) needs to be optimized in order to determine nominal optimal control $\mathbf{u}^o(t)$, the full-state feedback control problem boils down to the optimization problem (3.12).

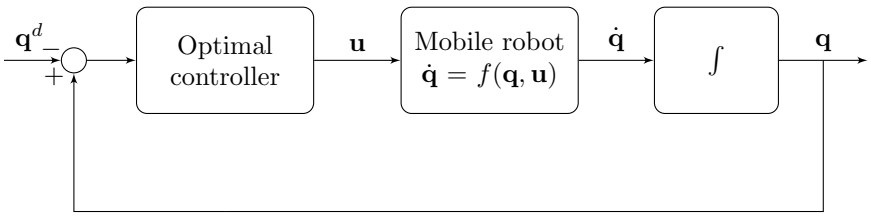


Figure 7.1: Full-state feedback structure.

For the robot to find the nominal optimal trajectory $\mathbf{q}^o(t)$, define the cost functional similar to (6.5) except that the measurement noise $\boldsymbol{\zeta}(t) = 0,\ \forall t \in I$, *i.e.*, $J(\mathbf{K}, \boldsymbol{\zeta}) \equiv J(\mathbf{K})$. The performance index $J(\mathbf{K})$ in (6.5) depends on the feedback control gain matrix $\mathbf{K}(t)$ through the state variable $\mathbf{q}(t)$ as it is clear from the feedback system (7.2). A compact form of (6.5) is

rewritten as

$$J(\mathbf{K}) = \phi[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \ell[t, \mathbf{q}(t)]dt. \tag{7.3}$$

The nominal solution can be stated as follows:

find $\mathbf{u}^o(t)$ and $\mathbf{q}^o(t),\ t \in I$,

minimizing $J(\mathbf{K})$,

subject to (3.7), (3.8), and (3.2).

Assume that the feedback gain $\mathcal{K} \subset \mathbb{R}^{2\times 3}$ is a closed bounded convex set and

$$\mathcal{K}_{ad} \equiv \left\{\mathbf{K}(t) \in \mathcal{L}_\infty^{\ell oc}([0,\infty), \mathbb{R}^{2\times 3}) : \mathbf{K}(t) \in \mathcal{K}\right\}$$

where $\mathcal{L}_p^{\ell oc}([0,\infty), \mathbb{R}^3)$ are locally convex topological function spaces of p-th power locally integrable functions containing the spaces $\mathcal{L}_p(I, \mathbb{R}^{2\times 3})$

Lemma 7.1 (Solutions of feedback system)**.** *Consider the feedback system* (7.2)*, which is defined over the finite time horizon of* I*, then for every initial condition* $\mathbf{q}(0) \in \mathbb{R}^2 \times \mathbb{S}^1$*, and feedback control gain* $\mathbf{K}(t) \in \mathcal{K}_{ad}$*, the system* (7.2) *has a unique absolutely continuous solution* $\mathbf{q}(t) \in C(I, \mathbb{R}^2 \times \mathbb{S}^1)$*. Further, the solution set* $\mathcal{Q} \equiv \{\mathbf{q}(t) \equiv \mathbf{q}[t, K(t)] \in C(I, \mathbb{R}^2 \times \mathbb{S}^1) : \mathbf{K}(t) \in \mathcal{K}_{ad}\}$ *is a bounded subset of* $C(I, \mathbb{R}^2 \times \mathbb{S}^1)$*.*

The proof of Lemma 7.1 is similar to the one presented in [Ahmed, 2006, Theorem 3.5.1, page 89]. Hence, the robot's actual trajectory in this case can also be given by

$$\mathbf{q}(t) = \mathbf{q}(t_0) + \frac{r}{2}\int_{t_0}^{t} \{\mathbf{B}[\mathbf{q}(\tau)]\mathbf{K}(\tau)\mathbf{q}(\tau)\}\, d\tau, \tag{7.4}$$

for $t \in I$.

To solve for the nominal optimal trajectory using the feedback system (7.2) that minimizes the objective functional (7.3), the necessary conditions of optimality need to be derived. For

that, define Hamiltonan

$$\mathcal{H} : I \times \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^3 \times \mathbb{R}^{2\times 3} \longmapsto \mathbb{R},$$

which is given by

$$\mathcal{H}[t, \mathbf{q}(t), \boldsymbol{\psi}(t), \mathbf{K}(t)] = \boldsymbol{\psi}^T \hat{\mathbf{f}}[\mathbf{q}(t), \mathbf{K}(t)] + \ell[t, \mathbf{q}(t)]. \tag{7.5}$$

Theorem 7.1 (Necessary Conditions of Optimality)**.** *Consider the feedback system* (7.2) *satisfying the Lemma 7.1. The optimal trajectory* $\mathbf{q^o}(t),\ t \in I$ *for the model* (7.2) *can be obtained if there exists an optimal feedback control gain* $\mathbf{K}^o(t) \in \mathcal{K}_{ad}$ *and an optimal multiplier* $\boldsymbol{\psi}^o(t) \in C(I, \mathbb{R}^3)$ *such that the triple* $\{\mathbf{q}^o, \boldsymbol{\psi}^o, \mathbf{K}^o\}$ *satisfies the following necessary conditions:*

$$\mathcal{H}[t, \mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}(t)] \geq \mathcal{H}[t, \mathbf{q^o}(t), \boldsymbol{\psi}^o(t), \mathbf{K^o}(t)],\ \mathbf{K}(t) \in \mathcal{K}, t \in I, \tag{7.6}$$

$$\dot{\mathbf{q}}^{\mathbf{o}} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\psi}}[t, \mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K^o}(t)],\ \mathbf{q^o}(t_0) = \mathbf{q_0},\ t \in I, \tag{7.7}$$

$$\dot{\boldsymbol{\psi}}^o = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}[t, \mathbf{q^o}(t), \boldsymbol{\psi}^o(t), \mathbf{K^o}(t),\ \boldsymbol{\psi}^o(t_f) = \frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)]. \tag{7.8}$$

Proof. Let $\mathbf{q}(t) \equiv \mathbf{q}[t, \mathbf{K}(t)]$ be the solution of the feedback system (7.2), with the cost functional (7.3) for any choice of $\mathbf{K}(t) \in \mathcal{K}_{ad}$. Since $\mathbf{K}^o(t)$ is optimal with the associated trajectory $\mathbf{q}^o(t)$, it is clear that

$$J(\mathbf{K}^o) \leq J(\mathbf{K}),\ \ \forall\, \mathbf{K}(t) \in \mathcal{K}_{ad},\ t \in I.$$

Again suppressing the variable t for clarity and for any $\varepsilon \in [0, 1]$, we define $\mathbf{K}^\varepsilon = \mathbf{K}^o + \varepsilon(\mathbf{K} - \mathbf{K^o})$. Since $\mathcal{K}$ is a closed convex set, $\mathcal{K}_{ad}$ is also a closed convex subset of $\mathcal{L}_\infty(I, \mathbb{R}^{2\times 3})$

and therefore $\mathbf{K}^{\varepsilon} \in \mathcal{K}_{ad}$. Thus $J(\mathbf{K^o}) \leq J(\mathbf{K}^{\varepsilon})$, which follows that

$$Tr\left[(\mathbf{K}-\mathbf{K}^{o})^{T}\, dJ(\mathbf{K}^{o})\right] \geq 0, \tag{7.9}$$

where $dJ(\mathbf{K}^{o})$ denotes the Gateaux (directional) derivative of J evaluated at $\mathbf{K} = \mathbf{K}^{o}$ in the direction of $(\mathbf{K} - \mathbf{K}^{o})$.

Let $\mathbf{q}^{\varepsilon}$ be the solution of the feedback system (7.2) corresponding to the gain $\mathbf{K}^{\varepsilon}$ with the same initial state $\mathbf{q}^{\varepsilon}(t_0) = \mathbf{q}_0$. It is easy to very that

$$\lim_{\varepsilon \to 0} \mathbf{K}^{\varepsilon}(t) \longrightarrow \mathbf{K}^{o}(t), \text{ and}$$
$$\lim_{\varepsilon \to 0} \mathbf{q}^{\varepsilon}(t) \longrightarrow \mathbf{q}^{o}(t).$$

Note that corresponding to the gains $\mathbf{K}^{\varepsilon}(t)$ and $\mathbf{K}^{o}(t)$, the state trajectories $\mathbf{q}^{\varepsilon}(t)$ and $\mathbf{q}^{o}(t)$ satisfy the following state equations:

$$\dot{\mathbf{q}}^{\varepsilon} = \hat{\mathbf{f}}(\mathbf{q}^{\varepsilon}, \mathbf{K}^{\varepsilon}), \;\; \mathbf{q}^{\varepsilon}(t_0) = \mathbf{q}_0, \; t \in I, \text{ and}$$
$$\dot{\mathbf{q}}^{\mathbf{o}} = \hat{\mathbf{f}}(\mathbf{q}^{o}, \mathbf{K}^{o}), \;\; \mathbf{q}^{o}(t_0) = \mathbf{q}_0, \; t \in I.$$

Subtracting one from the other we have the following equation

$$\dot{\mathbf{q}}^{\varepsilon} - \dot{\mathbf{q}}^{o} = \hat{\mathbf{f}}(\mathbf{q}^{\varepsilon}, \mathbf{K}^{o}) - \hat{\mathbf{f}}(\mathbf{q}^{o}, \mathbf{K}^{o}) + \varepsilon \hat{\mathbf{f}}(\mathbf{q}^{\varepsilon}, \mathbf{K} - \mathbf{K}^{o}). \tag{7.10}$$

Dividing by ε and denoting

$$\boldsymbol{\eta}(t) \equiv \lim_{\varepsilon \to 0} \left(\frac{\mathbf{q}^{\varepsilon}(t) - \mathbf{q}^{o}(t)}{\varepsilon} \right)$$

it follows from the expression (7.10) that $\eta(t)$ must satisfy the following initial value problem

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{q}}(\mathbf{q}^o, \mathbf{K}^o)\boldsymbol{\eta} + \hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K} - \mathbf{K}^o) \\ \boldsymbol{\eta}(t_0) &= 0. \end{aligned} \tag{7.11}$$

Eq. (7.11) is a linear non-homogeneous equation with $\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K} - \mathbf{K}^o)$ being the driving force. As a result, it has a continuous solution $\boldsymbol{\eta}(t) \in C(I, \mathbb{R}^3)$, which is continuously dependent on $\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K} - \mathbf{K}^o)$.

By definition of Gateaux (directional) derivative we can derive the following expression

$$\begin{aligned} &Tr\left[(\mathbf{K} - \mathbf{K}^o)^T dJ(\mathbf{K}^o)\right] = \lim_{\varepsilon \to 0} \frac{J(\mathbf{K}^o + \varepsilon(\mathbf{K} - \mathbf{K}^o)) - J(\mathbf{K}^o)}{\varepsilon} \\ &= \boldsymbol{\eta}^T(t)\frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t)\frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}(t)]dt. \end{aligned}$$

Hence, the inequality (7.9) yields

$$\boldsymbol{\eta}^T(t)\frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t)\frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}(t)]dt \geq 0. \tag{7.12}$$

Since $\boldsymbol{\eta}(t)$ of the variational equation (7.11) is continuously dependent on $\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K} - \mathbf{K}^o)$, it follows that the map

$$\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K} - \mathbf{K}^o) \longmapsto \boldsymbol{\eta}(t), \quad t \in I$$

is continuous from $\mathcal{L}_1(I, \mathbb{R}^3)$ to $C(I, \mathbb{R}^3)$. Hence, the map

$$\boldsymbol{\eta}(t) \longmapsto \boldsymbol{\eta}^T(t)\frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t)\frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}(t)]dt$$

is a continuous linear functional on $C(I, \mathbb{R}^3)$. Thus the composition map

$$\hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K} - \mathbf{K}^o) \longmapsto \boldsymbol{\eta}^T(t)\frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t)\frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}(t)]dt$$

is a continuous linear functional on $\mathcal{L}_1(I, \mathbb{R}^3)$. Therefore, by the Riesz representation theorem or by the duality between $\mathcal{L}_1(I, \mathbb{R}^3)$ and $\mathcal{L}_\infty(I, \mathbb{R}^3)$, we may conclude that there exists an element $\boldsymbol{\psi}^o \in \mathcal{L}_\infty(I, \mathbb{R}^3)$ such that

$$\begin{aligned} Tr\left[(\mathbf{K} - \mathbf{K}^o)^T dJ(\mathbf{K}^o)\right] &= \boldsymbol{\eta}^T(t)\frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f) + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t)\ell_{\mathbf{q}}(t, \mathbf{q}(t))dt \\ &= \int_{t_0}^{t_f} (\boldsymbol{\psi}^o)^T \hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K} - \mathbf{K}^o)dt. \end{aligned} \tag{7.13}$$

It follows from inequality (7.12) that

$$\int_{t_0}^{t_f} (\boldsymbol{\psi}^o)^T \hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K} - \mathbf{K}^o)dt \geq 0, \quad \forall\, \mathbf{K} \in \mathcal{K}_{ad}. \tag{7.14}$$

Using the variational equation (7.11), it follows from the second identity of (7.13) that

$$\boldsymbol{\eta}^T(t)\frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t)\frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}(t)]dt = \int_{t_0}^{t_f} \left\{ (\boldsymbol{\psi}^o)^T \left[\dot{\boldsymbol{\eta}} - \frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{q}}(\mathbf{q}^o, \mathbf{K}^o)\boldsymbol{\eta}(t) \right] \right\} dt. \tag{7.15}$$

Integrating by parts and recalling that $\boldsymbol{\eta}(t_0) = 0$, it follows from (7.15) that

$$\begin{aligned} &\int_{t_0}^{t_f} (\boldsymbol{\psi}^o)^T \left[\dot{\boldsymbol{\eta}} - \frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{q}}(\mathbf{q}^o, \mathbf{K}^o)\boldsymbol{\eta}(t) \right] dt \\ &\qquad = \boldsymbol{\eta}^T(t_f)\boldsymbol{\psi}^o(t_f) + \int_{t_0}^{t_f} \boldsymbol{\eta}^T(t) \left(-\dot{\boldsymbol{\psi}} - \left[\frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{q}}(\mathbf{q}^o, \mathbf{K}^o) \right]^T \boldsymbol{\psi}^o \right) dt. \end{aligned}$$

Expression (7.15) can now be written as

$$\begin{aligned}&\boldsymbol{\eta}^T(t)\frac{\partial\phi}{\partial\mathbf{q}}[t_f,\mathbf{q}(t_f)]+\int_{t_0}^{t_f}\boldsymbol{\eta}^T(t)\frac{\partial\ell}{\partial\mathbf{q}}[t,\mathbf{q}(t)]dt\\&=\boldsymbol{\eta}^T(t_f)\boldsymbol{\psi}^o(t_f)+\int_{t_0}^{t_f}\boldsymbol{\eta}^T\left\{-\dot{\boldsymbol{\psi}}^o-\left[\frac{\partial\hat{\mathbf{f}}}{\mathbf{q}}(\mathbf{q}^o,\mathbf{K}^o)\right]^T\boldsymbol{\psi}^o\right\}dt\end{aligned}\tag{7.16}$$

It is clear from (7.16) that

$$\begin{aligned}&\dot{\boldsymbol{\psi}}^o=-\left[\frac{\partial\hat{\mathbf{f}}}{\mathbf{q}}(\mathbf{q}^o,\mathbf{K}^o)\right]^T\boldsymbol{\psi}^o-\frac{\partial\ell}{\partial\mathbf{q}}[t,\mathbf{q}^o(t)]\ \text{ and}\\&\boldsymbol{\psi}^o(t_f)=\frac{\partial\phi}{\partial\mathbf{q}}[t_f,\mathbf{q}(t_f)].\end{aligned}\tag{7.17}$$

The expression (7.17) is called the costate dynamics and is linear along the optimal trajectories. Thus, the necessary conditions of optimality is given by the integral inequality (7.14), the costate dynamics (7.17), and the state equation (7.2). In other words, the choice of $\mathbf{K}\in\mathcal{K}_{ad}$ determines the optimality conditions (7.14), (7.17), and (7.2).

Consider the optimality condition (7.14) and rewriting it as follows

$$\int_{t_0}^{t_f}(\boldsymbol{\psi}^o)^T\,\hat{\mathbf{f}}(\mathbf{q}^o,\mathbf{K})dt\geq\int_{t_0}^{t_f}(\boldsymbol{\psi}^o)^T\,\hat{\mathbf{f}}(\mathbf{q}^o,\mathbf{K}^o)dt,\ \ \forall\,\mathbf{K}\in\mathcal{K}_{ad}.\tag{7.18}$$

Using the integral inequality (7.18), it is easy to derive the point-wise inequality given by

$$(\boldsymbol{\psi}^o)^T\,\hat{\mathbf{f}}(\mathbf{q}^o,\mathbf{K})\geq(\boldsymbol{\psi}^o)^T\,\hat{\mathbf{f}}(\mathbf{q}^o,\mathbf{K}^o),\ \ \forall\,\mathbf{K}\in\mathcal{K}_{ad}.\tag{7.19}$$

Now adding the term $\ell[t,\mathbf{q}^o(t)]$ in both sides of (7.19) yields

$$(\boldsymbol{\psi}^o)^T \hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}) + \ell[t, \mathbf{q}^o(t) \geq (\boldsymbol{\psi}^o)^T \hat{\mathbf{f}}(\mathbf{q}^o, \mathbf{K}^o) + \ell[t, \mathbf{q}^o(t),$$

which gives the Hamiltonian inequality

$$\mathcal{H}[t, \mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}(t)] \geq \mathcal{H}[t, \mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)].$$

This is exactly the inequality (7.6) stated in the theorem. Differentiating $\mathcal{H}^o$ with respect to the costate variable $\boldsymbol{\psi}$, we get

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{\psi}}[t, \mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)] = \hat{\mathbf{f}}[\mathbf{q}^o(t), \mathbf{K}^o(t)],$$

which gives the state equation

$$\dot{\mathbf{q}}^o = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\psi}}^o [t, \mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)], \quad \mathbf{q}^o(t_0) = \mathbf{q}_0,$$

as given in (7.7).

Differentiating $\mathcal{H}$ with respect to the state variable $\mathbf{q}$ yields

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}}[t, \mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)] = \left[\frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{q}}(q^o, K^o)\right]^T \boldsymbol{\psi}^o + \frac{\partial \ell}{\partial \mathbf{q}}[t, \mathbf{q}^o(t)],$$

Hence, the costate dynamics (7.17) can be expressed in terms of Hamiltonian as

$$\dot{\boldsymbol{\psi}}^o = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}^o [t, \mathbf{q}^o(t), \boldsymbol{\psi}^o(t), \mathbf{K}^o(t)], \quad \text{and}$$
$$\boldsymbol{\psi}^o(t_f) = \frac{\partial \phi}{\partial \mathbf{q}}[t_f, \mathbf{q}(t_f)],$$

which is precisely the expression (7.8). This completes the proof of all the necessary conditions given in (7.6) - (7.8).

□

In Theorem 6.1, it is assumed that there exists a feedback control gain $\mathbf{K}^o \in \mathcal{K}_{ad}$. The following Theorem gives the proof of this assumption.

Theorem 7.2 (Existence of optimal feedback gain $\mathbf{K}^o(t)$)**.** *Given the feedback system* (7.2)*, there exists an optimal feedback operator* $\mathbf{K}^o(t) \in \mathcal{K}_{ad}$ *that solves the problem* (3.12)*.*

Theorem 7.2 states that there exists a feedback control gain $\mathbf{K}^o(t) \in \mathcal{K}_{ad}$ for the robot to determine nominal optimal control inputs for its actuator and its proof is given in Appendix A.3. In order to solve for $\mathbf{K}^o(t)$, we express the gradient of the Hamiltonian defined in (7.5) as

$$\frac{\partial \mathcal{H}}{\partial \mathbf{K}} = \frac{r}{2}\mathbf{B}^T[\mathbf{q}(t)]\boldsymbol{\psi}(t)\mathbf{q}^T(t). \tag{7.20}$$

The gradient $\mathcal{H}_{\mathbf{K}} \equiv \frac{\partial \mathcal{H}}{\partial \mathbf{K}}$ indicates the direction of convergence for the optimal trajectory of (3.7). Having defined $\mathbf{q}(t)$ in (7.4) and $\boldsymbol{\psi}(t)$ in (7.8) for $t \in I$, the optimal $\mathbf{K}^o(t)$ can be found by setting the gradient (7.20) to be zero, *i.e.*,

$$\frac{r}{2}\mathbf{B}^T[\mathbf{q}(t)]\boldsymbol{\psi}(t)\mathbf{q}^T(t) = 0. \tag{7.21}$$

Note that the expression in (7.21) is independent of the gain matrix $\mathbf{K}(t)$. Hence, the problem boils down to finding $\mathbf{K}(t),\ t \in I$, such that the robot's actual trajectory (7.4) and the costate trajectory from (7.8) satisfy (7.21). The optimal feedback control gain $\mathbf{K}^o(t)$ can be determined by satisfying the Hamiltonian inequality (7.6). In other words, $\mathbf{K}(t)$ is to be adaptively tuned to minimize the robot's tracking error.

The following Corollary is similar to the one presented in section 6.4 of the previous chapter

(Corollary 6.1) except that the robot's full state feedback model (7.2) is considered and is repeated here for convenience.

Corollary 7.1 (Adapting the gain $\mathbf{K}$)**.** *Consider the robot's feedback system* (7.2) *defined over the time horizon* I*. Adapting the gain* $\mathbf{K}$ *according to the following offline update rule*

$$\mathbf{K}^{new} = \mathbf{K}^{old} - \epsilon \mathcal{H}_{\mathbf{K}}, \text{ for } 0 < \epsilon < 1 \tag{7.22}$$

satisfies the Hamiltonian inequality (7.6) *and, hence, guarantees the converge of the robot's trajectory towards its target.*

The proof of this Corollary is very similar to the one presented for Corollary 6.1 of Chapter 6. The key steps for numerical solution of the gain $\mathbf{K}(t),\ t \in I$, aggregating the components described earlier is presented below.

Key Steps for Nominal Full State Feedback Control Solution:

Let $\mathbf{K}_i \equiv \mathbf{K}_i(t)$, $t \in I$, be the gain at the i-th iteration of the optimization procedure.

Step 0 (initialization): Subdivide the time interval $I \equiv [t_0, t_f]$ into N subintervals. Assume a piecewise-contant $\mathbf{K_i}(t) = \mathbf{K_i}(t_k)$, $t \in [t_k, t_{k+1}]$, for $k = 0, \ldots, k_f - 1$.

Find the optimal gain $\mathbf{K}^o$ by repeating Steps 1–5 until the stopping criterion in Step 5 is met.

Step 1: Integrate the robot's feedback system (7.2) as in (7.4) with $\mathbf{K} \equiv \mathbf{K_i}(t)$, $t \in I$.

Step 2: Solve costate equation (7.8) backward for $\boldsymbol{\psi}_i$.

Step 3: Define the Hamiltonian $\mathcal{H}(t, \mathbf{q}_i, \boldsymbol{\psi}_i, \mathbf{K}_i)$ as in (7.5).

Step 4: Compute the cost function $J(\mathbf{K}_i)$ using (7.3), the gradients of the Hamiltonian $\mathcal{H}_{\mathbf{K}}$ using (7.20), and its corresponding intergrated norm $\int_{t_0}^{t_f} \|\mathcal{H}_{\mathbf{K}}\|^2 dt$.

Step 5: If $J(\mathbf{K}_i) \leq \delta_1$ or $\int_{t_0}^{t_f} \|\mathcal{H}_{\mathbf{K}}\|^2 dt \leq \delta_2$, for pre-defined small positive tolerance constants δ_1 and δ_2, then $\mathbf{K_i}$ is regarded close enough to its nominal optimal value, and so the algorithm is halted.

Otherwise, use the following update rule to adjust the piecewise-constant feedback control gain:

$$\mathbf{K}_{i+1}(t_k) = \mathbf{K}_i(t_k) - \epsilon\, \mathcal{H}_{\mathbf{K}(t_k)} + \lambda\, \Delta\mathbf{K}_i(t_k)$$
$$\Delta\mathbf{K}_i(t_k) = \mathbf{K}_i(t_k) - \mathbf{K}_{i-1}(t_k)$$

for $k = 0, \ldots, k_f - 1$, where ϵ and λ are the step size and the momentum constant (for faster convergence), respectively.

We now have the optimal feedback gain, $\mathbf{K}^o$. Using the robot's initial pose $\mathbf{q}^o(t) = \mathbf{q}(t_0) = \mathbf{q}_0$, it's nominal-optimal control can thus be computed by

$$\mathbf{u}^o(t) = \mathbf{K}^o(t)\mathbf{q}^o(t), \tag{7.23}$$

with the corresponding nominal-optimal state model

$$\dot{\mathbf{q}}^o(t) = \mathbf{f}[\mathbf{q}^o(t), \mathbf{u}^o(t)]. \tag{7.24}$$

The models (7.23) and (7.24) will be employed to solve for the robot's actual optimal control inputs and its corresponding state trajectory in real-time. This is illustrated in the next section.

7.3 Robot Optimal Trajectory

The previous section described the deterministic solution for the nominal optimal trajectories and the control inputs of a differential drive mobile robot where no external disturbances in control inputs were considered. As noted in the chapter introduction that when a dynamic system is subject to disturbances or parameter variations that cannot be specified ahead of time, then the deterministic solution discussed in section 7.2 is not applicable to minimize a certain objective function as in (7.3) [Stengel, 1994]. Hence, RSS measurements of RFID tags are employed in real-time as feedback information to describe the actual state of the robot. These uncertain RSS measurements are necessary for estimating the robot's state. The estimated state is then used to find the neighboring optimal control input. As such, the neighbouring optimal control principles and state estimation principles can be used together to solve the problem (3.12). The neighboring optimal control and the optimal pose estimation techniques for a differential drive mobile robot are illustrated in the following sections.

7.3.1 Neighbouring Optimal Control

For the robot to operate in real-time, it is conceivable that the exact optimal control could be updated continuously to provide the instantaneous control input to the robot's actuators. A practical method of doing so is to partition the robot's actual trajectory and control into: a)

nominal and b) perturbational parts, where the former represents the off-line deterministic solution (nominal) which is illustrated in section 7.2 and the later represents the on-line (real-time) solution [Stengel, 1994]. Let us denote the nominal trajectory and control as $(\mathbf{q}^o(t), \mathbf{u}^o(t))$, for $t \in I$, and rewrite the cost function (6.5) as

$$\begin{aligned} J(\mathbf{q}) = & \frac{1}{2}[\mathbf{q}(t_f) - \mathbf{q}^o(t_f)]^T \mathbf{P}(t_f)[\mathbf{q}(t_f) - \mathbf{q}^o(t_f)] \\ & + \frac{1}{2}\int_{t_0}^{t_f} \left\{[\mathbf{q}(t) - \mathbf{q}^o(t)]^T \mathbf{Q}(t)[\mathbf{q}(t) - \mathbf{q}^o(t)]\right\} dt \\ & + \frac{1}{2}\int_{t_0}^{t_f} \left\{[\mathbf{u}(t) - \mathbf{u}^o(t)]^T \hat{\mathbf{R}}(t)[\mathbf{u}(t) - \mathbf{u}^o(t)]\right\} dt \,, \end{aligned} \tag{7.25}$$

where $\hat{\mathbf{R}}(t) \in \mathbb{R}^{2\times 2}$ is a symmetric positive definite matrix. The compact form of the cost function (7.25) is

$$J(\mathbf{q}) = \phi[t_f, \mathbf{q}(t_f)] + \int_{t_0}^{t_f} \ell[t, \mathbf{q}(t), \mathbf{u}(t)]dt, \tag{7.26}$$

where

$$\begin{aligned} \phi[t_f, \mathbf{q}(t_f)] &= \frac{1}{2}[\mathbf{q}(t_f) - \mathbf{q}^o(t_f)]^T \mathbf{P}(t_f)[\mathbf{q}(t_f) - \mathbf{q}^o(t_f)] \\ \ell[t, \mathbf{q}(t), \mathbf{u}(t)] &= \frac{1}{2}\left\{[\mathbf{q}(t) - \mathbf{q}^o(t)]^T \mathbf{Q}(t)[\mathbf{q}(t) - \mathbf{q}^o(t)] + [\mathbf{u}(t) - \mathbf{u}^o(t)]^T \hat{\mathbf{R}}(t)[\mathbf{u}(t) - \mathbf{u}^o(t)]\right\} \end{aligned}$$

Defining the perturbations from the nominal optimal solutions as

$$\begin{aligned} \Delta\mathbf{q}(t) &= \mathbf{q}(t) - \mathbf{q}^o(t) \\ \Delta\mathbf{u}(t) &= \mathbf{u}(t) - \mathbf{u}^o(t) \end{aligned} \qquad t \in I, \tag{7.27}$$

the robot's model (3.7) can be expanded as the Taylor series

$$\dot{\mathbf{q}}^o(t) + \Delta\dot{\mathbf{q}}(t) = \mathbf{f}\left[\mathbf{q}^o(t), \mathbf{u}^o(t)\right] + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}\left[\mathbf{q}^o(t), \mathbf{u}^o(t)\right]\Delta\mathbf{q}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\left[\mathbf{q}^o(t), \mathbf{u}^o(t)\right]\Delta\mathbf{u}(t) + \mathcal{O}[\Delta\mathbf{q}, \Delta\mathbf{u}],$$

where $\mathcal{O}[\Delta\mathbf{q}, \Delta\mathbf{u}]$ is the higher order terms of $\Delta\mathbf{q}(t)$ and $\Delta\mathbf{u}(t)$. Using the model (7.24) and assuming the perturbation variables to be "relatively small", the above expression can be truncated to first degree, yielding the robot's linear kinematic constraint

$$\Delta\dot{\mathbf{q}}(t) = \mathbf{F}(t)\Delta\mathbf{q}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t), \quad \Delta\mathbf{q}(t_0) = \Delta\mathbf{q}_0, \tag{7.28}$$

where

$$\mathbf{F}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}\left[\mathbf{q}^o(t), \mathbf{u}^o(t)\right] \qquad \text{and}$$
$$\mathbf{G}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\left[\mathbf{q}^o(t), \mathbf{u}^o(t)\right].$$

The cost function (7.26) can be expanded as

$$J[\mathbf{q}^o + \Delta\mathbf{q}] \cong J[\mathbf{q}^o] + \Delta J[\Delta\mathbf{q}] + \Delta^2 J[\Delta\mathbf{q}].$$

However, the optimality guarantees that the first variation of $J[\cdot]$ (*i.e.*, $\Delta J[\Delta\mathbf{q}(t)]$) is zero [Stengel, 1994], which yields

$$J[\mathbf{q}^o + \Delta\mathbf{q}] \cong J[\mathbf{q}^o] + \Delta^2 J[\Delta\mathbf{q}],$$

where the second variation of $J[\cdot]$ can be expressed as

$$\Delta^2 J[\Delta\mathbf{q}] = \frac{1}{2}\Delta\mathbf{q}^T(t_f)\phi_{qq}(t_f)\Delta\mathbf{q}(t_f)+ \frac{1}{2}\int_{t_0}^{t_f}\left\{\begin{bmatrix}\Delta\mathbf{q}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}\ell_{\mathbf{qq}} & \ell_{\mathbf{qu}}\\ \ell_{\mathbf{uq}} & \ell_{\mathbf{uu}}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{q}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}\right\}dt, \tag{7.29}$$

subject to (7.28).

Let us rewrite the expression (7.29) as

$$\Delta^2 J[\Delta\mathbf{q}] \triangleq J = \frac{1}{2}\Delta\mathbf{q}^T(t_f)\mathbf{P}(t_f)\Delta\mathbf{q}(t_f) + \frac{1}{2} \int_{t_0}^{t_f}\left\{\begin{bmatrix}\Delta\mathbf{q}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}\mathbf{Q}(t) & \mathbf{M}(t)\\ \mathbf{M}^T(t) & \mathbf{R}(t)\end{bmatrix}\begin{bmatrix}\Delta\mathbf{q}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}\right\}dt, \tag{7.30}$$

where

$$\begin{aligned}\mathbf{P}(t_f) &\equiv \phi_{qq}(t_f) \equiv \frac{\partial^2\phi}{\partial\mathbf{q}^2}\left[t_f, \mathbf{q}^o(t_f)\right]\\ \mathbf{Q}(t) &\equiv \ell_{\mathbf{qq}} \equiv \frac{\partial^2\ell}{\partial\mathbf{q}^2}\left[t, \mathbf{q}^o(t), \mathbf{u}^o(t)\right],\\ \mathbf{M}(t) &\equiv \ell_{\mathbf{qu}} \equiv \frac{\partial^2\ell}{\partial\mathbf{q}\partial\mathbf{u}}\left[t, \mathbf{q}^o(t), \mathbf{u}^o(t)\right], \text{ and}\\ \mathbf{R}(t) &\equiv \ell_{\mathbf{uu}} \equiv \frac{\partial^2\ell}{\partial\mathbf{u}^2}\left[t, \mathbf{q}^o(t), \mathbf{u}^o(t)\right].\end{aligned}$$

Since $\mathbf{M}(t) = \mathbf{0}$, it follows from (7.30) that

$$J = \frac{1}{2}\Delta\mathbf{q}^T(t_f)\mathbf{P}(t_f)\Delta\mathbf{q}(t_f)+ \frac{1}{2}\int_{t_0}^{t_f}\left\{\begin{bmatrix}\Delta\mathbf{q}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}\mathbf{Q}(t) & \mathbf{0}\\ \mathbf{0} & \mathbf{R}(t)\end{bmatrix}\begin{bmatrix}\Delta\mathbf{q}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}\right\}dt. \tag{7.31}$$

Equation (7.31) defines the quadratic cost functional.

Theorem 7.3 (linear-quadratic control law)**.** *Consider the robot's linear kinematic model* (7.28) *and its quadratic cost functional given by* (7.31)*. The optimal linear-quadratic state feedback control law is given by*

$$\Delta\mathbf{u}^o(t) = -\mathbf{R}^{-1}(t)\mathbf{G}^T(t)\mathbf{P}(t)\Delta\mathbf{q}(t) = -\mathbf{C}(t)\Delta\mathbf{q}(t), \tag{7.32}$$

where $\mathbf{C}(t)$ *is the* (2×3) *neghboring-optimal control gain matrix and* $\mathbf{P}(t)$ *is the solution of the differential matrix Riccati equation*

$$\dot{\mathbf{P}} = -\mathbf{F}^T(t)\mathbf{P}(t) - \mathbf{Q}(t) - \mathbf{P}(t)\mathbf{F}(t) + \mathbf{P}(t)\mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{G}^T(t)\mathbf{P}(t), \;\; \mathbf{P}(t_f) = \mathbf{P}_f. \tag{7.33}$$

The proof this Theorem is given in Appendix A.4. It is interesting to note that the solution for $\mathbf{P}(t)$ and, therefore, for $\mathbf{C}(t)$ is independent of $\Delta\mathbf{q}(t)$. Variations in $\Delta\mathbf{q}(t_0)$ or $\Delta\mathbf{q}(t_f)$ have no effect on $\mathbf{C}(t)$, although the linear-optimal control history obviously is affected by state perturbations [Stengel, 1994].

It is clear from Theorem 7.3 that once the solution of the differential matrix Riccati equation (7.33) is available, the feedback control law given by (7.32) can be formally constructed. From the perturbation (7.27), the total control is formed as the sum of the nominal and the perturbation optimal controls as stated in the chapter introduction:

$$\begin{aligned} \mathbf{u}(t) &= \mathbf{u}^o(t) + \Delta\mathbf{u}^o(t) \\ &= \mathbf{u}^o(t) - \mathbf{C}(t)[\hat{\mathbf{q}}(t) - \mathbf{q}^o(t)], \end{aligned} \tag{7.34}$$

where $\hat{\mathbf{q}}(t)$ is the robot's estimated pose which will be determined in section 7.3.2.

Substituting perturbed optimal control (7.32) in (7.28) yields the perturbed stated feedback

system

$$\begin{aligned}\Delta\dot{\mathbf{q}}(t) &= \left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{G}^T(t)\mathbf{P}(t)\right]\Delta\mathbf{q}(t), \\ &\equiv \mathbf{A}(t)\Delta\mathbf{q}(t),\ \Delta\mathbf{q}(t_0) = \Delta\mathbf{q}_0 \neq \mathbf{0},\end{aligned} \tag{7.35}$$

with $\mathbf{A}(t) \equiv \left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{G}^T(t)\mathbf{P}(t)\right]$ and the corresponding state trajectory can then be described by

$$\Delta\mathbf{q}(t) = \mathbf{\Phi}(t, t_0)\Delta\mathbf{q}(t_0),$$

where $\mathbf{\Phi}(t, t_0) = e^{tA(t)}$ is the state transition matrix.

The feedback model (7.35) with the quadratic cost functional (7.30) is similar to the optimal linear quadratic regulator problem, which is stable in the Laypunov sense [Ahmed, 2006]. In other words, the optimality condition guarantees the controller's stability. Before illustrating the controller's stability condition, let us present the following Lemmas.

Lemma 7.2 (Lyapunov properties [Khalil, 2002])**.** *If* $V : [0, \infty] \times \mathbb{R}^3 \rightarrow \mathbb{R}$ *is a Lyapunov-candidate-function and is given by*

$$V[t, \Delta\mathbf{q}(t)] = \frac{1}{2}\Delta\mathbf{q}^T(t)\mathbf{P}(t)\Delta\mathbf{q}(t) \tag{7.36}$$

and $\mathbf{P}(t)$*, which is the solution of* (7.33)*, is a real, symmetric, positive semi-definite matrix; then* $V[\cdot]$ *satisfies the Lyapunov basic properties:*

- $V[t, \mathbf{0}] = 0$
- $V[\cdot]$ *is at least once continuously differentiable with respect to* $\Delta\mathbf{q}$*, i.e.,* $\frac{\partial V}{\partial \Delta\mathbf{q}}[t, \Delta\mathbf{q}(t)] \in C^1(I, \mathbb{R}^3)$
- $V[t, \Delta\mathbf{q}(t)] \geq 0$ *for* $t \in I$.

The proof of this Lemma is given in Appendix A.5.

Lemma 7.3 (Controllability)**.** *Consider the matrix* $\mathbf{Q}(t)$ *in* (7.30) *is defined such that* $\mathbf{Q}(t) = \mathbf{D}^T(t)\mathbf{D}(t)$*, where the matrix* $\mathbf{D}(t)$ *is nonsingular. The feedback system* (7.35) *is then locally controllable in the time horizon* I *if the* (3×3) *controllability Grammian matrix* $\mathbb{M}(t_f, t_0)$*, defined as*

$$\mathbb{M}(t_f, t_0) = \int_{t_0}^{t_f} \left[\mathbf{\Phi}^T(t, t_0)\mathbf{D}^T(t)\mathbf{D}(t)\mathbf{\Phi}(t, t_0)\right] dt \tag{7.37}$$

is nonsingular, i.e., $\mathbb{M}(t_f, t_0) > \mathbf{0}$*.*

See Appendix A.6 for its proof.

Theorem 7.4 (Optimality to stability)**.** *The feedback system given by* (7.35) *is*

(i) stable in the Lyapunov sense with respect to the Lyapunov function given by (7.36) *if* $\mathbf{Q}(t)$ *is a real, symmetric, positive semi-definite matrix*

(ii) asymptotically stable if one of the following conditions holds:

- $\mathbf{Q}(t)$ *is a real, symmetric, positive definite matrix; and*
- $\mathbf{Q}(t) = \mathbf{D}^T(t)\mathbf{D}(t)$ *and* $\{\mathbf{A}(t), \mathbf{D}(t)\}$ *is controllable.*

Proof. By taking the time-derivative of the Lyapunov function (7.36) and using the expression for $\Delta\dot{\phi}$ in (A.19), it follows that

$$\dot{V} = \frac{1}{2}\Delta\dot{\phi} = \frac{1}{2}\left\{-\Delta\mathbf{q}^T\mathbf{Q}\Delta\mathbf{q} - \Delta\mathbf{u}^T\mathbf{R}\Delta\mathbf{u}\right\} \leq 0. \tag{7.38}$$

(i) It is certain from (7.38) that if $\mathbf{Q}(t)$ is a real, symmetric, positive semi-definite matrix, then $\dot{V}[t, \Delta\mathbf{q}(t)] = (1/2)\Delta\dot{\phi} \leq 0$. Hence, the feedback system (7.35) is stable in the Lyapunov sense with respect to the Lyapunov function (7.36).

(ii) Since $\mathbf{R}$ is positive definite and if $\mathbf{Q}(t)$ is also positive definite, the expression (7.38) yields $\dot{V} < 0$. Therefore, the feedback system (7.35) is asymptotically stable.

Since $\mathbf{Q}(t) = \mathbf{D}^T(t)\mathbf{D}(t)$ and $\Delta\mathbf{q}(t) \equiv \Delta\mathbf{q}(t, \Delta\mathbf{q}_0)$ is the solution of the feedback system (7.35) with $\Delta\mathbf{q}_0 \neq \mathbf{0}$, the expression (7.38) can rewritten as

$$\dot{V} = -\frac{1}{2}\left\{\|\mathbf{D}(t)\Delta\mathbf{q}(t, \Delta\mathbf{q}_0)\|^2 + \Delta\mathbf{u}^T\mathbf{R}\Delta\mathbf{u}\right\}.$$

For the system to be asymptotically stable, it is required that for any $t_f < \infty$,

$$\int_{t_0}^{t_f} \|\mathbf{D}(t)\Delta\mathbf{q}(t, \Delta\mathbf{q}_0)\|^2 dt = \int_{t_0}^{t_f} \Delta\mathbf{q}^T(t, \Delta\mathbf{q}_0)\mathbf{D}^T(t)\mathbf{D}(t)\Delta\mathbf{q}(t, \Delta\mathbf{q}_0)dt > 0.$$

Using Lemma 7.3 it is clear that the above expression is valid if $\{\mathbf{A}(t), \mathbf{D}(t)\}$ is controllable and hence,

$$\mathbb{M}(t_f, t_0) = \int_{t_0}^{t_f} \left[\mathbf{\Phi}^T(t, t_0)\mathbf{D}^T(t)\mathbf{D}(t)\mathbf{\Phi}(t, t_0)\right] dt > \mathbf{0}.$$

Thus, $\dot{V} < 0$ which concludes the asymptotic stability of the feedback system (7.35).

□

7.3.2 Optimal Pose Estimation

The DDMR employed in this thesis is subject to external disturbance (process noise) and is driven by the control law given in (7.34). Note that the robot's control input (speeds) $\mathbf{u}(t)$ in (7.34) requires the state feedback which is subjected to noise, hence, dynamic measurements must be taken into account for the robot to estimate its current pose $\hat{\mathbf{q}}(t)$. These measurements are also subjected to noise of the robot's operating environment. As can be seen form Figure 6.1, the robot receives RSS measurements from RFID tags deployed in its environment. In an indoor reverberant environment, these RSS measurements can be highly contaminated by ambient noise.

Hence, estimating the robot's pose in such environment is a challenging task. In the following, an optimal filter is presented to "filter out" the noise embedded in the RSS measurements for estimating the robot's pose.

Since the robot itself is subject to the noisy speed (process noise), the model (3.7) can be rewritten as

$$\dot{\mathbf{q}}(t) = \mathbf{f}[\mathbf{q}(t), \mathbf{u}(t), \boldsymbol{\xi}(t)], \tag{7.39}$$

where $\boldsymbol{\xi}(t)$ is the noise associated with control input $\mathbf{u}(t)$. The Taylor series expansion of (7.39), neglecting the higher order terms, yields

$$\Delta\dot{\mathbf{q}}(t) = \mathbf{F}(t)\Delta\mathbf{q}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t) + \mathbf{L}(t)\Delta\boldsymbol{\xi}(t),\ \Delta\mathbf{q}(t_0) = \Delta\mathbf{q}_0,$$

where

$$\mathbf{L}(t) = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}}\left[\mathbf{q}^o(t), \mathbf{u}^o(t), \boldsymbol{\xi}^o(t)\right], \text{ and } \Delta\boldsymbol{\xi}(t) = \boldsymbol{\xi}(t) - \boldsymbol{\xi}^o(t).$$

Note that $\boldsymbol{\xi}^o(t) = \mathbf{0}$ because the deterministic solution of (7.24) has no process noise. The expected values of the initial state and covariance are

$$\begin{aligned} \mathbb{E}[\mathbf{q}(t_0)] &= \hat{\mathbf{q}}_0, \\ \mathbb{E}\left\{[\mathbf{q}(t_0) - \hat{\mathbf{q}}_0][\mathbf{q}(t_0) - \hat{\mathbf{q}}_0]^T\right\} &= \mathbf{S}_0. \end{aligned} \tag{7.40}$$

For simplicity, assume that the robot's input and measurement noise are a white, zero-mean Gaussian random process. If $\mathbf{W}_C$ and $\mathbf{N}_C$ are spectral density matrices of the robot's input and measurement noise, respectively, the following expression holds:

$$\mathbb{E}\begin{bmatrix}\boldsymbol{\xi}^T(t) & \boldsymbol{\zeta}^T(t)\end{bmatrix} = \begin{bmatrix}\bar{\boldsymbol{\xi}}^T & \bar{\boldsymbol{\zeta}}^T\end{bmatrix}$$
$$\mathbb{E}\left\{\begin{bmatrix}\boldsymbol{\xi}(\tau)\\ \boldsymbol{\zeta}(\tau)\end{bmatrix}\begin{bmatrix}\boldsymbol{\xi}^T(t) & \boldsymbol{\zeta}^T(t)\end{bmatrix}\right\} = \begin{bmatrix}\mathbf{W}_C(t & \mathbf{0}\\ \mathbf{0} & \mathbf{N}_C(t)\end{bmatrix}\delta(t-\tau), \tag{7.41}$$

where $\delta(\cdot)$ is the dirac delta function defined by

$$\delta(t-\tau) = \begin{cases}\infty, & \tau = t\\ 0, & \tau \neq t\end{cases}$$
$$\lim_{\kappa\to 0}\int_{t-\kappa}^{t+\kappa}\delta(t-\tau)d\tau = 1 \quad \text{(unit impulse function)}.$$

The robot's a priori state estimate is described by

$$\hat{\mathbf{q}}(t) = \hat{\mathbf{q}}_0 + \int_{t_0}^{t}\mathbf{f}[\hat{\mathbf{q}}(t), \mathbf{u}(t)]dt. \tag{7.42}$$

From the measurement model (6.1), the matrix $\mathbf{H}(t)$ is determined along the a priori estimate $\hat{\mathbf{q}}(t)$ found in (7.42) as

$$\mathbf{H}(t) = \frac{\partial \mathbf{h}}{\partial \mathbf{q}}[\hat{\mathbf{q}}(t)].$$

The optimal filter gain can then be computed as

$$\mathbb{K}_C = \mathbf{S}(t)\mathbf{H}^T(t)\mathbf{N}_C^{-1}(t), \tag{7.43}$$

where the state covariance matrix $\mathbf{S}(t)$ is the solution of the differential matrix Riccati equation

$$\begin{aligned}\dot{\mathbf{S}}(t) =&\mathbf{F}(t)\mathbf{S}(t) + \mathbf{S}(t)\mathbf{F}^T(t) + \mathbf{L}(t)\mathbf{W}_C(t)\mathbf{L}^T(t) \\ &- \mathbf{S}(t)\mathbf{H}^T(t)\mathbf{N}_C^{-1}(t)\mathbf{H}(t)\mathbf{S}(t), \;\; \mathbf{S}(t_0) = \mathbf{S}_0.\end{aligned} \tag{7.44}$$

Using the current RSS measurement, $\mathbf{z}(t)$ given in (6.1), the robot's a posteriori state estimate is determined by solving the following state model:

$$\dot{\hat{\mathbf{q}}}(t) = \mathbf{f}[\hat{\mathbf{q}}(t), \mathbf{u}(t)] + \mathbb{K}_C \left\{\mathbf{z}(t) - \mathbf{h}[\hat{\mathbf{q}}(t)]\right\}, \; \hat{\mathbf{q}}(t_0) = \hat{\mathbf{q}}_0. \tag{7.45}$$

Hence, finding the robot's optimal trajectory contains four parts which are given below:

Neighboring Optimal Control Algorithm:

1. computation of the robot's nominal-optimal control and trajectory as given in (7.23) and (7.24),

2. computation of neighboring-optimal control gain matrix:

   (a) specify the cost function as given in (7.31) subject to the robot's linear kinematic contraint (7.28).

   (b) define the Hamiltonian as in (A.8) for the neighboring-optimal trajectory and control.

   (c) solve the differential matrix Riccati equation (7.33) that results from minimizing the Hamiltonian to obtain the adjoint covariance matrix [Stengel, 1994], $\mathbf{P}(t)$, from t_f to t_0.

   (d) compute the neighboring-optimal gain matrix, $\mathbf{C}(t)$, as given in (A.17).

3. optimal estimation of the robot's pose:

   (a) initialize estimated pose and state error covariance matrix as in (7.40).

   (b) use (7.41) to compute error covariance matrices, $\mathbf{W}_C$ and $\mathbf{N}_C$.

   (c) integrate the differential matrix Riccati equation (7.44).

   (d) compute the filter gain, $\mathbb{K}_C$, as in (7.43).

   (e) optimal pose is estimated using (7.45).

4. actual optimal control and trajectory generation

   (a) compute the robot's actual optimal control by (7.34).

   (b) robot's actual trajectory is then computed from the solution of (7.45).

The above steps can be used in conjunction with simulation or real-time control to generate the robot's actual optimal trajectories corresponding to its cost function (7.31), kinematic contraint (3.7), and measurements (6.1). Figure 7.2 shows the schematic diagram of the key steps of the proposed trajectory tracking system for a DDMR. Having computed the filter gain $\mathbb{K}_C$ and neighboring optimal control gain matrix $\mathbf{C}(t)$, the linear stochastic controller is seen to be driven by the nonlinear RSS measurements, $\mathbf{z}(t)$. Its output is added to the nonminal-optimal control, $\mathbf{u}^o(t)$. The nominal-optimal state, $\mathbf{q}^o(t)$, is used to derive the total pose estimate $\hat{\mathbf{q}}(t)$.

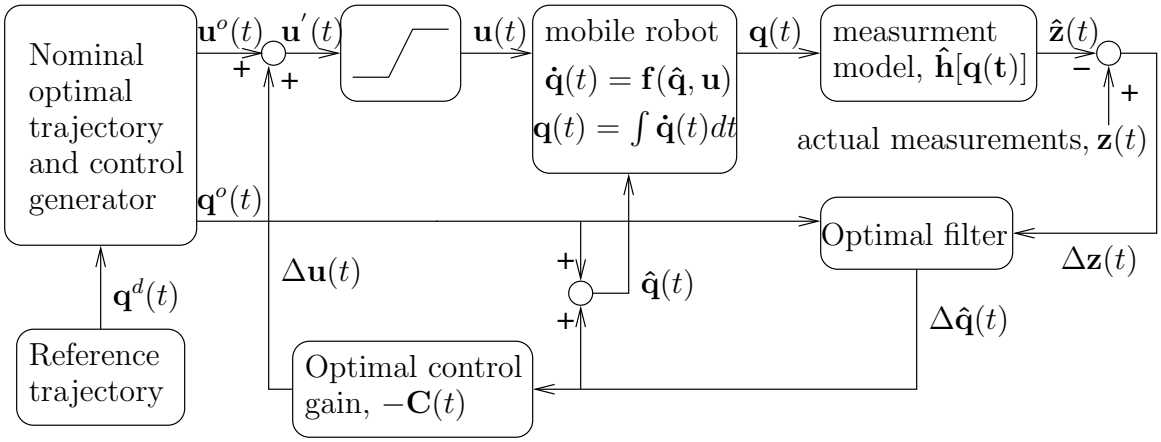


Figure 7.2: Schematic of the robot's stochastic neighboring-optimal control law in continuous time.

7.4 Simultaneous Localization and Mapping Method

The prime aim of this thesis is to navigate a mobile robot using motion control techniques which are modular and cost effective. For that, estimating the robot's pose using optimal filtering method is the essential part which is illustrated in section 7.3.2. The problem of Simultaneous Localization and Mapping (SLAM) has been an active research topic in recent years [Castellanos et al., 1999, Choi and Lee, 2010, Dissanayake et al., 2001, Montemerlo et al., 2002, Smith et al., 1990]. In order to compare the pose estimation technique proposed in this Chapter with an EKF SLAM algorithm, we dedicate this section to formally define a SLAM

algorithm. For this purpose, the SLAM technique suggested in [Choi and Lee, 2010] is adopted.

7.4.1 Formulating the SLAM Problem

The SLAM problem in the context of this thesis is to simultaneously estimate the position of the robot and the RFID tags placed in the robot's operating environment. It is important to stress out the fact that since the robot navigation methods illustrated in this thesis are using RFID systems, there is no data association problem while receiving the measurements from RFID tags. This is to say that the robot is perfectly able to distinguish the RSS measurements coming from the RFID tags.

For the robot to solve the SLAM problem, the discrete-time model (5.1) is assumed and the robot's pose (x, y, θ) is augmented with the tag positions. In order to simplify the derivation, and since the z-component of the RFID tag location is already known, the unknown positions of the tags are assumed to be 2-D and the robot's orientation θ is taken from the real line $\mathbb{R}$. Hence the new state vector is defined as

$$\bar{\mathbf{q}} = \begin{bmatrix} x & y & \theta & x^1 & y^1 & x^2 & y^2 & \dots & x^s & y^s \end{bmatrix}^T,$$

where s is the number of RFID tags, $(x^j, \ y^j)$ is the jth tag's unknown coordinates, respectively. Clearly $\bar{\mathbf{q}} \in \mathbb{R}^{2s+3}$. If $\bar{\mathbf{q}}^{\mathbf{d}}$ represents the desired robot's state and map of its environment, then the problem is to find

$$\|\bar{\mathbf{q}} - \bar{\mathbf{q}}^{\mathbf{d}}\| \leq \epsilon,$$

for smal positive real number ϵ.

7.4.2 Mathematical Derivation of the EKF SLAM

Since the RFID tags are fixed, the resulting dynamic model for the augmented system is

$$\bar{\mathbf{q}}_k = \bar{\mathbf{q}}_{k-1} + T\frac{r}{2}\begin{bmatrix} (u_{R,k-1} + u_{L,k-1})\cos\theta_{k-1} \\ (u_{R,k-1} + u_{L,k-1})\sin\theta_{k-1} \\ \frac{2}{l}((u_{R,k-1} - u_{L,k-1})) \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} (\xi_{R,k-1} + \xi_{L,k-1})\cos\theta_{k-1} \\ (\xi_{R,k-1} + \xi_{L,k-1})\sin\theta_{k-1} \\ \frac{1}{l}(\xi_{R,k-1} - \xi_{L,k-1}) \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where $\boldsymbol{\xi_k} \equiv [\xi_{R,k} \;\; \xi_{L,k}]^T$ is a zero-mean white Gaussian noise with covariance matrix

$$\mathbf{W}_k = \begin{bmatrix} \sigma_{R,k}^2 & 0 \\ 0 & \sigma_{L,k}^2 \end{bmatrix}$$

and $\sigma_{L,k}$ and $\sigma_{R,k}$ being the standard deviation errors of the left and right wheels, respectively. Recall that the kinematic model of the DDMR has the form

$$\mathbf{q}_k = \mathbf{f}(\mathbf{q}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\xi_k}) \tag{7.46}$$

where

$$\mathbf{f}(\mathbf{q}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\xi_k}) = \begin{bmatrix} x_{k-1} + T\frac{r}{2}((u_{R,k-1} + \xi_{R,k-1}) + (u_{L,k-1} + \xi_{L,k-1}))\cos\theta_{k-1} \\ y_{k-1} + T\frac{r}{2}((u_{R,k-1} + \xi_{R,k-1}) + (u_{L,k-1} + \xi_{L,k-1}))\sin\theta_{k-1} \\ \theta_{k-1} + T\frac{r}{2}\frac{2}{l}((u_{R,k-1} + \xi_{R,k-1}) - (u_{L,k-1} + \xi_{L,k-1})) \end{bmatrix}.$$

Using Taylor series expansion at $\boldsymbol{\mu}_{k-1} \approx \mathbf{q}_{k-1}$, model (7.46) can be linearized as

$$\mathbf{q}_k \approx \mathbf{g}(\boldsymbol{\mu}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\xi_k}) + \mathbf{F}_{k-1}(\mathbf{q}_{k-1} - \boldsymbol{\mu}_{k-1}),$$

where $\mathbf{F}_{k-1}$ is a Jacobian matrix of dimension 3×3 and is defined as

$$\begin{aligned}\mathbf{F}_{k-1} &= \frac{\partial}{\partial \mathbf{q}} \left[\mathbf{f}(\mathbf{q}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\xi_k})\right] \\ &= \begin{bmatrix} 1 & 0 & -T\frac{r}{2}((u_{R,k-1}+\xi_{R,k-1}) + (u_{L,k-1}+\xi_{L,k-1}))\sin\theta_{k-1} \\ 0 & 1 & T\frac{r}{2}((u_{R,k-1}+\xi_{R,k-1}) + (u_{L,k-1}+\xi_{L,k-1}))\cos\theta_{k-1} \\ 0 & 0 & 1 \end{bmatrix} \\ &= \mathbf{I}_3 + \begin{bmatrix} 0 & 0 & -T\frac{r}{2}((u_{R,k-1}+\xi_{R,k-1}) + (u_{L,k-1}+\xi_{L,k-1}))\sin\theta_{k-1} \\ 0 & 0 & T\frac{r}{2}((u_{R,k-1}+\xi_{R,k-1}) + (u_{L,k-1}+\xi_{L,k-1}))\cos\theta_{k-1} \\ 0 & 0 & 0 \end{bmatrix},\end{aligned}$$

where $\mathbf{I}_3$ is a 3×3 identity matrix. The Jacobian matrix with respect to the robot's model uncertainties is given by

$$\mathbf{L}_{k-1} = T\frac{r}{2}\begin{bmatrix} \cos\theta_{k-1} & \cos\theta_{k-1} \\ \cos\theta_{k-1} & \cos\theta_{k-1} \\ \frac{2}{l} & -\frac{2}{l}. \end{bmatrix}$$

Hence, the full motion model of the robot in cooperation with the tags is described as follows

$$\bar{\mathbf{q}}_k = \bar{\mathbf{q}}_{k-1} + T\frac{r}{2}\bar{\mathbf{F}}^T \begin{bmatrix} (u_{R,k-1}+u_{L,k-1})\cos\theta_{k-1} \\ (u_{R,k-1}+u_{L,k-1})\sin\theta_{k-1} \\ \frac{2}{l}(u_{R,k-1}-u_{L,k-1}) \end{bmatrix} + \mathcal{N}(0, \bar{\mathbf{F}}^T\mathbf{L}_{k-1}\mathbf{W}_{k-1}\mathbf{L}_{k-1}^T\bar{\mathbf{F}}),$$

where

$$\bar{\mathbf{F}} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}_{3\times 2s}$$

The predicted state and covariance equations for the EKF SLAM are given by

$$\bar{\mathbf{q}}_k = \bar{\mathbf{q}}_{k-1} + T\frac{r}{2}\bar{\mathbf{F}}^T \begin{bmatrix} (u_{R,k-1} + u_{L,k-1})\cos\theta_{k-1} \\ (u_{R,k-1} + u_{L,k-1})\sin\theta_{k-1} \\ \frac{2}{l}(u_{R,k-1} - u_{L,k-1}) \end{bmatrix}$$

$$\bar{\mathbf{S}}_k = \bar{\mathbf{G}}_{k-1}\bar{\mathbf{S}}_{k-1}\bar{\mathbf{G}}_{k-1} + \bar{\mathbf{F}}^T\mathbf{L}_{k-1}\mathbf{W}_{k-1}\mathbf{L}_{k-1}^T\bar{\mathbf{F}},$$

where $\bar{\mathbf{S}}$ is the augmented state covariance matrix and $\bar{\mathbf{G}}_k$ is defined as follows

$$\bar{\mathbf{G}}_k = \mathbf{I}_{2s+3} + \bar{\mathbf{F}}^T \begin{bmatrix} 0 & 0 & -T\frac{r}{2}(u_{R,k-1} + u_{L,k-1})\sin\theta_{k-1} \\ 0 & 0 & T\frac{r}{2}(u_{R,k-1} + u_{L,k-1})\cos\theta_{k-1} \\ 0 & 0 & 0 \end{bmatrix} \bar{\mathbf{F}}.$$

Let us consider that the robot is able to measure the line-of-sight distance between itself and the jth tag using the nonlinear model (6.2) and is equipped with a bearing sensor for measuring the bearing φ^j between the robot and the jth tag. Hence, the measurement model for the SLAM problem at time instant k becomes

$$z_k^j = \begin{bmatrix} r_k^j \\ \varphi^j \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k - x^j)^2 + (y_k - y^j)^2} \\ \text{ATAN2}\left[y_k - y^j, x_k - x^j\right] - \theta_k \end{bmatrix} + \mathbf{n}_k^j = \bar{\mathbf{h}}(\bar{\mathbf{q}}, j) \tag{7.47}$$

where $\mathbf{n}_k^j$ is the noise associated with the range and bearing measurements from the jth tag at time instant k. Note that the measurement model (7.47) depends on the robot's pose (x, y, θ) and the jth tag position (x^j, y^j). Similar to [Choi and Lee, 2010], the Jacobian with respect to full state vector is given by

$$\bar{\mathbf{H}}_k^j = \bar{\mathbf{H}}^{'}\bar{\mathbf{F}}^j,$$

where

$$\bar{\mathbf{H}}' = \begin{bmatrix} \frac{\partial \bar{\mathbf{h}}}{\partial x} & \frac{\partial \bar{\mathbf{h}}}{\partial y} & \frac{\partial \bar{\mathbf{h}}}{\partial \theta} & \frac{\partial \bar{\mathbf{h}}}{\partial x^j} & \frac{\partial \bar{\mathbf{h}}}{\partial y^j} \\ \frac{\partial \bar{\mathbf{h}}}{\partial x} & \frac{\partial \bar{\mathbf{h}}}{\partial y} & \frac{\partial \bar{\mathbf{h}}}{\partial \theta} & \frac{\partial \bar{\mathbf{h}}}{\partial x^j} & \frac{\partial \bar{\mathbf{h}}}{\partial y^j} \end{bmatrix}$$

and

$$\bar{\mathbf{F}}^j = \begin{bmatrix} 1 & 0 & 0 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ & & & & & & \\ 0 & 0 & 0 & 0 \dots 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & \underbrace{0 \dots 0}_{2j-2} & 0 & 1 & \underbrace{0 \dots 0}_{2s-2j} \end{bmatrix}.$$

The EKF SLAM gain matrix is then computed as

$$\bar{\mathbf{K}}_k^j = \bar{\mathbf{S}}_k \left(\bar{\mathbf{H}}_k^j\right)^T \left(\bar{\mathbf{H}}_k^j \bar{\mathbf{S}}_k \left(\bar{\mathbf{H}}_k^j\right)^T + \bar{\mathbf{N}}_k\right),$$

where $\bar{\mathbf{N}}_k$ is noise covariance matrix associated with the range and bearing measurements. The updated state and covariance matrix are as follows

$$\bar{\mathbf{q}}_k = \bar{\mathbf{q}}_k + \bar{\mathbf{K}}_k^j(\mathbf{z}_k^j - \hat{\mathbf{z}}_k^j)$$
$$\bar{\mathbf{S}}_k = (\mathbf{I}_{2s+3} - \bar{\mathbf{K}}_k^j \bar{\mathbf{H}}_k^j)\bar{\mathbf{S}}_k.$$

The key implementation steps of the EKF SLAM is similar to the one presented in [Choi and Lee, 2010, Thrun et al., 2005] except that a DDMR model is incorporated in this thesis.

7.5 Simulation Results

We now illustrate the performance of the proposed neighboring optimal controller using a continuous-time model of a mobile robot, which is expected to follow a pre-defined trajectory for the time horizon of $I \equiv [0, 60]$ s. To make the simulation results close to the real experiments, the parameters of the mobile robot (wheel radius and base) are chosen exactly the same as the Scout robot shown in Figure 3.3 in Chapter 3. Its wheel speeds are constrained as $|u_R(t)| \leq u_R^{max} = 10\,\mathrm{rad{\cdot}s^{-1}}$ and $|u_L(t)| \leq u_L^{max} = 10\,\mathrm{rad{\cdot}s^{-1}}$, for $t \in I$. The robot's navigation performance using the nominal optimal control (without RFID measurements) and the neighboring optimal control (with RFID measurements) techniques is presented with some computer simulations which are given in the following sections.

7.5.1 Navigation Performance Using Nominal Optimal Control

In this section, the robot's navigation performance using nominal optimal controller is presented. As illustrated in section 7.2, the robot's model is assumed to be ideal (no process or measurement noise). The validity of the nominal optimal controller is verified for three different test scenarios: stabilizing to a point, tracking a rectilinear trajectory, and tracking a curvilinear trajectory. The performance metrics adopted for this study are the robot's pose (position and orientation) tracking error given by

$$\mathbf{q}(t) - \mathbf{q}^d(t) = \tilde{\mathbf{q}}(t) = \begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \\ \tilde{\theta}(t) \end{bmatrix} = \begin{bmatrix} x(t) - x^d(t) \\ y(t) - y^d(t) \\ \theta(t) - \theta^d(t) \end{bmatrix},$$

for $t \in I$, and the average cumulative position error defined in (3.11) over the time interval of $I \equiv [0, 60]$ s. In order to find the nominal solution as described in section 7.2, the initial

feedback control gain $\mathbf{K}(t)$ is chosen as

$$\mathbf{K}(t) = 10^{-6} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

which is then optimized to find the nominal control $\mathbf{u}^o(t)$ and its corresponding nominal trajectory $\mathbf{q}^o(t)$, for $t \in I$. The sampling time period for the simulation is set to $T = 0.6\,\mathrm{s}$. Note that there is no need for solving the desired control inputs, $\mathbf{u}^d(t)$, from the desired Cartesian trajectory $(x(t), y(t))$, which is not the case for most of the controllers proposed in the literature [Chwa, 2010, Luca et al., 2000].

7.5.1.1 Point Stabilization

The desired trajectory $\mathbf{q}^d(t)$ in this case is a fixed point $\mathbf{q}^d = [3.0\ \mathrm{m}\ \ 8.0\ \mathrm{m}\ \ 90°]^T$, where the robot has to stabilize itself after $60\ \mathrm{s}$ with the feedback control (7.1). The robot's initial pose is set to $(x, y, \theta)|_{t=0} = (0, 0, 28.6°)$. Figure 7.3(a) shows how fast the robot is converging to its desired configuration, where the hollow and solid arrows indicate the robot's initial and final pose, respectively. The initial error shown in Figure 7.3(c) is about $8.5\ \mathrm{m}$ which drastically reduces to $\approx 0.2\ \mathrm{m}$ in about $10\ \mathrm{s}$ before converging and remaining at $\pm\ 20\ \mathrm{mm}$ until end of the time interval. The left and right wheel speeds are shown in Figure 7.3(d). Since initially the robot is far from the desired point, both wheel started almost at their maximum speeds $\approx 10\,\mathrm{rad{\cdot}s^{-1}}$ for about $10\ \mathrm{s}$. It is then natural to gradually reduce speeds as the robot approaches the target. Figure 7.3(b) represents the robot's convergence behaviour from different initial poses (eight initial poses, in this case) towards the target at $(0, 0, 90°)$. As can be seen, the robot is able to stabilize itself to a target pose regardless of its initial position and orientation.

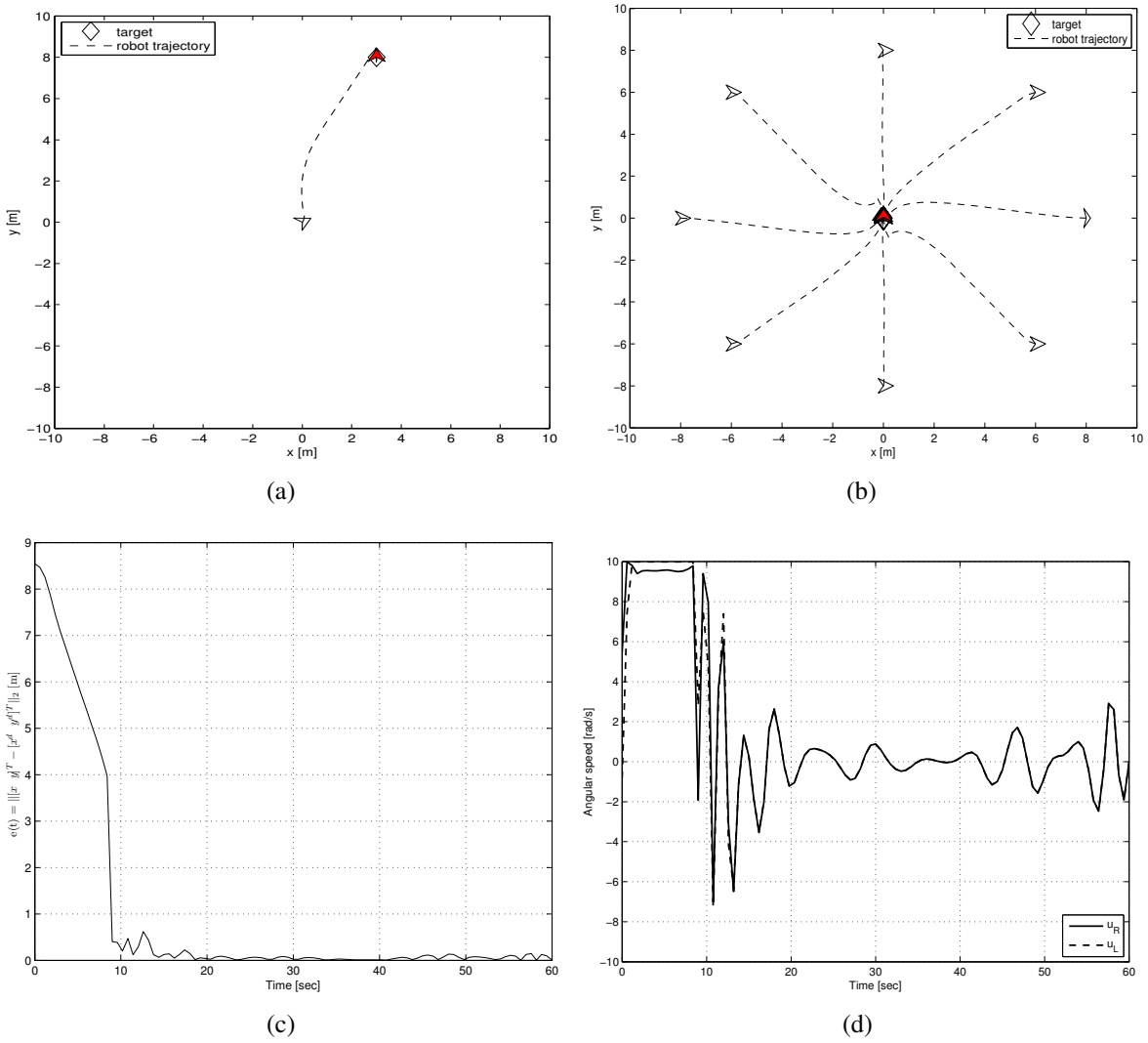


Figure 7.3: Controller's performance for point stabilization: (a) and (b) robot's trajectory from single and multiple initial configurations, respectively; (c) error convergence for Figure (a), and (d) control inputs for trajectory of Figure (a).

7.5.1.2 Tracking Curvilinear Trajectories

In this case, the robot's purpose is to track a sigmoidal trajectory given by

$$y^d(t) = a + \frac{b-a}{1+\exp[-x^d(t)/w]},\ x^d(t) = 10t/60 - 5,$$

and $\theta^d(t) = \tan^{-1}[dy^d/dx^d]$, for $t \in [0, 60]$ s, $a = -5$, $b = 5$, $w = 0.7$. The initial robot pose is $(-3,\ -3,\ 45°)$. Fig 7.4 shows the controller's performance. Figure 7.4(c) shows how the pose tracking error converges to zero, yielding an average cumulative position error ($\mathscr{E}/60$ of (3.11)) of only 0.09 mdespite the large initial error. This error is insignificant with respect to the total distance (19.3 m) traveled by the robot. The time history of the actuator's control inputs (speeds) are revealed in Figure 7.4(e). As expected, the robot's speed is zero at the final destination of the trajectory.

In order to show the robot's tracking ability regardless of the complexity of the desired trajectories, we tested with a parabolic trajectory defined as $y^d(t) = 0.4[x^d(t)]^2$, $x^d(t) = 10t/60 - 5$, and $\theta^d(t) = tan^{-1}[0.8x^d(t)]$. As can be seen from Figure 7.4(b), the robot could still track the desired trajectory even though its initial pose is far from the desired one. Figure 7.4(d) shows that the robot has quickly converged to yield a tracking error ≈ 0. The higher tracking errors at about $t = 30$ s are due to the sharp bend of the desired trajectory. The left and right wheel angular speeds are almost same all the time and zero at the end (see Figure 7.4(f)). The position tracking error, $\mathscr{E}/60$, recorded in this case was 0.18 m. This is mainly due to the large initial error and the sharp turn in the path. The total distance traveled by the robot was 23.7 m.

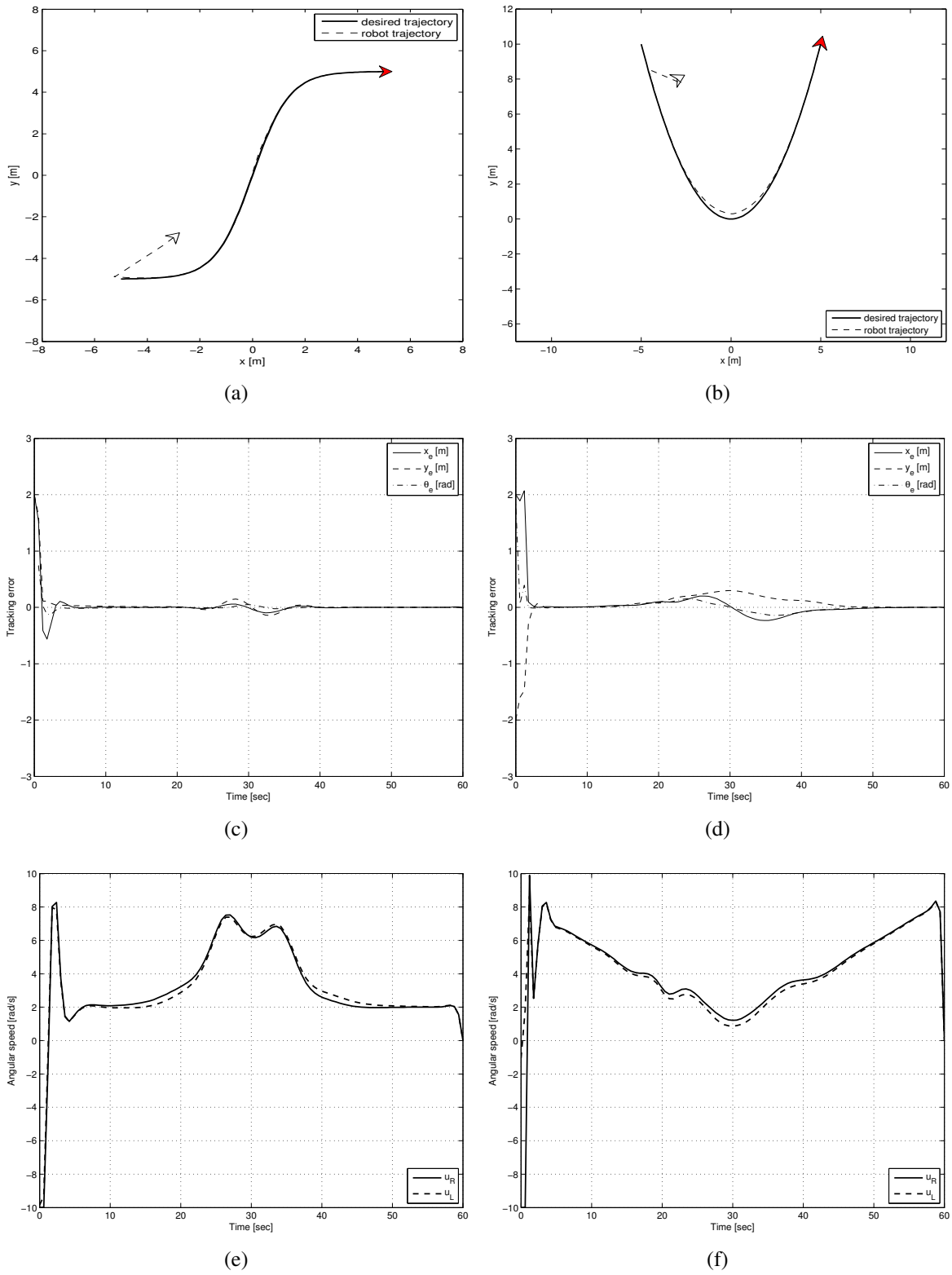


Figure 7.4: Performance of tracking sigmoidal and parabolic trajectories: (a) and (b) robot's trajectory; (c) and (d) tracking error, and (e) and (f) control inputs.

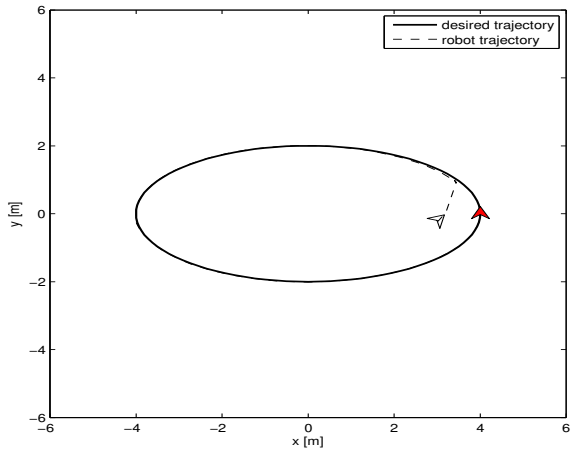
7.5.1.3 Tracking an Elliptic Trajectory

Attention is now shifted to test the robot's tracking ability along an elliptic trajectory, which is given by $x^d(t) = 4\cos(2\pi t/60)$, $y^d(t) = 2\sin(2\pi t/60)$, and $\theta^d(t) = \tan^{-1}[\dot{y}^d(t)/\dot{x}^d(t)]$. The robot is initially placed at $(3,\ -0.2)$ m with an initial orientation of $45°$. The results are summarized in Figure 7.5. Despite deliberately setting a large initial error, the robot quickly converged to the target trajectory. Tracking errors in Figure 7.5(b) reveal overshoots at the beginning but decayed down to zero after about 6 s. The boundedness of the wheel speeds are clear from Figure 7.5(c). The average position tracking error throughout the whole trajectory ($\mathscr{E}/60$ of (3.11)), including the significantly large initial error, is 0.07 m. These error values are satisfactory taking into account the total distance traveled (19.6 m) and the complexity of the robot trajectory.

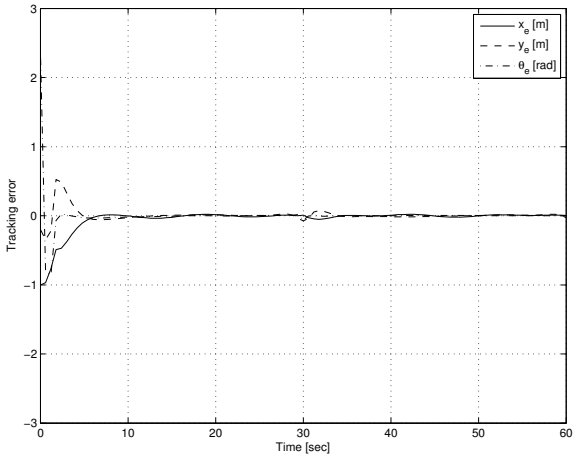
7.5.2 Navigation Performance Using Neighboring Optimal Control

As can be seen, the nominal optimal controller is able to successfully navigate the robot regardless the complexity of desired trajectories. However, the robot model used to demonstrate the performance of the nominal optimal controller is assumed to be ideal. It is certain that the deterministic nominal control signals can not provide satisfactory performance for trajectory if the robot is deployed in real-time. In addition, the navigation error gets accumulated if the robot is driven over a long period of time. This limitation is successfully tackled by placing 25 ($s = 25$) RFID tags uniformly placed in 3-D positions in the workspace. The RSS model and noise adopted in these simulations are the same as those described in Section 6.5.1.

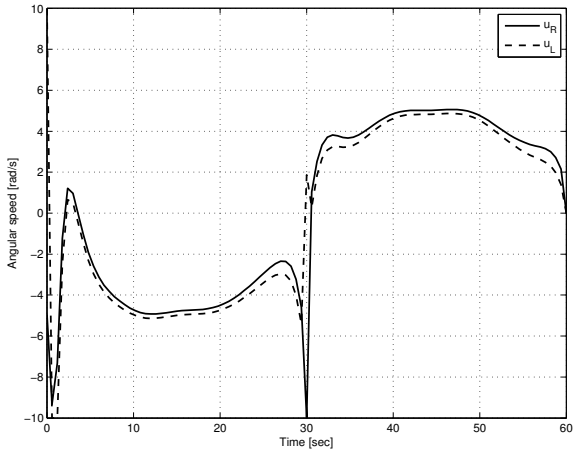
The nominal trajectory and control signals, $\mathbf{q}^o(t)$ and $\mathbf{u}^o(t)$ are employed to compute the on-line neighboring optimal control signals. In the following sections, the robot's stabilization and tracking abilities in the presence of model uncertainty and noisy RSS measurements from the RFID tags are presented.



(a)



(b)

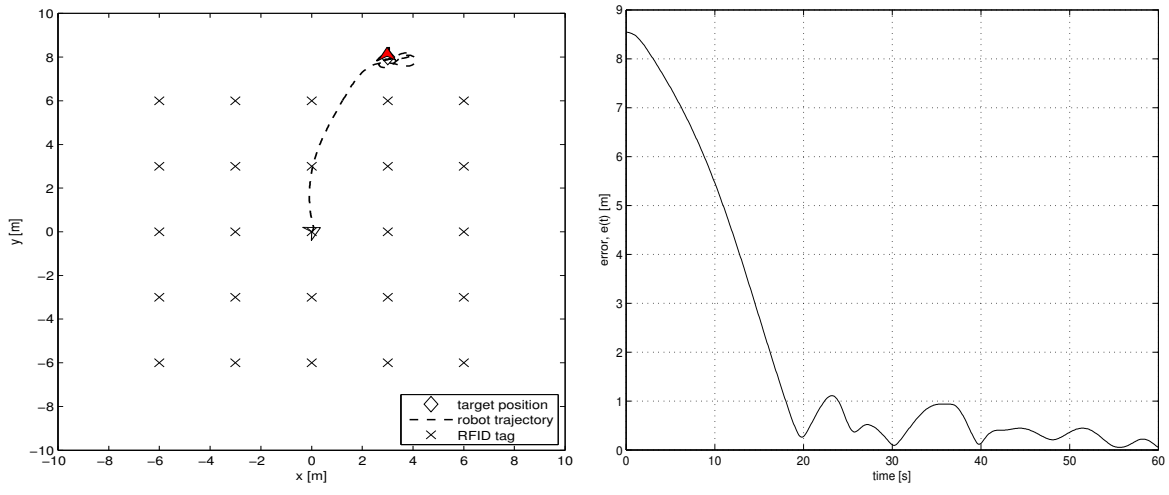


(c)

Figure 7.5: Performance of tracking elliptic trajectory: (a) robot's trajectory, (b) tracking error, and (c) control inputs.

7.5.2.1 Robot Stabilization on a Fixed Configuration

In this section, we present the robot's ability to stabilize on a fixed configuration regardless of its initial position and orientation. The stabilization performance of the neighboring optimal control scheme is evaluated by choosing the weight matrices as $\mathbf{P}(t_f) = diag(2, 2, 2)$ and $\mathbf{Q}(t) = diag(0.01, 0.01, 0.01)$, $\forall\, t \in I$. Hence, the stabilization at the target point is regarded 20 times more important than guiding the robot towards the target. The robot's goal is to stabilize itself at $(x, y) = (3, 8)$ m with an orientation of $90°$. The initial position and orientation of the robot are set to $(0, 0)$ m and $28.6°$, respectively. Figure 7.6(a) shows the simulation results, where the hollow and solid arrows represent the initial and final poses, respectively. The dashed path represents the robot's actual trajectory while the x's depict the 2-D projections of the RFID tags mounted on the ceiling. The distance between the robot and its target is shown in Figure 7.6(b). It reveals how fast the robot is approaching towards the target. The robot reached its target in about 20 s. Then, after some zigzagging, it could stabilize itself eventually with a position error of practically nil.

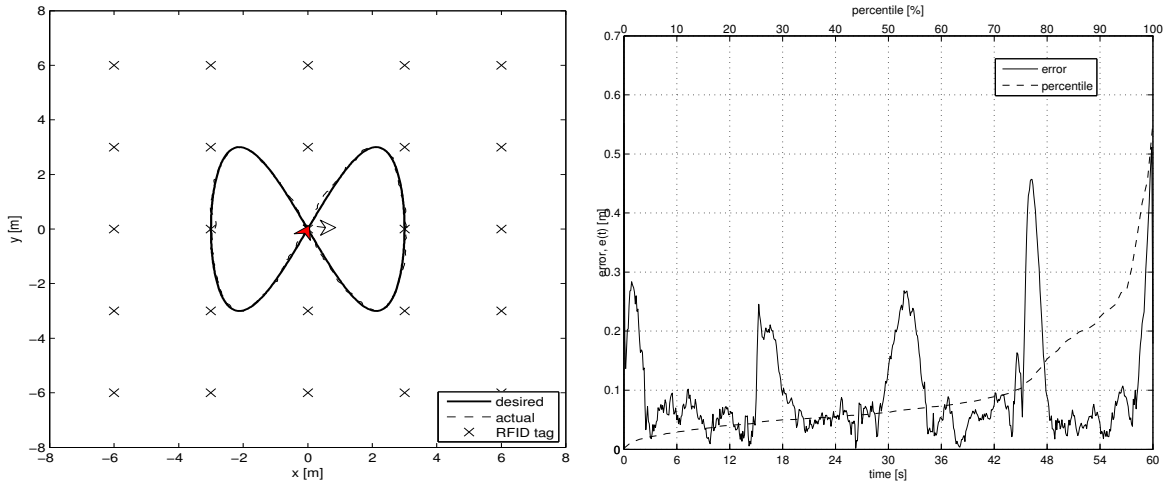


(a) (b)

Figure 7.6: Controller's performance in stabilizing a fixed point (a) optimal trajectory and (b) tracking error.

7.5.2.2 Tracking a Curvilinear Trajectory

The purpose of this test is to study the robot's tracking ability along a complex trajectory. To do that, we define a desired trajectory as $x^d(t) = 3\sin(\pi t/30)$, $y^d(t) = 3\sin(\pi t/15)$, and $\theta^d(t) = \tan^{-1}\left(\dot{y}^d/\dot{x}^d\right)$, for $t \in I$. The robot is initially placed at $(0.5, 0)$ m with an initial orientation of $0°$. Figure 7.7 summarizes the performance of the proposed neighboring optimal controller. The tracking error is plotted in Figure 7.7(b). Starting off its desired path, the robot converges in less than 3 s while keeping the left and right wheel rotational speeds within their limits. It is noticed that the robot looses track of its trajectory momentarily at a few sharp turns before converging back to it. On the same figure, the percentile plot (whose x-axis is on top of the figure) shows that the tracking error is kept less than the average value (0.1 m) for about 75% of the time, and less than 0.22 m for 90% of the time. Once again, taking into account the aforementioned constraints, which are quite typical in any real-world robotic system, these values are considered very satisfactory.



(a) (b)

Figure 7.7: Robot's performance in following a curvilinear trajectory (a) optimal trajectory and (b) tracking error.

7.5.3 Analysis on Simulation Results

Robot's navigation performance using the nominal optimal control strategy (see section 7.5.1) is certainly better based on the assumption that the robot can perfectly estimate it's pose at any time $t \in I$. Using such assumption, the tracking errors achieved in Figures 7.3(c), 7.4(c), 7.4(d), and 7.5(b) are practically negligible compared to the robot's path complexities. However, this assumption violates the applicability of the controller in real-life due to the noise associated with the robot model itself. This limitation is clearly tackled using the partially-observed measurement model (RSS measurements from RFID tags) and the robot's stabilization and tracking performance is revealed in section 7.5.2. Note that the practically "nil" stabilization error (see Figure 7.6(b)) was achieved despite the excessively noisy RSS signals transmitted by the tags and the noisy actuator signals of the robot. The zigzagging behavior is expected due to the complexity of this task, especially for nonholonomic robots. As for the trajectory tracking error shown in Figure 7.7(b), the average tracking error throughout the whole trajectory ($\mathscr{E}/60$ of (3.11)) is $0.1\,\mathrm{m}$. This is a very small error taking into account the total traveled distance of $30.8\,\mathrm{m}$, the wheel speed constraints (maximum of $10\,\mathrm{rad{\cdot}s^{-1}}$), and the excessive noise associated with the RF signals transmitted by the RFID tags. It is important to articulate that this error is non-cumulative. It is rather affected by the signal-to-noise ratio of the RF signals, but not by the traveled distance or navigation time.

7.5.4 Comparison with RH Open-Loop Controller

The tracking performance of the proposed neighboring optimal feedback controller is compared with the relavant RH open-loop optimal control method presented by Gu and Hu in [Gu and Hu, 2006] which is tested on a two-wheel mobile robot. Based on the above results, some important observations can be made with respect to the results presented in [Gu and Hu, 2006]. In RH optimization method, the control time horizon is dependent on the complexity of reference

trajectories, whereas it is independent of the complexity of the reference trajectories in the proposed control scheme. The proposed optimal feedback controller is simpler in the sense that it is only a matter of determining the optimal feedback operator $\mathbf{K}$ to solve both stabilization and tracking problems. In addition, the suggested controller achieves faster convergence towards the desired trajectories. Furthermore, it is tested with longer trajectories to deliberately show the satisfactory performance.

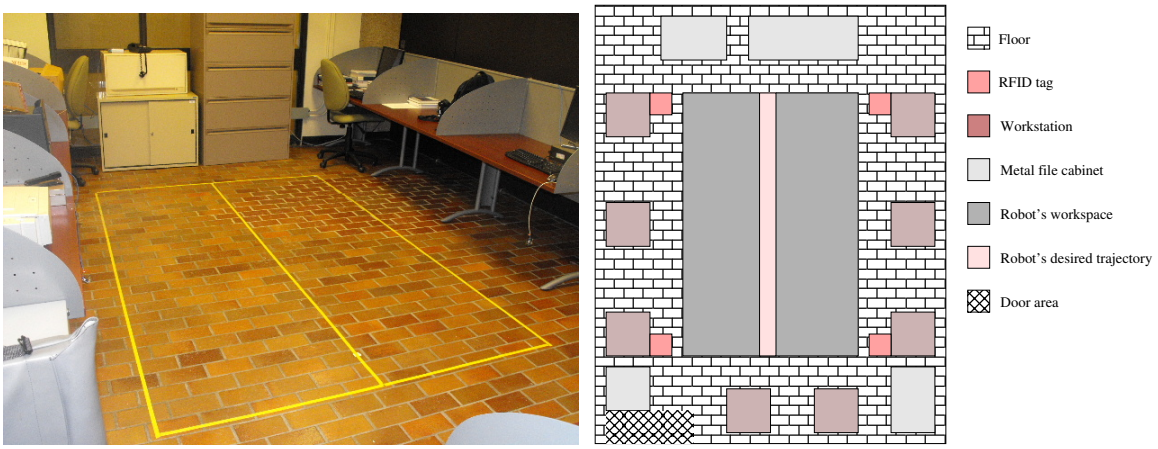
7.6 Experimental Results

The section presents the results demonstrating the real-time performance of the neighboring optimal controller where a robot is supposed to track a pre-defined trajectory and stabilize a fixed configuration in indoor lab environments. For that, three different experiments are conducted where first two experiments demonstrate the robot's tracking ability and the third experiment yields the robot's stabilization performance.

7.6.1 Tracking a Line Segment

The first experiment is carried out to show the robot's tracking performance along a simple straight-line trajectory in a lab environment of about $\approx 5 \times 5$ m^2 at the University of Ottawa. The environment is populated with metal cabinets, computers, and obstacles of various metal components. The test scenario for the robot is a 2×3 m^2 free space in the middle of the laboratory. The workspace seen by the robot is pictured in Figure 7.8(a). Figure 7.8(b) shows the top-view of the floor plan where four XBee modules (emulating RFID tags) are located at 3-dimensional positions $(-0.2, 0.1, 0.5)$ m, $(2.1, 0.1, 0.5)$ m, $(2.1, 3.1, 0.5)$ m, and $(-0.2, 3.1, 0.5)$ m with their corresponding 16-bit IDs 0x5001, 0x5002, 0x5003, and 0x5004, respectively. The robot is mounted with a MakeController kit with a base XBee module of ID

0x5000, which is acting as an RFID reader.

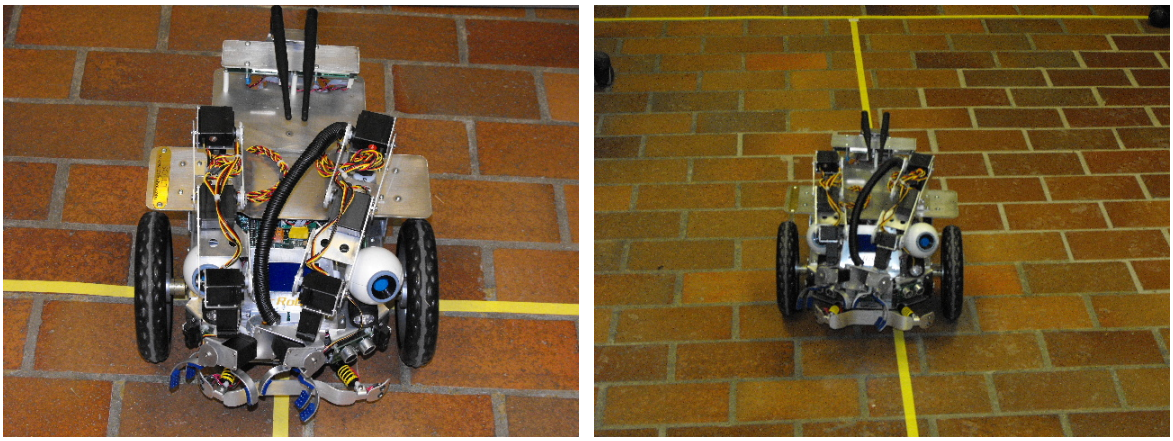


(a) (b)

Figure 7.8: (a) Robot's workspace and (b) floor plan.

Due to limited free workspace in the laboratory, a straight-line trajectory with $x^d(t) = 1,$ and $y^d(t) = 3t/t_f$, and $\theta^d(t) = \pi/2, \forall t \in [t_0, t_f]$ is chosen as the desired trajectory that the robot is supposed to track in $t_f = 30$ s with its initial position and orientation being right at the beginning of the desired trajectory, $x(0) = 1,\ y(0) = 0, \theta(0) = \pi/2$. The results are summarized in Figures 7.9 and 7.10. Figure 7.9 shows the robot's tracking capability along the line without taking into account the RSS measurements from RFID tags. As can be seen, initially it started tracking the line (see Figure 7.9(a)) but diverging along the way (see Figures 7.9(b), 7.9(c), 7.9(d)). This is natural due to the parametric uncertainty of the robot's wheel, actuators, and odometric sensors.

Figure 7.10 demonstrates the robot's trajectory tracking ability by taking into account the RSS measurements from the RFID tags emulated by the XBee modules. In this case, the robot broadcasts a command signal to all RFID tags in its operating range at every sampling time interval, the tags then backscatter their RSS values along with IDs to the RFID reader mounted on the robot. These RSS measurements are filtered to estimate the robot's pose using the optimal filter as discussed in section 7.3.2. This is followed by computing the

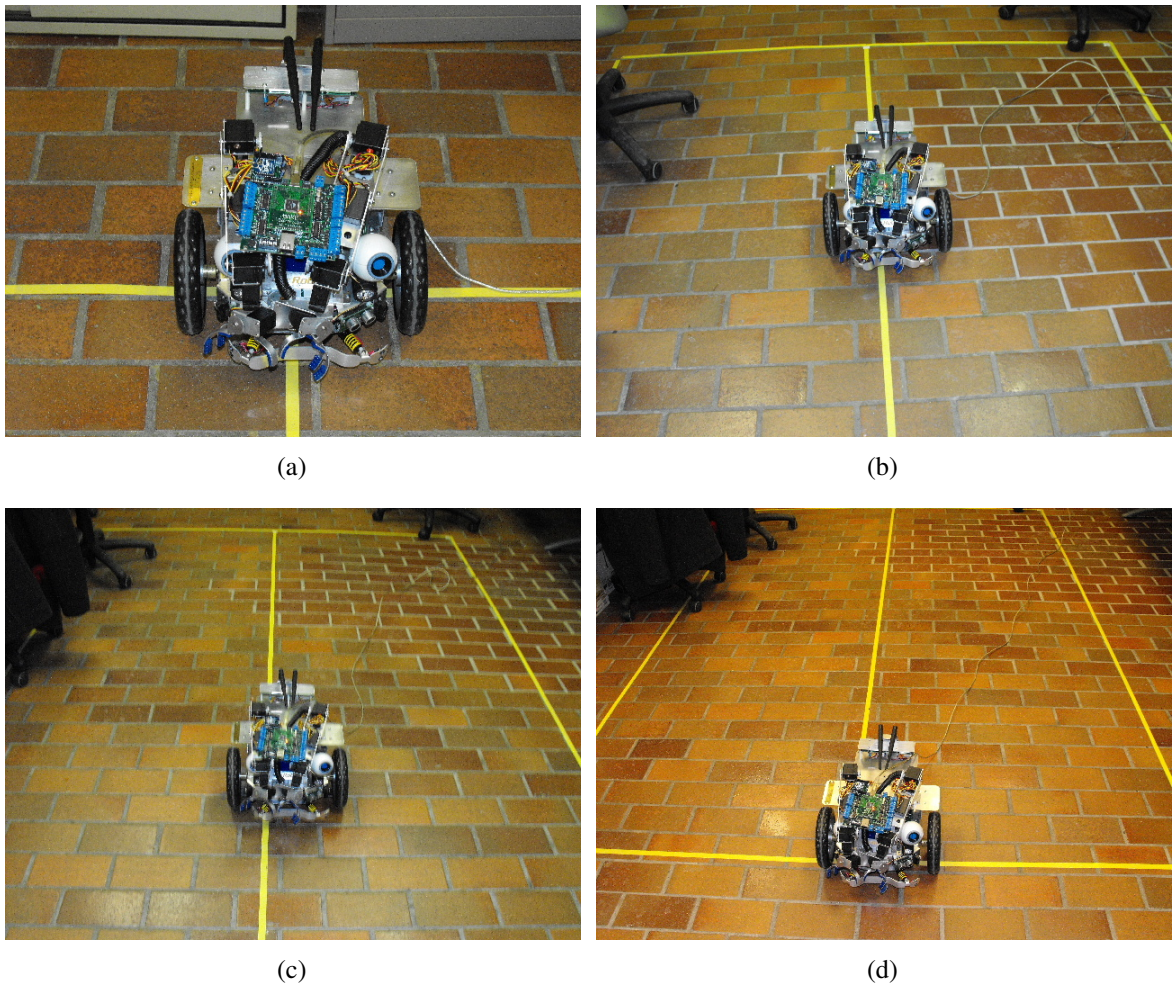


(a) (b)

(c) (d)

Figure 7.9: Controller's real-time performance for line tracking (optimal state feedback control): (a) robot placed at initial position, (b) robot on its way, (c) robot on its way, and (d) robot is diverging from desired trajectory.

actual control speeds (left and right wheel angular speeds) as in (7.34). As can be seen from Figures 7.10(a), 7.10(b), 7.10(c), and 7.10(d), the tracking error is negligible compared to that of the experiments without RSS measurements (see Figure 7.9). The robot has stopped at the final target destination of the trajectory with an error of about 13cm.



(a) (b)

(c) (d)

Figure 7.10: Controller's real-time performance for line tracking (neighboring optimal control): (a) robot placed at initial position, (b) robot on its way, (c) robot on its way, and (d) robot reached the end of trajectory.

Figure 7.11: Sputnik mobile robot used for tracking an U-shaped trajectory in the MIRaM laboratory.

7.6.2 Tracking an U-shaped Trajectory

The purpose of this experiment is to show the modular property of the proposed neighboring optimal controller suggested in the Chapter. For that, the Sputnik[1] differential drive mobile robot shown in Figure 7.11 is deployed. The hardware configuration of this robot is same as that of Scout robot except that the chasis diameter of the Sputnik robot is 26cm. The MakeController board emulating the RFID reader is mounted on the robot. This board is however is connected to a laptop computer using an USB cable which allows the robot to receive the RSS measurements coming from the XBee modules. Note that the laptop computer and the robot is wirelessly connected (locally) through an wireless router in the robot's workspace.

This experiment is conducted in the MIRaM laboratory of dimension about $10 \times 9 \times 3$ m at the University of Ottawa. The top-view of the workspace floor plan is depicted in Figure 7.12. The robot is supposed to follow the U-shaped rectilinear trajectory of length 16.5 m which is

[1] http://www.drrobot.com/

divided into three unequal segments (7.5 m + 4.0 m + 5.0 m = 16.5 m). The four XBee modules in this case are located at positions $(1.7, 5.0, 0.7)$ m, $(-1.5, 4.9, 0.7)$ m, $(-1.5, 2.1, 0.7)$ m, and $(1.0, 2.6, 0.7)$ m and their corresponding 16-bit IDs are same as the previous experiment. The robot is initially placed at the beginning of the segment of length 7.5 m (see Figure 7.13(a)). Since the robot receives the RSS measurements from four XBee modules placed in the vicinity of this segment, it's navigation performance is quite satisfactory in the sense that the tracking error at any instant of time is about 10 cm. The snapshots of this experiment while the robot is navigating along the U-shaped trajectory are summarized in Figure 7.13, where Figures 7.13(a)-7.13(f) reveal the navigation performance of the first segment. After that, the robot had to turn at the first sharp corner to follow the second segment of length 4.0 m (see Figure 7.13(h)). Note that the robot is no longer in line-of-sight with the XBee modules which shows the power of RFID systems in navigating a mobile robot in indoor environment. When the robot is in the segment of length 5.0 m, it is completely out of line-of-sight from the XBee modules since the workstations are in between the first and the last segments. As can be seen from Figures 7.13(m)-7.13(p), the robot is still able to track this line segment with the tracking error of about 8 cm. It is important to articulate the fact that the main purpose of this experiment was to track the desired trajectory rather than stabilizing on a fixed point. Hence, the robot has stopped at about 50 cm away from the desired end point the trajectory (see Figure 7.13(p)).

7.6.3 Real-time Performance in Stabilizing on a Fixed Configuration

As stated in the literature, stabilizing a wheeled mobile robot to a point is more difficult than path following or trajectory tracking. Hence, the performance of the controller is demonstrated using the Scout mobile robot for stabilizing it to a fixed configuration in a real-time environment. The snapshots of the experiment are shown in Figure 7.14. The initial position is set at the origin $(0, 0)$ of the test area with an initial orientation of $90°$ (see Figure 7.14(b)) in a Cartesian

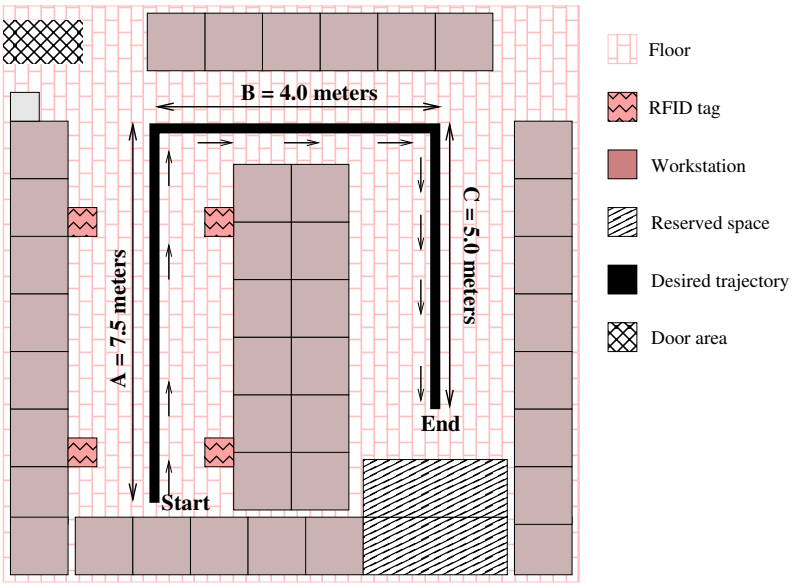


Figure 7.12: Floor plan of the robot's workspace at the MIRaM laboratory.

coordinate system. Figure 7.14(c) depicts the posture of the robot on its way. Initially, the robot travel at its maximum available speeds and gradually decreases its speed as it approach towards the target point. This makes sense as the controller avoids the overshoot which may arise once it reaches the target. Hence, the robot reaches the target at exactly 30 s and stabilized therein. The robot has finally reached close to its desired point which is located at $(2, 4)$ m with the desired orientation of $90°$. As can be seen from Figure 7.14(d) that the robot has stopped about 20 cm away from the desired point and the orientation error is about $20°$.

7.6.4 Analysis on Experimental Results

An significant remark that can be made based on the above experimental results is that the proposed controller is independent of the robot's hardware architecture. The proposed neighboring optimal control algorithm is employed in two different mobile robots in order to deliberately show its modularity. Even though the nominal control scheme is perfect (tracking error ≈ 0) for simulating the robot's tracking performance, it is not the case when it is applied on a real robot

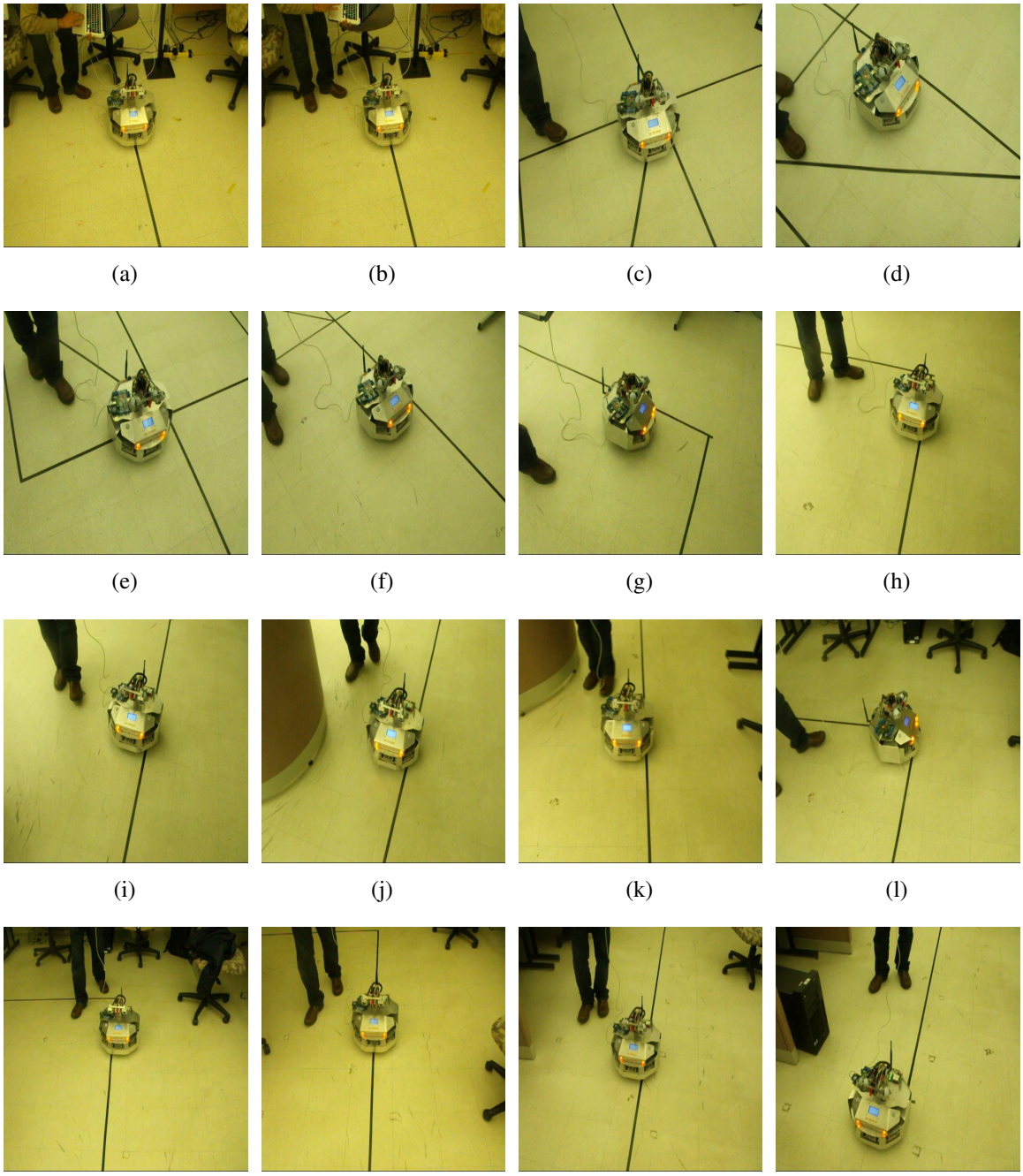


Figure 7.13: Robot's real-time performance for tracking a rectilinear trajectory using neighboring optimal control.

(a) (b)

(c) (d)

Figure 7.14: Controller's real-time performance for point stabilization: (a) initial position, (b) robot placed at initial position, (c) robot on its way, and (d) robot is close to the target.

in an operating environment. This is clearly shown in Figure 7.9 as the robot was diverging from the desired trajectory. On the other hand, the neighboring optimal controller using RSS measurements successfully guides the robot along the same trajectory despite the significant noise which are bounced off the walls and metal obstacles in the workspace.

The second experiment (tracking a U-shaped trajectory), however, reveals that the robot has the ability to generate appropriate control signals to its actuators which are based on the RSS measurements despite the non line-of-sight communication between the RFID reader mounted on the robot and the RFID tags placed in the workspace. The higher position error (≈ 50 cm) at the end of the trajectory was due to the larger tracking weight matrix because of its priority rather than reaching target. The third and the last experiment (stabilizing on a fixed configuration) also provided satisfactory performance in reaching the target destination even though the robot's final position was off by about 20 cm. This error is, however, acceptable taking into account the uneven surface of the lab.

7.6.5 Comparison with the Sliding Mode Control Scheme

There are not many optimal control schemes in the literature that solve both stabilization and tracking problems of nonholonomic mobile robots using noisy RFID systems. Nevertheless, the experimental results of the proposed neighboring optimal control scheme are compared with the conventional sliding mode control (SMC) method using an RFID sensor space presented by Lee et al. in [Lee et al., 2009]. It is important to point out that, the average position tracking error for the proposed control scheme is about 25 cm which is almost same as the tracking performance presented in [Lee et al., 2009]. However, the suggested controller is taking into account the time constraint on the robot's geometric path as opposed to [Lee et al., 2009]. Moreover, the proposed controller does not require the reference robot velocities for generating reference trajectories as it is the case for most of the tracking and stabilization control strategies

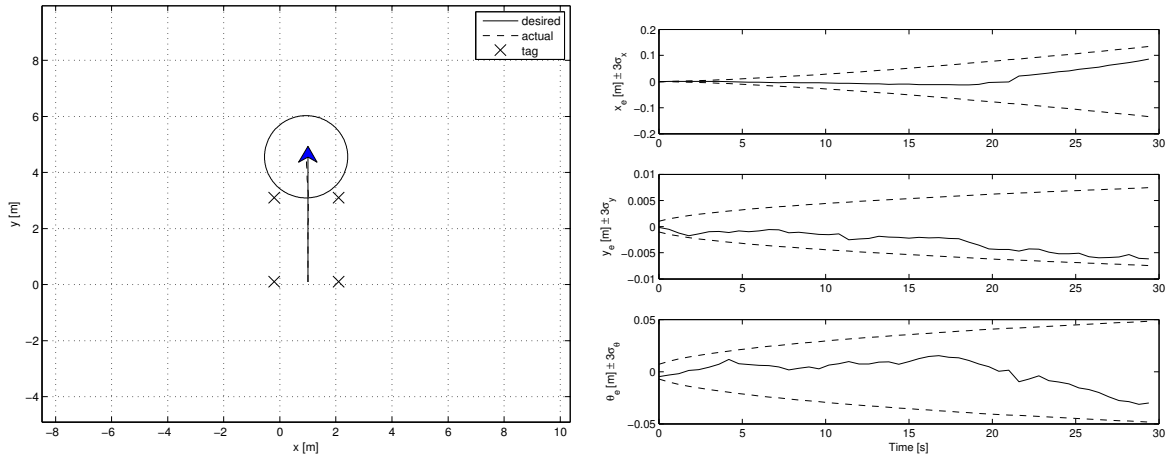
including [Lee et al., 2009]. In addition, it is worth noting that the robot's stabilization at a fixed configuration becomes the special case of tracking a reference trajectory which is not the case in [Lee et al., 2009], for example.

7.7 Pose Estimation Comparison With the EKF SLAM

In order to assess the performance of the neighboring optimal controller for navigating mobile robot in indoor environment, its state estimation performance is compared with a conventional SLAM algorithm, EKF SLAM. Since, EKF SLAM is a state estimation (localization) method (not navigation), neighboring optimal controller's state performance is compared with that of the EKF SLAM. For that, EKF SLAM is tested on the same robot model with the constant wheel speed of $2\,\mathrm{rad{\cdot}s^{-1}}$ (both left and right wheels). The standard deviation of the noise associated with the speed is chosen as $0.01\,\mathrm{rad{\cdot}s^{-1}}$. The noise associated with the range and bearing measurements are also taken into account whose standard deviations are $1.5\,\mathrm{m}$ and $0.9\,\mathrm{rad}$, respectively. The workspace and tag locations remain the same as in the real experiments with the neighboring optimal controller (see Figure 7.8).

Figure 7.15 demonstrates the state estimation performance of the robot traveling along the $3\,\mathrm{m}$ long straight line trajectory. The tag locations, and the robot's desired and actual trajectory are shown in Figure 7.15(a). The ellipse shown at the final position of the robot $\approx (1, 3)$ m indicates the uncertainty of its states. This is clearly shown in the error plot revealed in Figure 7.15(b). Note that $(\pm\sigma_x, \pm\sigma_y, \pm\sigma_\theta)$ in Figure 7.15(b) show the error bound of the state along the robot's position (x, y) and orientation θ. As can be seen, the robot could still estimate its state using EKF SLAM with a satisfactory performance. This is due to the fact that the robot's initial perturbation was negligible and the robot's model and the measurement model were linearized around the desired trajectory as illustrated in section 7.4. However, the robot is diverging as it travels towards the end of the trajectory. On the contrary, the trajectory tracking

showing the robot's poses on the same geometric line is presented in Figure 7.10. Note that robot is not diverging at the end of the trajectory as opposed to that of the EKF SLAM. Moreover, the EKF SLAM fails to perform state estimation if the process noise (noise associate the wheel speed) standard deviation is higher than $0.01\,\mathrm{rad{\cdot}s^{-1}}$ which is quite possible with many real robots.



(a) (b)

Figure 7.15: Robot's state estimation performance using EKF SLAM (a) robot's actual and desired trajectory; and (b) state estimation error.

7.8 Summary

In this chapter, a neighboring optimal control strategy for solving trajectory tracking problem is proposed. It relies on dividing the whole control process into two sub-processes: finding nominal and neighboring optimal control inputs. The nominal trajectory is computed off-line which is deterministic. The neighboring trajectory is computed on-line. It requires an optimal filter for estimating the robots pose taking into account the noise associated with the RSS measurements from the RFID tags and the robot's wheel speeds. The actual control actions are then computed by the sum of the nominal and the neighboring control inputs which lead the

robot to track its pre-defined desired trajectory. Simulation results using the nominal optimal controller revealed the robot's ability to stabilize on a fixed configuration and to guide itself along a pre-defined trajectory with a satisfactory tracking error (about 10 cm in average) despite the large initial error. The neighboring optimal control strategy is first simulated to realize it's real-time performance in solving tracking and stabilization problems of mobile robot. Since the RSS measurements and the robot's model uncertainty are taken into consideration, the average tracking error was higher than that of the nominal controller's performance, as expected. The real-time navigation performance of the neighboring optimal controller is evaluated using the two commercial robots (scout and sputnik) which were deployed in two different lab environments. Surprisingly, the navigation errors while tracking the pre-defined trajectories in real-time environment were almost same as those in simulations. Another interesting feature which is worth to point out that the same control algorithm is implanted in two different mobile robots for tracking pre-defined trajectories. This shows the modular property of the proposed control strategy suggested in this chapter. Finally, the simulation and experimental results clearly demonstrate the robot's ability to navigate along pre-defined trajectories regardless of their complexities with negligible tracking error despite the ambient noise due to RSS signal reverberations. Unlike other research works on RSS-based tracking system which are mostly dependent on environment, there is no restriction on the placements of RFID tags in the robot's workspace. It is important to stress that RSS measurements from more RFID tags significantly improves the robot's tracking ability even in a very complex desired trajectory which is natural.

Chapter 8

Conclusion and Future Work

8.1 Thesis Summary and Analysis

This dissertation has contributed three different control techniques for navigating a differential drive mobile robot in indoor environments based on minor customizations of the existing commercial RFID system architecture. For convenience, these controllers are reiterated as follows:

1. Bearing-sensitive controller (Chapter 5)
2. Optimal measurement-feedback controller (Chapter 6), and
3. Neighboring optimal controller (Chapter 7).

Based on the previous description and results, some observations can be drawn after weighing the pros and cons of the proposed control methods for mobile robot navigation.

Chapter 5 presents a novel robot navigation system using a bearing-sensitive RFID reader which is specifically designed to operate in indoor reverberant environments. An accurate mathematical model to map the RSS measurements to the LOS distance is also not required

in this navigation technique. A series of computer simulations is conducted to evaluate the performance of this navigation system where a commercial electromagnetic simulation software (FEKO) is adopted to realistically assess the reverberations effect on the developed architecture. It has been shown that the robot successfully converges to its pre-defined destinations under various magnitudes of reverberations. This navigation scheme is simple in the sense that robot's target destinations are simply specified by RFID tags. However, it fails to track a nonlinear path on the ground which is independent of RFID tag locations. In addition, this navigation scheme is unable to deal with time constraint on the robot's desired trajectory.

Chapter 6 overcomes these issues by designing an optimal measurement (output) feedback controller based on a modified RFID tag architecture. In this case, RSS measurement are used as feedback information to generate control actions for the robot's actuators. Similar to the previous navigation scheme, determining the LOS distance between the robot and the RFID tags is also not absolutely necessary in this method. One appealing feature of the optimal feedback control strategy presented in this chapter is that it has the ability to solve both tracking and stabilization problems of a nonholonomic mobile robot in a unified manner. However, since the robot itself is subjected to external disturbances that can not be known ahead of time, generating feedback control based on RSS measurements is not adequate for highly accurate tracking performance. In addition, relatively high computational resources are required for the robot to track a pre-defined trajectories in real-life scenarios.

The difficulties addressed in Chapters 5 and 6 are better tackled in Chapter 7 by partitioning the robot's control problem into two parts: the nominal (deterministic) and the neighboring control solutions. The former represents an off-line solution and the latter represents an on-line real-time tuning mechanism. Hence, the resulting optimal trajectory for the mobile robot could be well approximated by the sum of the nominal and the neighboring optimal trajectories. Simulation and experimental results demonstrate the robot's ability to guide along a pre-defined

Table 8.1: Summary of different features of the proposed control schemes.

Features \ Controller	Bearing-sensitive	Optimal measurement-feedback	Neighboring optimal
RFID architecture	modified RFID reader	modified RFID tag	modified RFID tag
RFID tag locations	not known to robot	not known to robot	known to robot
Computational complexity	$\mathcal{O}(s+c)$, where c is a constant	$\mathcal{O}(nsk_f t_f N)$	$\mathcal{O}(n^2 k_f t_f N + nsk_f t_f)$
Dynamic linearization	Not required	Not required	Not required
Trajectory tracking capability	No	Yes	Yes
Point stabilization capability	No	Yes	Yes

trajectory with a satisfactory tracking error (in the range of 0.1 – 0.2 m). An important point that can be drawn from the experimental results of this controller is that the neighboring optimal controller is independent of the robot's hardware architecture. In addition, RFID systems' non-LOS property has been effectively demonstrated in navigating a mobile robot in an indoor environment.

Table 8.1 summarizes the important features of the three control algorithms proposed in this thesis for navigating a mobile robot in an indoor environment. The bearing sensitive controller requires a modified RFID reader architecture without changing that of the tags unlike the other two controllers which require a modified RFID tag architecture without changing that of the reader. The RFID tag locations do not have to be known to the robot in the case of the first two controllers as opposed to the third controller (neighboring optimal controller). The computational complexity of the bearing sensitive controller depends on the number of RFID tags. It also relies on the complexity of the FLC which is basically constant in this case. On the other hand, the other two controllers are more complex as their computational complexities are $\mathcal{O}(nsk_f t_f N)$ and $\mathcal{O}(n^2 k_f t_f N + nsk_f t_f)$, respectively, where N is the number of off-line

iterations of the optimization procedures employed in these controllers and n is the number of degrees of freedom of the robot ($n = 3$ in this thesis). It is important to note that the dynamic linearization of neither the robot model nor the measurement model is required for all the three controllers. Finally, the bearing sensitive controller is unable to track a geometric path or to stabilize at a fixed configuration with an associated time law, unlike the other two controllers.

8.2 Conclusion

Mobile robots using the proposed control (navigation) strategies presented in this thesis can be operated as service robots in indoor environments, such as hospitals, museums, offices, for example. Moreover, these navigation schemes can be applied where the robot is also constrained in time to accomplish its mission. Undoubtedly, the proposed navigation techniques are modular in the sense that they are independent of any specific robot architecture. They are also cost-effective in the sense that RFID tags are extremely cheap (in the range of few cents each), hence, increasing the number of RFID tags in the robot's operating environment will not represent a major burden to the total cost of the navigation systems. Sometimes it may be worth to increase the number of RFID tags as it improves the robot's navigation accuracy. An important point to note is that RSS measurements from RFID tags are effectively used for navigation purpose and not just for localization of a mobile robot which is another contribution of this thesis. This is due to the fact that RSS measurements can be obtained relatively effortlessly and inexpensively. In addition, no extra hardware (e.g., ultrasonic or infra-red) is needed for network-centric localization. The RFID systems can be deployed in an environment where the robot needs to be navigated at night, which may not be possible using vision-based navigation systems, for example. Having said that, the proposed navigation techniques are not meant to substitute vision-based navigation algorithms. Rather, they might be regarded as alternative navigation solutions for many robotic applications where vision systems might not be necessary.

Hence, this study opens the doors for a new class of robot navigation techniques that are simple, modular, and cost-effective.

8.3 Future Research Directions

Even though navigation algorithms in this thesis yielded promising performance for indoor differentially-drive wheeled robots, they can easily be applied to car-like robots in indoor/outdoor environments as well. In continuation of the current work on developing optimal feedback laws for control-affine nonlinear systems, the next stage is to incorporate obstacle avoidance strategies for path planning problems by mounting some range sensors on the robots which can be deployed in indoor or outdoor applications. Furthermore, it can be envisioned such control methods to be applicable in underground mining vehicles, Unmanned Aerial Vehicles (UAV), and, in general, a wide range of mechatronic applications. Fusing multiple sensory data (vision, proximity, for instance) along with RFID systems can extend the possibilities of applying the proposed navigation methods in indoor/outdoor environments using just state feedback control mechanisms.

Appendix A

Definitions and Proofs

A.1 Definitions

In this section, some terms that are used throughout the thesis are defined. In addition, a number of important definitions from functional analysis that are applied in this thesis are presented without proofs. These definitions are mostly repeated from [Ahmed, 2006, Knapp, 2005].

Definition A.1 (Navigation/Motion control)**.** *In general, navigation is the process of observing the robot's environment through sensors and controlling its movement through the actuator signals for travelling from one place to another.*

Definition A.2 (Motion planning or path planning)**.** *It is the process of performing the navigation task of a robot through a series of discrete motions while avoiding obstacles in its operating environment.*

Definition A.3 (Localization)**.** *It is the process of determining the robot's position and orientation (pose) using it's sensory information while knowing the internal or external reference frame of the operating environment. On the other hand, mapping is defined as the process of building the robot's map of the environment while knowing it's pose.*

Definition A.4 (Simultaneous Localization and Mapping (SLAM))**.** *It is the process of determining the robot's pose and the map of the environment simultaneously.*

Definition A.5 (Gronwall Inequality)**.** *Basically, Gronwall inequality allows to check the boundness of a function. A function is said to be bounded if it satisfies a certain differential or integral inequality by the solution of the corresponding differential or integral equation.*

Definition A.6 (Weak-star convergence)**.** *Let X be a Banach space and X^* denotes its dual space. A sequence $x_n^* \in X^*$ as $n \to \infty$ is said to converge in the weak star topology to x^*, denoted by*

$$x_n^* \overset{w^*}{\longmapsto} x^*,$$

if, for every $x \in X$,

$$x_n^*(x) \longmapsto x^*(x).$$

Definition A.7 (Alaoglu's Theorem, preliminary form)**.** *If X is a separable normed linear space, then any sequence in X^* that is bounded in norm has a subsequence that converges weak–star against X.*

Definition A.8 (Riesz Representation Theorem for $\mathcal{L}_p$)**.** *Let $(X, \mathcal{A}, \mu)$ be a σ-finite measure space, let $1 \leq p < \infty$, and let q be the dual index of p. If x^* is a continuous linear functional on $\mathcal{L}_p(X, \mu)$, then there exists a unique member g of $\mathcal{L}_q(X, \mu)$ such that*

$$x^*(f) = \int_X fg d\mu, \qquad \forall f \in \mathcal{L}_p.$$

For this function g, $\|x^*\| = \|g\|_q$.

A.2 Proof of Lemma 3.1

Proof. Let the configuration of the left wheel of the DDMR shown in Figure 3.1 be $\mathbf{q}_L = [x_L \;\; y_L \;\; \theta_L]^T$ and that of the right wheel be $\mathbf{q}_R = [x_R \;\; y_R \;\; \theta_R]^T$. Considering each wheel as a unicycle, then using the model (3.1), the nonholonomic constraints acting on each wheel can be written as:

$$\begin{aligned} \dot{x}_L \sin\theta_L - \dot{y}_L \cos\theta_L = 0 \;\; \text{and} \\ \dot{x}_R \sin\theta_R - \dot{y}_R \cos\theta_R = 0. \end{aligned} \tag{A.1}$$

If the wheels themselves can not turn, then the pose of the left and right wheels can be written in terms of the robot coordinates as follows:

$$\begin{aligned} x_L &= x - \frac{l}{2}\sin\theta, \\ y_L &= y + \frac{l}{2}\cos\theta, \\ \theta_L &= \theta, \\ x_R &= x + \frac{l}{2}\sin\theta, \\ y_R &= y - \frac{l}{2}\cos\theta, \\ \theta_R &= \theta. \end{aligned}$$

and their derivatives are

$$
\begin{aligned}
\dot{x}_L &= \dot{x} - \frac{l}{2}\dot{\theta}\cos\theta, \\
\dot{y}_L &= \dot{y} - \frac{l}{2}\dot{\theta}\sin\theta, \\
\dot{\theta}_L &= \dot{\theta}, \\
\dot{x}_R &= \dot{x} + \frac{l}{2}\dot{\theta}\cos\theta, \\
\dot{y}_R &= \dot{y} + \frac{l}{2}\dot{\theta}\sin\theta, \\
\dot{\theta}_R &= \dot{\theta}.
\end{aligned}
\tag{A.2}
$$

Using the nonholonomic constraints on each wheel given in (A.1) and the expressions above, the nonholonomic constraint acting on the left wheel can be written in terms of robot's pose as

$$
\dot{x}_L \sin\theta_L - \dot{y}_L \cos\theta_L = \left(\dot{x} - \frac{l}{2}\dot{\theta}\cos\theta\right)\sin\theta - \left(\dot{y} - \frac{l}{2}\dot{\theta}\sin\theta\right)\cos\theta = \dot{x}\sin\theta - \dot{y}\cos\theta = 0. \tag{A.3}
$$

Similarly, the nonholonomic constraint acting on the right wheel can be written as

$$
\dot{x}_R \sin\theta - \dot{y}_R \cos\theta = \left(\dot{x} - \frac{l}{2}\dot{\theta}\cos\theta\right)\sin\theta - \left(\dot{y} - \frac{l}{2}\dot{\theta}\sin\theta\right)\cos\theta = \dot{x}\sin\theta - \dot{y}\cos\theta = 0. \tag{A.4}
$$

Clearly, the expressions (A.3) and (A.4) are equivalent and, therefore, the robot's nonholonomic constraints acting on the left and right wheels are dependent. Hence, the nonholonomic constraint of a DDMR can be written as

$$
\mathbf{w}(\mathbf{q_L})\dot{\mathbf{q}}_\mathbf{L} = \mathbf{w}(\mathbf{q_R})\dot{\mathbf{q}}_\mathbf{R} = \mathbf{w}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \sin\theta & -\cos\theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0. \tag{A.5}
$$

□

A.3 Proof of Theorem 7.2

Proof. Using the well known Alaoglu's theorem, $\mathcal{K}_{ad} \subset \mathcal{K} \subset \mathbb{R}^{2\times 3}$ is a (weak star) w^o compact set and it suffices to prove that $\mathbf{K} \longmapsto J(\mathbf{K})$ is sequentially weak star continuous. Let $\{\mathbf{K}^i, i \in \mathcal{N}\} \in \mathcal{K}_{ad}$ be a sequence and suppose $\mathbf{K}^i \overset{w^o}{\longmapsto} \mathbf{K}^o$. Since $\mathcal{K}_{ad}$ is w^o closed, we have $\mathbf{K}^o \in \mathcal{K}_{ad}$ [Ahmed and Miah, 2011].

Suppressing the variable t for clarity, let $\{\mathbf{q}^i,\ i \in \mathcal{N}\}$ and $\mathbf{q}^o$ denote the solutions of the system (7.2) corresponding to $\{\mathbf{K}^i,\ i \in \mathcal{N}\}$ and $\mathbf{K}^o$, respectively. Hence, the corresponding state equation becomes

$$\dot{\mathbf{q}}^i = \mathbf{f}(\mathbf{q}^i, \mathbf{K}^i) \text{ and } \dot{\mathbf{q}}^o = \mathbf{f}(\mathbf{q}^o, \mathbf{K}^o),\ \mathbf{q}_0^i = \mathbf{q}_0^o = \mathbf{q}_0.$$

Using (7.4), the solutions of the above two state-space model can be described by

$$\mathbf{q}^i(t) = \mathbf{q}_0 + \int_{t_0}^{t} \mathbf{f}[\mathbf{q}^i(\tau), \mathbf{K}^i(\tau)]d\tau, \text{ and}$$
$$\mathbf{q}^o(t) = \mathbf{q}_0 + \int_{t_0}^{t} \mathbf{f}[\mathbf{q}^o(\tau), \mathbf{K}^o(\tau)]d\tau.$$

Subtracting one from another, we get

$$\mathbf{q}^i(t) - \mathbf{q}^o(t) = \int_{t_0}^{t} \left\{\mathbf{f}[\mathbf{q}^i(\tau), \mathbf{K}^i(\tau)] - \mathbf{f}[\mathbf{q}^o(\tau), \mathbf{K}^o(\tau)]\right\} d\tau. \tag{A.6}$$

Note that $\mathbf{F}(\mathbf{q}^i),\ i \in \mathcal{N}$ and $\mathbf{F}(\mathbf{q}^o)$ are uniformly bounded functions and satisfy Lipschitz condition

$$\|\mathbf{F}(\mathbf{q}^i) - \mathbf{F}(\mathbf{q}^o)\| \leq L_c\|\mathbf{q}^i - \mathbf{q}^o\|,$$

where L_c is the Lipschitz constant. Taking the Euclidean norm in both sides of expression (A.6)

and using the triangle inequality yield

$$\|\mathbf{q}^i(t) - \mathbf{q}^o(t)\| \leq v^i(t) + \int_{t_0}^{t} \beta(\tau)\|\mathbf{q}^i(\tau) - \mathbf{q}^o(\tau)\|d\tau,$$

where

$$v^i(t) = \Big\| \int_{t_0}^{t} \left[\mathbf{F}(\mathbf{q}^i)(\mathbf{K}^i - \mathbf{K}^o)\mathbf{q}^i\right] d\tau \Big\|,$$

$\beta(t) = (c_1(t) + c_2(t)L_c) \in \mathcal{L}_1^+(I)$, with

$$c_1(t) = \sup\{\|\mathbf{F}(\mathbf{q}^i)\|\}][\sup\{\|\mathbf{K}\|\}],\ c_2(t) = \sup\{\|\mathbf{K}\|\|\mathbf{q}\|\},$$

for $i \in \mathcal{N},\ \mathbf{K} \in \mathcal{K},$ and $\mathbf{q} \in \mathcal{Q}$ (see Theorem 8.3.4 of [Ahmed, 2006, p. 273] for more details). Thus, it follows from Gronwall inequality that

$$\| \mathbf{q}^i(t) - \mathbf{q}^o(t) \| \leq v^i(t) + \int_{t_0}^{t} \exp\{\int_{\tau}^{t} \beta(\tau_1)d\tau_1\}\beta(\tau)v^i(\tau)d\tau. \tag{A.7}$$

Clearly, $v^i(t) \to 0$, for $t \in I$, $i \in \mathcal{N}$, as $\mathbf{K}^i \overset{w^o}{\longmapsto} \mathbf{K}^o$. Thus, it follows from inequality (A.7) that $\mathbf{q}^i \overset{\mathbf{K}}{\longmapsto} \mathbf{q}^o$.

Since both $\ell(t,\cdot)$ and $\Phi(t,\cdot)$ are continuous on $\mathbb{R}^3$, we have $\ell[t, \mathbf{q}_i(t)] \longrightarrow \ell[t, \mathbf{q}^o(t)]$ for almost all $t \in I$ and $\Phi[t, \mathbf{q}^i(t_f)] \longrightarrow \Phi[t, \mathbf{q}^o(t_f)]$ as $i \to \infty$. Thus it follows from the expression (6.5) that $\lim_{i\to\infty} J(\mathbf{K}^i) = J(\mathbf{K}^o)$ proving weak star continuity of J on $\mathcal{K}_{ad}$. Since $\mathcal{K}_{ad}$ weak star compact, J attains its minimum on $\mathcal{K}_{ad}$. □

A.4 Proof of Theorem 7.3

Proof. Given the robot's linear kinematic constraint (7.28) and the quadratic cost function (7.31), the perturbational Hamiltonian, $\tilde{\mathcal{H}}$, can be written as

$$\begin{aligned}\tilde{\mathcal{H}}(t) = &\left[\mathbf{F}(t)\Delta\mathbf{q}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t)\right]^T \Delta\boldsymbol{\psi}(t) + \\ &\frac{1}{2}\left[\Delta\mathbf{q}^T(t)\mathbf{Q}(t)\Delta\mathbf{q}(t) + \Delta\mathbf{u}^T(t)\mathbf{R}(t)\Delta\mathbf{u}(t)\right],\end{aligned} \tag{A.8}$$

where $\Delta\boldsymbol{\psi}(t)$ is the costate (adjoint) variable perturbation from the nominal solution $\boldsymbol{\psi}^o(t)$. Therefore, the Euler-Lagrange equations [Stengel, 1994] are

$$\Delta\dot{\boldsymbol{\psi}} = -\mathbf{F}^T(t)\Delta\boldsymbol{\psi}(t) - \mathbf{Q}(t)\Delta\mathbf{q}(t) \tag{A.9}$$

$$\Delta\boldsymbol{\psi}(t_f) = \mathbf{P}(t_f)\Delta\mathbf{q}(t_f) = \Delta\boldsymbol{\psi}_f \tag{A.10}$$

$$\frac{\partial\tilde{\mathcal{H}}(t)}{\partial\Delta\mathbf{u}} = \mathbf{G}^T(t)\Delta\boldsymbol{\psi}(t) + \mathbf{R}(t)\Delta\mathbf{u}(t) = 0. \tag{A.11}$$

The perturbed control can be solved by rearranging (A.11) as

$$\Delta\mathbf{u}(t) = -\mathbf{R}^{-1}(t)\mathbf{G}^T\Delta\boldsymbol{\psi}(t). \tag{A.12}$$

Substituting (A.12) in (7.28) and (A.9) yield the linear, two-point boundary value problem as

$$\Delta\dot{\mathbf{q}} = \mathbf{F}(t)\Delta\mathbf{q}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{G}^T(t)\Delta\boldsymbol{\psi}(t),\ \Delta\mathbf{q}(t_0) = \Delta\mathbf{q}_0 \text{ and} \tag{A.13}$$

$$\Delta\dot{\boldsymbol{\psi}} = -\mathbf{F}^T(t)\Delta\boldsymbol{\psi}(t) - \mathbf{Q}(t)\Delta\mathbf{q}(t),\ \Delta\boldsymbol{\psi}(t_f) = \Delta\boldsymbol{\psi}_f. \tag{A.14}$$

Since $\Delta\mathbf{q}(t)$ and $\Delta\boldsymbol{\psi}(t)$ are adjoint, model (A.10) applies not only at the final time but during the entire time interval I:

$$\Delta\boldsymbol{\psi}(t) = \mathbf{P}(t)\Delta\mathbf{q}(t),\ \ \forall\, t \in I. \tag{A.15}$$

The expression for the perturb control (A.12) can now be written as the linear quadratic *optimal feedback control law*,

$$\begin{aligned}\Delta\mathbf{u}(t) &= -\mathbf{R}^{-1}(t)\mathbf{G}^T(t)\mathbf{P}(t)\Delta\mathbf{q}(t) \\ &= -\mathbf{C}(t)\Delta\mathbf{q}(t),\end{aligned} \tag{A.16}$$

where

$$\mathbf{C}(t) = \mathbf{R}^{-1}(t)\mathbf{G}^T(t)\mathbf{P}(t). \tag{A.17}$$

Note that model (A.16) is exactly (7.32) as stated in the theorem. The model (7.32) describes a linear-qudratic (LQ) control law, so called because the kinematic constraint (7.28) is linear and the cost function (7.31) is quadratic. A schematic of the optimal linear-quadratic control law is shown in Figure A.1.

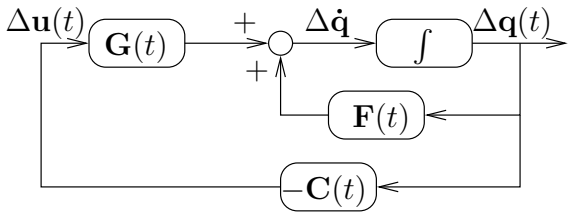


Figure A.1: Schematic of the robot's linear-quadratic control law in continuous time.

Note that the 3×3 matrix $\mathbf{P}(t)$ in (7.32) remains to be found. Differentiating Equation (A.15) incorporating the Equations (A.13), (A.14), and (A.15), a nonlinear differential equation for $\mathbf{P}(t)$ can be derived. Dropping the argument (t) for clarity,

$$\begin{aligned}\dot{\mathbf{P}}\Delta\mathbf{q} &= \Delta\dot{\boldsymbol{\psi}} - \mathbf{P}\Delta\dot{\mathbf{q}} \\ &= -\mathbf{F}^T\Delta\boldsymbol{\psi} - \mathbf{Q}\Delta\mathbf{q} - \mathbf{P}\left[\mathbf{F}\Delta\mathbf{q} - \mathbf{G}\mathbf{R}^{-1}\mathbf{G}^T\Delta\boldsymbol{\psi}\right] \\ &= -\mathbf{F}^T\mathbf{P}\Delta\mathbf{q} - \mathbf{Q}\Delta\mathbf{q} - \mathbf{P}\left[\mathbf{F}\Delta\mathbf{q} - \mathbf{G}\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}\Delta\mathbf{q}\right] \\ \dot{\mathbf{P}} &= -\mathbf{F}^T\mathbf{P} - \mathbf{Q} - \mathbf{P}\mathbf{F} + \mathbf{G}\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P},\end{aligned}$$

which is precisely the differential matrix Riccati equation given in (7.33). □

A.5 Proof of Lemma 7.2

Proof. Clearly, from (7.36), $V[t, \mathbf{0}] = 0$, and $\frac{\partial V}{\partial \Delta \mathbf{q}}[t, \Delta\mathbf{q}(t)] = \mathbf{P}(t)\Delta\mathbf{q}(t) \in C^1(I, \mathbb{R}^3)$. Using (A.15), let us define

$$\Delta\phi(t) = \Delta\mathbf{q}^T(t)\mathbf{P}(t)\Delta\mathbf{q}(t) = \Delta\mathbf{q}^T\Delta\boldsymbol{\psi}(t), \quad t \in I. \tag{A.18}$$

From (A.10), it is known that

$$\Delta\phi(t_f) = \Delta\mathbf{q}^T(t_f)\Delta\boldsymbol{\psi}(t_f) = \Delta\mathbf{q}^T(t_f)\mathbf{P}(t_f)\Delta\mathbf{q}(t_f) \geq 0.$$

Differentiating (A.18) with respect to t and dropping the variable (t) for clarity, we get

$$\begin{aligned}
\Delta\dot{\phi} &= \Delta\dot{\mathbf{q}}^T\Delta\boldsymbol{\psi} + \Delta\mathbf{q}^T\Delta\dot{\boldsymbol{\psi}} \\
&= -\Delta\mathbf{q}^T\mathbf{Q}\Delta\mathbf{q} - (\mathbf{G}^T\mathbf{P}\Delta\mathbf{q})^T\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}\Delta\mathbf{q} \\
&= -\Delta\mathbf{q}^T\mathbf{Q}\Delta\mathbf{q} - (\mathbf{R}\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}\Delta\mathbf{q})^T\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}\Delta\mathbf{q} \\
&= -\Delta\mathbf{q}^T\mathbf{Q}\Delta\mathbf{q} - (\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}\Delta\mathbf{q})^T\mathbf{R}\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}\Delta\mathbf{q} \\
&= -\Delta\mathbf{q}^T\mathbf{Q}\Delta\mathbf{q} - \Delta\mathbf{u}^T\mathbf{R}\Delta\mathbf{u} \\
&\leq 0.
\end{aligned} \tag{A.19}$$

Now integrating and using the above expression, we find that

$$\Delta\phi(t) = \Delta\phi(t_f) - \int_t^{t_f} \dot{\phi}(\tau)d\tau = \Delta\mathbf{q}^T(t_f)\mathbf{P}(t_f)\Delta\mathbf{q}(t_f) + \int_t^{t_f} [\Delta\mathbf{q}^T(\tau)\mathbf{Q}(\tau)\Delta\mathbf{q}(\tau) + \Delta\mathbf{u}^T(\tau)\mathbf{R}(\tau)\Delta\mathbf{u}(\tau)]d\tau \geq 0, \quad \forall\, t \in I,$$

since $\mathbf{P}(t) \geq \mathbf{0}$, $\mathbf{Q}(t) \geq \mathbf{0}$, $\mathbf{R}(t) > \mathbf{0}$, and all are symmetric matrices. Hence, the solution of the differential matrix Riccati equation (7.33) has to be real, symmetric, and at least positive semi-definite matrix for $V[t, \Delta\mathbf{q}(t)] = (1/2)\Delta\phi(t) \geq 0$. Thus, $V[t, \Delta\mathbf{q}(t)]$ in (7.36) satisfies the Lyapunov basic properties. □

A.6 Proof of Lemma 7.3

Proof. Consider $\Delta\mathbf{q}(t) \equiv \Delta\mathbf{q}(t, \Delta\mathbf{q}_0)$ is the solution of the feedback system (7.35). Since $\mathbf{Q}(t)$ is a symmetric positive semi-definite matrix, then obviously, for any $\Delta\mathbf{q}(t) \in \mathbb{R}^3$

$$
\begin{aligned}
&\int_{t_0}^{t_f} \Delta\mathbf{q}^T(t)\mathbf{Q}(t)\Delta\mathbf{q}(t)dt \geq 0 \\
\Rightarrow &\int_{t_0}^{t_f} \Delta\mathbf{q}^T(t, \Delta\mathbf{q}_0)\mathbf{D}^T(t)\mathbf{D}(t)\Delta\mathbf{q}(t, \Delta\mathbf{q}_0)dt \geq 0 \\
\Rightarrow &\int_{t_0}^{t_f} \Delta\mathbf{q}^T(t_0)\mathbf{\Phi}^T(t, t_0)\mathbf{D}^T(t)\mathbf{D}(t)\mathbf{\Phi}(t, t_0)\Delta\mathbf{q}(t_0)dt \geq 0 \\
\Rightarrow &\Delta\mathbf{q}^T(t_0)\mathbb{M}(t_f, t_0)\Delta\mathbf{q}(t_0) \geq 0.
\end{aligned}
$$

It clear from the above expression that $\mathbb{M}(t_f, t_0)$ given by (7.37) must be nonsingular. □

Bibliography

Yasunori Abe, Masaru Shikano, Toshio Fukuda, Fumihito Arai, and Yoshio Tanaka. Vision based navigation system for autonomous mobile robot with global matching. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1299–1304, 1999.

Amit Adam, Ehud Rivlin, and Hector Rotstein. Fusion of fixation and odometry for vehicle navigation. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 29(6):593–603, November 1999.

A. Pedro Aguiar and Joo P. Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE Transactions on Automatic Control*, 52(8):1362–1379, August 2007.

N. U. Ahmed and M. Suruz Miah. Optimal feedback control law for a class of partially observed uncertain dynamic systems: A min-max problem. *Dynamic Systems and Applications*, 20(1): 149–167, 2011.

Nasir Uddin Ahmed. *Elements of Finite-Dimensional Systems and Control Theory*. Longman Scientific & Technical, John Wiley & Sons, Inc., Essex, England, 1988.

Nasir Uddin Ahmed. *Dynamic Systems and Control with Applications*. World Scientific, New Jersey, 2006.

M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino. Closed loop steering of unicycle like vehicles via lyapunov techniques. *Robotics Automation Magazine, IEEE*, 2(1):27 –35, mar 1995.

F. Allgower, T. A Badgwell, S. J Qin, J. B Rawlings, and S. J Wright. Nonlinear predictive control and moving horizon estimation-an introductory overview. In P. M. Frank, editor, *Advances in Control, Hightlights of ECC'99*, pages 391–449. New York: Springer, 1999.

Bruno Apolloni and Adrian Moise. Mobile robot navigation using the mathematical model of the sensing system. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 2625–2630, 2000.

Brenna D. Argall. *Learning Mobile Robot Motion Control From Demonstration and Corrective Feedback*. PhD thesis, Robotics Institute Carnegie Mellon University, Pittsburgh, PA 15213, March 2009.

Constantine A. Balanis. *Antenna Theory: Analysis and Design*. Wiley, Socorro, New Maxico, 3rd edition, 2005.

Abdelmoula Bekkali and Mitsuji Matsumoto. RFID indoor tracking system based on inter-tags distance measurements. In *Wireless Technology*, volume 44 of *Lecture Notes in Electrical Engineering*, pages 41–62. Springer US, 2009.

Stephane Betge-Brezetz, Raja Chatila, and Michel Devy. Natural scene understanding for mobile robot navigation. In *IEEE International Conference on Robotics and Automation*, pages 730–736, 1994.

J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. Wellesley, 1996.

J. Borenstein, H. R. Everett, L. Feng, and D. Wehe. Mobile robot positioning: sensors and techniques. *Journal of Robotic Systems*, 14(4):231–49, April 1997. ISSN 0741-2223.

R. W. Brockett. Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory*, pages 181–191, 1983.

Zhengcai Cao, Yingtao Zhao, and Shuguo Wang. Trajectory tracking and point stabilization of noholonomic mobile robot. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1328 –1333, oct. 2010.

J. Castellanos, J. Montiel, J. Neira, and J. Tardos. The spmap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–953, 1999.

I. Cervantes and J. Alvarez-Ramirez. On the PID tracking control of robot manipulators. *Systems and Control Letters*, 42(1):37–46, January 2001.

Hong Chen, Miao-Miao Ma, Hu Wang, Zhi-Yuan Liu, and Zi-Xing Cai. Moving horizon $\mathcal{H}_\infty$ tracking control of wheeled mobile robots with actuator saturation. *IEEE Transactions on Control Systems Technology*, 17(2):449–57, March 2009.

Zhichao Chen and Stanley T. Birchfield. Qualitative vision-based mobile robot navigation. In *IEEE International Conference on Robotics and Automation*, 2006.

Byoung-Suk Choi, Joon-Woo Lee, Ju-Jang Lee, and Kyoung-Taik Park. A hierarchical algorithm for indoor mobile robot localization using RFID sensor fusion. *IEEE Transactions on Industrial Electronics*, 58(6):2226–2235, 2011.

Kyung-Sik Choi and Suk-Gyu Lee. Enhanced SLAM for a mobile robot using extended kalman filter and neural networks. *International Journal of Precision Engineering and Manufacturing*, 11(2):255–264, April 2010.

Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastain Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. The MIT Press, Cambridge, Massachusetts, London, England, 2004.

Dongkyoung Chwa. Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. *IEEE Transactions on Control Systems Technology*, 12(4):637–644, July 2004.

Dongkyoung Chwa. Tracking control of differential-drive wheeled mobile robots using a backstepping-like feedback linearization. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(6):1285–1295, November 2010.

J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, Boston, 2nd edition, 1989.

Tamoghna Das, I.N. Kar, and S. Chaudhury. Simple neuron-based adaptive controller for a nonholonomic mobile robot including actuator dynamics. *Neurocomputing*, 69(16–18): 2140–2151, 2006.

S.L. de Oliveira Kothare and M. Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 45(6):1053–1071, jun 2000.

C. Canudas de Wit, C. Samson, O. J. Srdalen, and Yuan F Zheng (Editor). *Nonlinear control design for mobile robots*, volume 11. World Scientific, Singapore, 1993.

Guilherme N. DeSouza and Avinash C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, February 2002.

E. Di Giampaolo and F. Martinelli. A passive UHF-RFID system for the localization of an

indoor autonomous vehicle. *IEEE Transactions on Industrial Electronics*, 2012. (early access).

M. Diehl, H. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear mpc and moving horizon estimation. In D. Raimondo L. Magni and F. Allgower, editors, *Nonlinear Model Predictive Control-Towards New Challenging Applications*, pages 391–417. New York: Springer, 2000.

M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–41, June 2001.

D.M. Dobkin and S.M. Weigand. Environmental effects on rfid tag antennas. In *IEEE MTT-S International Microwave Symposium Digest*, page 4 pp., june 2005.

T. D'Orazio, M. Ianigro, E. Stella, F. P. Lovergine, and A. Distante. Mobile robot navigation by multi-sensory integration. In *IEEE International Conference on Robotics and Automation, ICRA*, pages 373–9, Atlanta, GA, USA, 1993.

K. C. Drake, E. S. McVey, and R. M. Inigo. Sensor roll angle error for a mobile robot using a navigation line. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5): 727–731, September 1988.

P. Encarnacao and A. Pascoal. Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 1, pages 964 –969 vol.1, 2001.

Klaus Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. Wiley, 2 edition, May 2003.

F. A. C. C. Fontes. Discontinuous feedbacks, discontinuous optimal controls, and continuous-time model predictive control. *International Journal of Robust and Nonlinear Control*, 13(3): 191–209, March 2003.

Chaitanya Gharpure, Vladimir Kulyukin, and Aliasgar Kutiyanawala. A robotic shopping assistant for the blind: A pilot study in a supermarket. Technical report usu-csatl-1-01-06, Utah State University, Computer Science Assistive Technology Laboratory, Department of Computer Science, Utah State University, January 2006.

J. Graefenstein and M. E. Bouzouraa. Robust method for outdoor localization of a mobile robot using received signal strength in low power wireless networks. In *IEEE International Conference on Robotics and Automation*, pages 33–8, Pasadena, CA, USA, 2008.

K. Graichen and A. Kugi. Stability and incremental improvement of suboptimal mpc without terminal constraints. *IEEE Transactions on Automatic Control*, 55(11):2576–2580, nov. 2010.

Dongbing Gu and Huosheng Hu. Receding horizon tracking control of wheeled mobile robots. *IEEE Transactions on Control Systems Technology*, 14(4):743–749, july 2006.

Dirk Hahnel, Wolfram Burgard, Dieter Fox, Ken Fishkin, and Matthai Philipose. Mapping and localization with RFID technology. In *IEEE International Conference on Robotics and Automation*, pages 1015–1020, New Orleans, LA, United States, April 2004.

Ireneusz Hallmann and Barbara Siemiatkowska. Artificial landmark navigation system. In *International Symposium on Intelligent Robotic Systems*, July 2001.

Soonshin Han, HyungSoo Lim, and JangMyung Lee. An efficient localization scheme for a differential-driving mobile robot based on rfid system. *IEEE Transactions on Industrial Electronics*, 54(6):3362–3369, dec. 2007.

Bo He, Danwei Wang, Minhtuan Pham, and Tieniu Yu. Gps/encoder based precise navigation for a 4ws mobile robot. In *7th International Conference on Control, Automation, Robotics and Vision*, Singapore, December 2002.

Martial Hebert and Takeo Kanade. 3-d vision for outdoor navigation by an autonomous vehicle. In *Proceedings of the 1988 DARPA Image Understanding Workshop*, pages 593–601, April 1988.

Chae Heesung and Han Kyuseo. Combination of rfid and vision for mobile robot localization. In *Intelligent Sensors, Sensor Networks and Information Processing Conference*, volume 2005, 2005.

Harry Chia-Hung Hsu and A. Liu. A flexible architecture for navigation control of a mobile robot. *IEEE Transactions on Systems, Man & Cybernetics, Part A*, 37(3):310–18, May 2007.

Huosheng Hu and Dongbing Gu. Landmark-based navigation of mobile robots in manufacturing. In *IEEE International Conference on Emerging Technologies and Factory Automation*, Barcelona, Spain, October 1999.

Seo-Yeon Hwang and Jae-Bok Song. Monocular vision-based slam in indoor environment using corner, lamp, and door features from upward-looking camera. *IEEE Transactions on Industrial Electronics*, 58(10):4804 –4812, oct. 2011.

F. Janabi-Sharifi and M. Marey. A kalman-filter-based method for pose estimation in visual servoing. *IEEE Transactions on Robotics*, 26(5):939–947, October 2010.

Z. P. Jiang, E. Lefeber, and H. Nijmeijer. Saturated stabilization and tracking of a nonholonomic mobile robot. *Systems & Control Letters*, 42(5):327–32, April 2001.

Zhong-Ping Jiang and H. Nijmeijer. Tracking control of mobile robots: a case study in backstepping. *Automatica*, 33(7):1393–9, July 1997.

Zhong-Ping Jiang and H. Nijmeijer. A recursive technique for tracking control of nonholonomic systems in chained form. *IEEE Transactions on Automatic Control*, 44(2):265 –279, feb 1999.

Liu Jing and Po Yang. A localization algorithm for mobile robots in rfid system. In *International Conference on Wireless Communications, Networking and Mobile Computing*, pages 2109–2112, sept. 2007.

K.G. Jolly, R.S. Kumar, and R. Vijayakumar. An artificial neural network based dynamic controller for a robot in a multi-agent system. *Neurocomputing*, 73(1-3):283–294, 2009.

Thomas Kampke, Boris Kluge, and Matthias Strobel. Exploiting RFID capabilities onboard a service robot platform. *Springer Tracts in Advanced Robotics*, 76(1):215–225, April 2012.

Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. In *Proceedings-IEEE International Conference on Robotics and Automation*, pages 384–9, Cincinnati, OH, USA, 1990.

G. Kantor and S. Singh. Preliminary results in range-only localization and mapping. In *Proceedings 2002 IEEE ICRA*, pages 1818–23, Washington, DC, USA, 2002.

Fakhreddine Karray and Clarence W. de Silva. *Soft Computing and Intelligent Systems Design, Theory, Tools and Applications.* Addison-Wesley, Pearson Education Limited,, Essex, England, 4th edition, 2004.

S. S. Keerthi and E. G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(2):265–93, May 1988. ISSN 0022-3239.

Hassan K. Khalil. *Nonlinear Systems*. Addison-Wesley, Pearson Education Limited,, Upper Saddle River, New Jersey, 3rd edition, 2002.

Chang-Sei Kim, Keum-Shik Hong, and Moon-Ki Kim. Nonlinear robust control of a hydraulic elevator: experiment-based modeling and two-stage lyapunov redesign. *Control Engineering Practice*, 13(6):789–803, June 2005.

Myungsik Kim and Nak Young Chong. Direction sensing RFID reader for mobile robot navigation. *IEEE Transactions on Automation Science and Engineering*, 6(1):44–54, January 2009.

Anthony W. Knapp. *Real Analysis*. Birkhauser, Boston, Basel, Berlin, 2005.

K. Kodaka, H. Niwa, Y. Sakamoto, M. Otake, Y. Kanemori, and S. Sugano. Pose estimation of a mobile robot on a lattice of rfid tags. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1385–1390, sept. 2008.

S. Kotani, K. Kaneko, T. Shinoda, and H. Mori. Mobile robot navigation based on vision and dgps information. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2524–2529, 1998.

Olaf Kubitz, Matthias O. Berger, Marcus Perlick, and Rene Dumoulin. Application of radio frequency identification devices to support navigation of autonomous mobile robots. In *IEEE Vehicular Technology Conference*, pages 126–130, Phoenix, AZ, USA, 1997.

V. Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran. RFID in robot-assisted indoor navigation for the visually impaired. In *2004 IEEE/RSJ IROS*, pages 1979–84, Sendai, Japan, 2004.

Derek Kurth, George Kantor, and Sanjiv Singh. Experimental Results in Range-only Localization with Radio. In *IEEE International Conference on Intelligent Robots and Systems*, volume 1, pages 974–979, Las Vegas, NV, United States, October 2003.

Michail G. Lagoudakis and Anthony S. Maida. Neural maps for mobile robot navigation. In *Proceedings of the International Joint Conference on Neural Networks*, volume 3, pages 2011–2016, 1999.

Steven Lanzisera, David T. Lin, and Kristofer S. J. Pister. RF time of flight ranging for wireless sensor network localization. In *Proceedings of the Fourth Workshop on Intelligent Solutions in Embedded Systems, WISES 2006*, pages 165–176, Vienna, Austria, June 2006.

Jun Ho Lee, Cong Lin, Hoon Lim, and Jang Myung Lee. Sliding mode control for trajectory tracking of mobile robot in the RFID sensor space. *International Journal of Control, Automation and System*, 7(3):429–435, June 2009.

Ti-Chung Lee, Kai-Tai Song, Ching-Hung Lee, and Ching-Cheng Teng. Tracking control of unicycle-modeled mobile robots using a saturation feedback controller. *IEEE Transactions on Control Systems Technology*, 9(2):305–18, March 2001.

Tsong-Li Lee, Li-Chun Lai, and Chia-Ju Wu. A fuzzy algorithm for navigation of mobile robots in unknown environments. In *IEEE International Symposium on Circuits and Systems*, volume 4, pages 3039– 3042, May 2005.

Yu-Cheol Lee, C. Christiand, Wonpil Yu, and Sunghoon Kim. Urban localization method for mobile robots based on dead reckoning sensors, GPS, and map matching. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2363–2368, oct. 2011.

John J. Leonard and Hugh F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.

T. H. S. Li and Shih-Jie Chang. Autonomous fuzzy parking control of a car-like mobile robot. *IEEE Transactions on Systems, Man & Cybernetics, Part A*, 33(4):451–65, July 2003.

Tzu Hao Li. Open Platform Semi-Passive UHF RFID Sensor Tag. Master's thesis, University of Ottawa, Ottawa, Ontario, Canada, 2011.

M.Ni Lionel, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil. LANDMARC: Indoor location sensing using active RFID. *Wireless Networks*, 10(6):701–710, November 2004.

Hsin-Chin Liu, Meng-Chang Hua, Chih-Guo Peng, and Jhen-Peng Ciou. A novel battery-assisted class-1 generation-2 rf identification tag design. *IEEE Transactions on Microwave Theory and Techniques*, 57(5):1388–1397, may 2009.

Ran Liu, P. Vorst, A. Koch, and A. Zell. Path following for indoor robots with RFID received signal strength. In *19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–7, sep 2011.

David A. Lizarrage. Obstructions to the existence of universal stabilizers for smooth control systems. *Mathematics of Control, Signals, and Systems*, 16(4):255–277, June 2004.

A. De Luca, G. Oriolo, and C. Samson. Feedback control of a nonholonomic car-like robot. In Jean-Paul Laumond, editor, *Robot Motion Planning and Control*, volume 229 of *Lecture Notes in Control and Information Sciences*, chapter 4, pages 170–253. Springer, 2000.

R.C. Luo, Chi-Tao Chuang, and Sung-Sheng Huang. Rfid-based indoor antenna localization system using passive tag and variable rf-attenuation. In *IEEE International Conference on Industrial Electronics Society*, pages 2254–2259, nov. 2007.

Wencheng Luo, Yun-Chung Chu, and Keck-Voon Ling. Inverse optimal adaptive control for attitude tracking of spacecraft. *IEEE Transactions on Automatic Control*, 50(11):1639 – 1654, nov. 2005.

Hannu Makela and Kari Koskinen. Navigation of outdoor mobile robots using dead reckoning

and visually detected landmarks. In *Fifth International Conference on Advanced Robotics*, pages 1051–6, Pisa, Italy, 1991.

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36(6):789–814, June 2000.

E. S. McVey, K. C. Drake, and R. M. Inigo. Range measurements by a mobile robot using a navigation line. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1): 105–109, January 1986.

Qinghao Meng and Rainer Bischoff. Odometry based pose determination and errors measurement for a mobile robot with two steerable drive wheels. *Journal of Intelligent and Robotic Systems archive*, 41(4):263–282, January 2005.

B.A. Merhy, P. Payeur, and E. M. Petriu. Application of segmented 2D probabilistic occupancy maps for mobile robot sensing and navigation. In *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, pages 2342 –2347, april 2006.

B.A. Merhy, P. Payeur, and E.M. Petriu. Application of segmented 2-d probabilistic occupancy maps for robot sensing and navigation. *IEEE Transactions on Instrumentation and Measurement*, 57(12):2827–2837, dec 2008.

M. Suruz Miah. Autonomous mobile robot navigation using rfid technology. Master's thesis, Department of Electrical and Computer Engineering, University of Ottawa, Ottawa, Ontario, Canada, October 2007.

M. Suruz Miah and Wail Gueaieb. Towards a computationally efficient relative positioning system for indoor environments: An RFID approach. In *ICINCO*, Milan, Italy, 2009a.

M. Suruz Miah and Wail Gueaieb. A stochastic approach of mobile robot navigation using

customized rfid systems. In *International Conference on Signals, Circuits and Systems*, Jerba, Tunisia, November 2009b.

M. Suruz Miah and Wail Gueaieb. Indoor robot navigation through intelligent processing of RFID signal measurements. In *Proceedings of the International Conference on Autonomous and Intelligent Systems*, Povoa de Varzim, Portugal, June 2010a.

M. Suruz Miah and Wail Gueaieb. On the implementation of an efficient mobile robot navigation system: An RFID approach. In *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS 2010)*, Ottawa, Canada, August 2010b.

M. Suruz Miah and Wail Gueaieb. A fuzzy logic approach for indoor mobile robot navigation using ukf and customized rfid communication. In *Proceedings of the International Conference on Autonomous and Intelligent Systems*, Burnaby, BC, Canada, June 2011a.

M. Suruz Miah and Wail Gueaieb. Mobile robot navigation using direction sensitive RFID reader. *International Journal of Control and Intelligent Systems*, 39(3):1–11, 2011b.

M. Suruz Miah and Wail Gueaieb. Mobile robot trajectory tracking using noisy RSS measurements: An RFID approach. *IEEE Transactions on Industrial Electronics*, 2012a. (conditionally accepted).

M. Suruz Miah and Wail Gueaieb. Optimal time-varying p-controller for a class of uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 2012b. (status:revised and resubmitted).

M. Suruz Miah and Wail Gueaieb. Rfid-based mobile robot trajectory tracking and point stabilization through on-line neighboring optimal control. *IEEE Transactions on Control Systems Technology*, 2012c. (submitted).

M. Suruz Miah and Wail Gueaieb. Experiments on optimal feedback control for stabilization and tracking of mobile robots. *IEEE Transactions on Robotics*, 2012d. (submitted).

M. Michalek and K. Kozowski. Vector-field-orientation feedback control method for a differentially driven vehicle. *IEEE Transactions on Control Systems Technology*, 18(1):45–65, jan 2010.

Annalisa Milella, Paolo Vanadia, Grazia Cicirelli, and Arcangelo Distante. RFID-based environment mapping for autonomous mobile robot applications. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, Zurich, Switzerland, September 2007.

Omid Mohareri, Rached Dhaouadi, and Ahmad B. Rad. Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks. *Neurocomputing*, 88:54 – 66, 2012.

Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *18th National Conference on Artificial Intelligence (AAAI-02), 14th Innovative Applications of Artificial Intelligence Conference (IAAI-02)*, pages 593–598, Edmonton, Alta., Canada, 2002.

Pascal Morin and Claude Samson. Control of nonholonomic mobile robots based on the transverse function approach. *IEEE Transactions on Robotics*, 25(5):1058–73, October 2009.

Don Murray and James J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2):161–171, April 2000.

Nejah Nasri, Nacer Kachouri, Mounir Samet, and Laurent Andrieux. Radio frequency identification (RFID): Working, design considerations and modelling of antenna. In *International Multi-Conference on Systems, Signals and Devices*, Amman, Jordan, July 2008.

L. M. Ni, Yunhao Liu, Yiu Cho Lau, and A. P. Patil. LANDMARC: indoor location sensing using active RFID. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, pages 407–15, TX, USA, 2003.

Kazunori Ohno, Takashi Tsubouchit, Bunji Shigematsut, Shoichi Maeyamas, and Shin'ichi Yuta. Outdoor navigation of a mobile robot between buildings based on DGPS and odometry data fusion. In *IEEE International Conference on Robotics and Automation. ICRA '03.*, volume 2, pages 1978–1984, September 2003.

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

Lauro Ojeda, Daniel Cruz, Giulio Reina, and Johann Borenstein. Current-based slippage detection and odometry correction for mobile robots and planetary rovers. *IEEE Transactions on Robotics*, 22(2):366–378, April 2006.

Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli. Wmr control via dynamic feedback linearizeation: Design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, November 2002.

Dayal R. Parhi. Navigation of mobile robots using a fuzzy logic controller. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 42(3):253–273, March 2005.

Bong Seok Park, Sung Jin Yoo, Jin Bae Park, and Yoon Ho Choi. A simple adaptive control approach for trajectory tracking of electrically driven nonholonomic mobile robots. *IEEE Transactions on Control Systems Technology*, 18(5):1199 –1206, sept. 2010.

KyuCheol Park, Dohyoung Chung, Hakyoung Chung, and Jang Gyu Lee. Dead reckoning

navigation of a mobile robot using an indirect kalman filter. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 132–138, 1996.

KyuCheol Park, Hakyoung Chung, Jongbin Choi, and Jang Gyu Lee. Dead reckoning navigation for an autonomous mobile robot using a differential encoder and a gyroscope. In *8th International Conference on Advanced Robotics, ICAR '97*, pages 441–446, July 1997.

Sunhong Park and Shuji Hashimoto. Autonomous mobile robot navigation using passive RFID in indoor environment. *IEEE Transactions on Industrial Electronics*, 56(7):2366–2373, 2009.

P. Payeur. Dealing with uncertain measurements in virtual representations for robot guidance. In *IEEE International Symposium on Virtual and Intelligent Measurement Systems*, pages 56–61, 2002.

Shi Peng and Wang Dong. Robot navigation system with RFID and sensors. In *International Conference on Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM)*, pages 610–612, march 2012.

P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. RFID systems: a survey on security threats and proposed solutions. In *International Conference Personal Wireless Communications*, pages 159–70, Albacete, Spain, 2006.

Liliane Peters, Michael Pauly, and Klaus Beck. Servicebot–mobile robots in co-operative environments. In *ERCIM News*, July 2000.

James Lyle Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall, N.J. Englewood Cliff, 1981.

D.T. Pham and Dayal R. Parhi. Navigation of multiple mobile robots using a neural networks and a petri net model. *Robotica*, 21(1):79–93, January/February 2003.

Jean-Baptiste Pomet. Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. *Systems & Control Letters*, 18(2):147–158, February 1992.

Rajesh Rajamani. Observers for lipschitz nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3):397–401, mar 1998.

K. V. S. Rao. An overview of backscattered radio frequency identification system (RFID). In *Asia Pacific Microwave Conference*, pages 746–9, Singapore, 1999.

J. Rawlings and D. Mayne. *Model Predictive Control: Theory and Design*. WI: Nob Hill Publishing, Madison, 2009.

Muhammad Rehan, Keum-Shik Hong, and Shuzhi Sam Ge. Stabilization and tracking control for a class of nonlinear systems. *Nonlinear Analysis: Real World Applications*, 12(3): 1786–1796, 2011.

Ioannis Rekleitis. *Cooperative Localization and Multi-robot Exploration*. PhD thesis, School of Computer Science, McGill University, Montreal, Quebec, Canada, January 2003.

H. L. Royden and Patrick M. Fitzpatrick. *Real Analysis*. Pearson Education Inc., Boston, 4th edition, 2010.

Petru Rusu, Emil M. Petriu, Thom E. Whalen, Aurel Cornell, and Hans J.W. Spoelder. Behavior-based neuro-fuzzy controller for mobile robot navigation. *IEEE Transactions on Instrumentation and Measurement*, 52(4):1335–1340, August 2003.

Alessandro Saffiotti. The uses of fuzzy logic for autonomous robot navigation: a catalogue raisonn'e. *Soft Computing Research journal*, 1(4):180–197, 1997.

L. Sciavicco and B. Siciliano. *Modeling and Control of Robot Manipulators*. McGraw-Hill, New York, 1996.

Reza Shahidi, Mark Shayman, and P. S. Krishnaprasad. Mobile robot navigation using potential functions. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2047–2053, 1991.

Li Sheng, Ma Guoliang, and Hu Weili. Stabilization and optimal control of nonholonomic mobile robot. In *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, volume 2, pages 1427 – 1430 Vol. 2, dec. 2004.

C. Siagian and L. Itti. Biologically inspired mobile robot vision localization. *IEEE Transactions on Robotics*, 25(4):861 –873, August 2009.

Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. The MIT Press, Cambridge, Massachusetts, London, England, 2002.

R. Smith, M. Self, and P. Cheeseman. *Estimating Uncertain Spatial Relationships in Robotics*. In I. Cox and G. Wilfong, editors, Autonomous Robot Vehicles, Springer Verlag, Menlo Park, California, 1990.

E. Stella, F.P. Lovergine, L. Caponetti, and A. Distante. Mobile robot navigation using vision and odometry. In *Proceedings of the Intelligent Vehicles '94 Symposium*, pages 417–422, October 1994.

Robert F. Stengel. *Optimal Control and Estimation*. Dover publications, inc., New York, 1994.

Kuo-Ho Su, Yih-Young Chen, and Shun-Feng Su. Design of neural-fuzzy-based controller for two autonomously driven wheeled robot. *Neurocomputing*, 73(13-15):2478 – 88, 2010/08/.

Dong Sun, Can Wang, Wen Shang, and Gang Feng. A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations. *IEEE Transactions on Robotics*, 25(5):1074–1086, October 2009.

Jian Sun and G.P. Liu. State feedback and output feedback control of a class of nonlinear systems with delayed measurements. *Nonlinear Analysis*, 67(5):1623–1636, September 2007.

C. Tarin, B. Tibken, H. Brugger, and E.P. Hofer. Optimal feedback position control for an autonomous mobile robot. In *American Control Conference, 2000. Proceedings of the 2000*, volume 3, pages 1491 –1492 vol.3, 2000.

Chuck Thorpe, Martial Hebert, Takeo Kanade, and Steven Shafer. Vision and navigation for the carnegie-mellon navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):362–373, 1988.

Sebastain Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, London, England, 2005.

Toshifumi Tsukiyama. Global navigation system with RFID tags. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4573, pages 256–264, 2001.

Toshifumi Tsukiyama. World map based on rfid tags for indoor mobile robots. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 6006, 2005.

Sada Wane and Cina Motamed. Navigation system for a mobile robot in an unstructured environment. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2797–2802, 1994.

Du Qu Wei, Xiao Shu Luo, Bo Zhang, and Ying Hua Qin. Controlling chaos in space-clamped fitzhughnagumo neuron by adaptive passive method. *Nonlinear Analysis: Real World Applications*, 11(3):1752–1759, June 2010.

O. Wijk and H.I. Christensen. Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data. *Robotics and Autonomous System*, 31(1-2):31–42, April 2000.

Lingfei Wu, M.Q.-H. Meng, Zijing Lin, Wu He, Chao Peng, and Huawei Liang. A practical evaluation of radio signal strength for mobile robot localization. In *IEEE International Conference on Robotics and Biomimetics*, pages 516–522, December 2009.

Jun Ye. Adaptive control of nonlinear pid-based analog neural networks for a nonholonomic mobile robot. *Neurocomputing*, 71(7-9):1561–1565, 2008.

Soo-Yeong Yi and Byoung-Wook Choi. Autonomous navigation of indoor mobile robots using a global ultrasonic system. *Robotica*, 22:369–74, July 2004.

Moustafa Youssef. *The Horus WLAN location determination system*. PhD thesis, University of Maryland, Maryland, 2004.