

INTELLIGENT

TRANSPORTATION

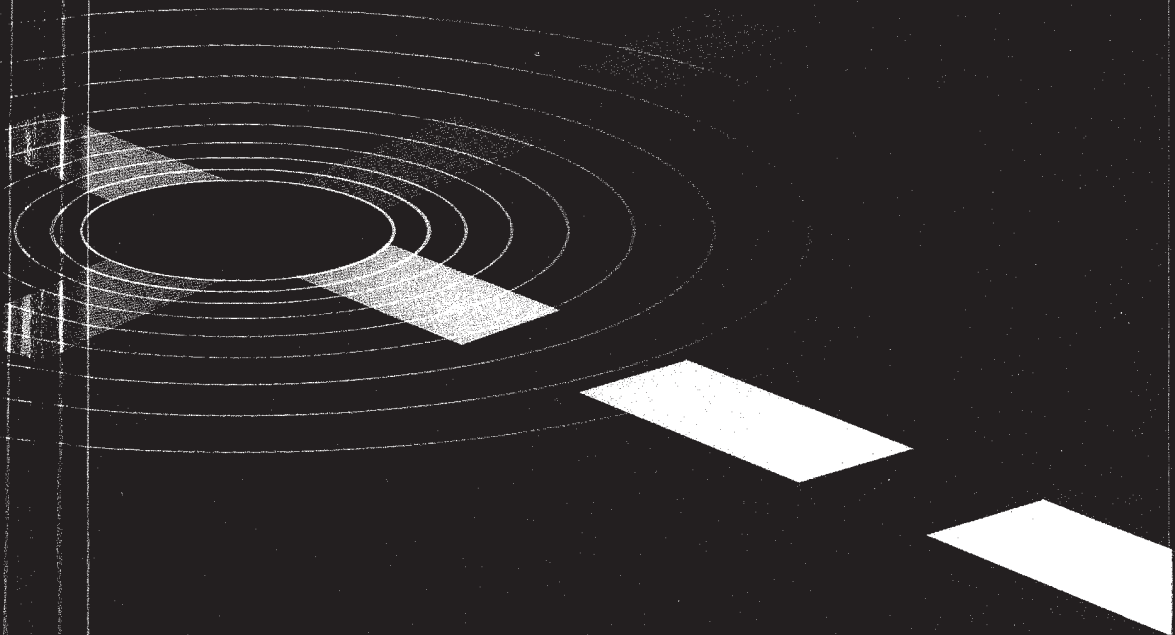
SYSTEMS

I

T

S

Vehicle Location and Navigation Systems



Yilin Zhao

VEHICLE LOCATION AND NAVIGATION SYSTEMS

For a complete listing of the *Artech House Intelligent Transportation Systems Library*, turn to the back of the book.

VEHICLE LOCATION AND NAVIGATION SYSTEMS

Yilin Zhao

Artech House, Inc.
Boston • London

Library of Congress Cataloging-in-Publication Data

Zhao, Yilin.

Vehicle location and navigation systems /Yilin Zhao.

p. cm.

Includes bibliographical references and index.

ISBN 0-89006-861-5 (alk. paper)

1. Intelligent Vehicle Highway Systems. 2. Motor vehicles—Automatic location systems. 3. Electronics in navigation.

I. Title.

TE228.3Z45 1997

629.2'7—dc21

97-4200

CIP

British Library Cataloguing in Publication Data

Zhao, Yilin

Vehicle location and navigation systems

1. Intelligent Vehicle Highway Systems 2. Motor vehicles—Automatic location systems

I. Title

625.7'94

ISBN 0-89006-861-5

Cover design by Jennifer Makower

© 1997 ARTECH HOUSE, INC.

685 Canton Street

Norwood, MA 02062

All rights reserved. Printed and bound in the United States of America. No part of this book may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher.

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Artech House cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

International Standard Book Number: 0-89006-861-5

Library of Congress Catalog Card Number: 97-4200

10 9 8 7 6 5 4 3

To everyone who provided help and support



CONTENTS

Preface	xiii
Acknowledgments	xvii
Chapter 1 Introduction	1
1.1 Brief History	2
1.2 Modern Vehicle Location and Navigation	6
References	12
Part I Basic Modules	15
Chapter 2 Digital Map Database Module	17
2.1 Introduction	17
2.2 Basic Representations	18
2.3 Reference Coordinate Systems	20
2.4 Standards	28
2.4.1 Geographic Data Files	28
2.4.2 Digital Road Map Association	30
2.4.3 Spatial Data Transfer Standard	31
2.4.4 Truth-in-Labeling Standard	32
2.5 Proprietary Digital Map Databases	33
2.5.1 Etak	33
2.5.2 Navigation Technologies	35

2.6	Digital Map Compilation	36
2.6.1	Data Structures	37
2.6.2	Compiler Structure	38
2.6.3	Hierarchical Maps	38
	References	42
Chapter 3	Positioning Module	43
3.1	Introduction	43
3.2	Dead Reckoning	45
3.3	Relative Sensors	46
3.3.1	Transmission Pickups	47
3.3.2	Wheel Sensors	50
3.3.3	Gyroscopes	53
3.4	Absolute Sensors	56
3.4.1	Magnetic Compasses	56
3.4.2	Global Positioning System	63
3.5	Sensor Fusion	75
3.5.1	Simple Filters	75
3.5.2	Kalman Filters	76
3.5.3	Other Fusion Methods	78
	References	79
Chapter 4	Map-Matching Module	83
4.1	Introduction	83
4.2	Conventional Map Matching	85
4.2.1	Semi-Deterministic Algorithms	85
4.2.2	Probabilistic Algorithms	86
4.3	Fuzzy-Logic-Based Map Matching	94
4.3.1	Fuzzy-Logic-Based Algorithms	95
4.4	Other Map-Matching Algorithms	100
4.5	Map-Aided Sensor Calibration	101
	References	102
Chapter 5	Route-Planning Module	105
5.1	Introduction	105
5.2	Shortest Path	107
5.2.1	Dijkstra's Shortest Path Algorithm	107
5.2.2	Modified Shortest Path Algorithm	109
5.3	Heuristic Search	112
5.3.1	A* Algorithm	112
5.4	Bidirectional Search	118
5.5	Hierarchical Search	122

5.6 Other Algorithms	123
References	125
Chapter 6 Route Guidance Module	129
6.1 Introduction	129
6.2 Guidance While En Route	130
6.2.1 Maneuver Generation	131
6.2.2 Route Following	133
6.3 Guidance While Off-Route	135
6.4 Guidance With Dynamic Information	136
References	141
Chapter 7 Human-Machine Interface Module	143
7.1 Introduction	143
7.2 Visual-Display-Based Interfaces	145
7.2.1 Display Technologies	146
7.2.2 Touch Screens	151
7.2.3 Additional Design Considerations	156
7.3 Voice-Based Interfaces	158
7.3.1 Speech Recognition	160
7.3.2 Speech Synthesis	164
References	166
Chapter 8 Wireless Communications Module	169
8.1 Introduction	169
8.2 Communications Subsystem Attributes	170
8.2.1 Coverage	171
8.2.2 Capacity	174
8.2.3 Cost	175
8.2.4 Connectivity	175
8.3 Existing Communications Technologies	175
8.3.1 Paging	178
8.3.2 Cellular	180
8.3.3 Personal Communications Services	183
8.3.4 Private Land Mobile Radio Systems	185
8.3.5 Radio Data Networks	191
8.3.6 Broadcast Subcarriers	196
8.3.7 Short-Range Beacons	202
8.3.8 Satellites	205
8.4 Communications Subsystem Integration	207
References	212

Part II	Systems	215
Chapter 9	Autonomous Location and Navigation	217
9.1	Introduction	217
9.2	Vehicle Location	218
9.2.1	Stand-Alone Technologies	219
9.2.2	Terrestrial Radio Technologies	221
9.2.3	Satellite Technologies	228
9.2.4	Interface Technologies	229
9.3	Vehicle Navigation	229
9.3.1	Coping With Complex Requirements	229
9.3.2	Dual-Use Navigation and Entertainment Components	236
	References	237
Chapter 10	Centralized Location and Navigation	239
10.1	Introduction	239
10.2	Automatic Vehicle Location	242
10.2.1	Centralized Approach	243
10.2.2	Distributed Approach	247
10.3	Dynamic Navigation	253
10.3.1	Centralized Approach	254
10.3.2	Distributed Approach	256
10.4	Applications: Mayday	259
	References	263
Chapter 11	A Case Study: ADVANCE	265
11.1	Introduction	265
11.2	Traffic Information Center	266
11.3	Mobile Navigation Assistant	274
11.3.1	Hardware	275
11.3.2	Software	281
11.4	Communications Network	286
11.5	Initial Evaluation Results	289
	References	293
Chapter 12	Conclusions	295
12.1	Past Lessons	295
12.2	Future Directions	299
	References	303
Appendix A	Transformation Between Cartesian and Ellipsoidal Coordinates	305
	References	307

Appendix B Transformation Between UTM and Ellipsoidal Coordinates	309
References	312
Appendix C Positioning Sensor Technologies	313
Appendix D Least Squares Algorithm	317
References	318
Appendix E Kalman Filter Algorithm	319
References	320
Appendix F Fuzzy Logic Background	321
References	326
About the Author	327
Index	329



PREFACE

This book can be used as a reference for engineers, managers, or professionals who wish to familiarize themselves with or become professionally involved in vehicle location and navigation systems, or who have worked in a specific area of vehicle location and navigation and wish to better understand the entire field. The book can also be used as a textbook for undergraduates or beginning graduate students in the field of intelligent transportation systems (ITS, formerly IVHS) and related disciplines.

This is the first book to provide a detailed description of both the principles and practices of modern vehicle location and navigation systems in a single source. It combines information scattered among many different engineering fields into a single volume, a comprehensive introduction to the highly interdisciplinary field of ITS. It is based on several years of hands-on experience, and focuses on important aspects of each individual system module: its principles, implementation, integration, and combinatorial complexity. It permits a broad audience to learn about this fast-growing field without any prior knowledge. The concepts and algorithms are applicable to a broad range of location and navigation systems and form the foundation needed to understand, design, and implement advanced ITS. To help professionals with different backgrounds, the book attempts to limit the amount of mathematics used, and includes many newly developed technologies, practical examples, summary tables, and recent references. Even nonengineers should be able to understand much of the material.

The book is divided into two parts: Part I, Basic Modules (building blocks, Chapters 2 through 8), and Part II, Systems (integrated modules, Chapters 9 through 11). Chapter 1 is a general introduction to the vehicle location and navigation field, with a historical background and introduction to modern vehicle location and navigation systems. Chapters 2 through 8 describe the basic building modules from which systems are constructed. Chapter 2 discusses the digital map database generally used in vehicle location and navigation systems, as well as relevant reference coordinate systems and standards. Chapter 3 studies the positioning sensors and fusion methods used to locate the vehicle. Chapter 4 describes the map-matching algorithms employed for enhancing the positioning module and systems. Chapter 5 discusses various route-planning algorithms for finding the route that results in a minimum travel cost. Chapter 6 explains how to guide the vehicle along a given planned route that was derived on the basis of either the static travel cost or the dynamic travel cost. Chapter 7 discusses various human factors guidelines and technologies used for the human-machine interface, which interacts with the location and navigation computer and devices. Chapter 8 examines the different wireless communications technologies available for supporting vehicle communications.

Chapters 9 through 11 are devoted to systems. Chapter 9 discusses how to integrate the modules discussed in Part I to construct a variety of realistic autonomous vehicle location and navigation systems capable of operating without the support of a centralized remote computing and communications facility. Chapter 10 explains how a wireless communications module might be integrated with the vehicle and infrastructure to enhance in-vehicle systems using a centralized host computer. Chapter 11 uses an actual dynamic route guidance based system as a case study to shed some light on system architecture and implementation issues. The book concludes with Chapter 12, which describes past lessons learned and future directions projected.

Many useful summary tables and recent references are given, and many new technologies are introduced and practical questions discussed. Even though some technologies may not be discussed in detail here, the author has attempted to give comprehensive listings of available technologies wherever possible, in order to present a broad view of the field. When applicable, recent references are cited to direct readers to the latest information on and trends in the relevant technologies. Many newly developed technologies and practical questions are discussed in the book, including new technologies such as fuzzy-logic-based map-matching algorithms, the field-emitter display (FED), a newly invented software technology for map display that simulates a three-dimensional view, along with practical questions such as why the output of a GPS receiver might not match a certain digital map and why a certain navigation system might not work as expected right after an automatic car wash, and so forth. In addition, questions are provided at the end of each chapter to stimulate readers to think about the material learned in that chapter and to ponder these interesting problems themselves.

Because of recent widespread and rapid development of the Internet's World Wide Web (WWW) technology, the book also lists many addresses of ITS-related home pages in its references. The advantage of these "soft" pages is that you can read these references instantaneously if you have access to the Internet and a WWW browser. However, unlike "hard" printed papers and books, they may be unreachable due to computer or network problems. In addition, the URL addresses listed may be changed at any time by their owners, making them inaccessible from the listed addresses.

Due to the broad and rapidly expanding nature of vehicle location and navigation systems, it is difficult to cover the entire field in great detail. To limit the scope and conserve space, more emphasis has been given to the vehicle side of the system. The author intended the materials selected in the book to represent the most fundamental and significant aspects of recent developments. However, relevant concepts and references may have been omitted. The author would appreciate your comments and seeks your advice for improvements to the book. He will also maintain a WWW page for online receipt of suggestions and on-line posting of any errors not detected and corrected before the book went to print. The URL address of this page is <http://birch.dlut.edu.cn/~yzhao/>.



ACKNOWLEDGMENTS

I would like first to thank Motorola, Inc., for their permission to publish this book. However, the opinions expressed in the book are the personal views of the author, and do not necessarily represent positions held by Motorola.

This book would not have been possible without the help and support of my family, friends, and colleagues. I would like to express my gratitude to all those who have contributed in one way or another to the completion of this book. In particular, I would like to thank the following people for their contributions: Allan M. Kirson, who carefully reviewed eleven chapters and made many useful suggestions in addition to his support; Steve Albrecht, who patiently reviewed six preliminary chapters; Phil Pollock and Larry Willis, who made valuable contributions to help meet the publication deadline; Paul D. Makinen, who copyedited the book with great enthusiasm; an anonymous reviewer who provided many constructive suggestions; and many other people who reviewed parts of the book in their respective specialties and are listed alphabetically: Michael Barnea, J. Blake Bullock, Yanming Cheng, John Dillenburg, Thomas E. Hayosh, Garry C. Hess, Kenneth B. Hohl, Wei-Wen Kao, Howard L. Kennedy, Daniel Schwartz, Witold Wojciechowski, and the ADVANCE Project Office.

I would also like to acknowledge the help given by Kan Chen, Paul Green, Jay Jayapalan, Cynthia Keesan, Huangsheng Li, Charles J. Mages, Don Mills, Jim Nickel, Vijay Ramanathan, Heinz Sodeikat, Chengyi Xu, and librarians Constance M. Bytnar

and Mary Lou Kutscha, as well as the support provided by Robert H. Reuss, Mike Smith, Cindi Moreland, Robert P. Denaro, Paul L. Dowell, Steven F. Gillig, and Lawrence E. Connell. I apologize in case anyone's name was accidentally omitted. This book is dedicated to all of those who provided assistance and support.

In addition, I would like to thank my teachers both in China and America for my early training and their later influence and inspiration. Thanks also go to many other Motorola IVHS team members who helped me during various projects that we worked on together, Motorola employees who provided and verified product information, and the outside companies which supplied their products and technical information for the book.

Finally, special thanks go to my wife, Ming Xiang, for her assistance and her sacrifice in allowing me to spend thousands of hours of family time on the book, and to my son, Lianhan Zhao, for his understanding.

Thanks also goes to my younger sisters Youlin Zhao and Yulin Zhao, who spent more time taking care of our elderly parents during my undergraduate and graduate studies, which is traditionally considered a son's responsibility in China. My late parents, Yueduan Lin and Weiduo Zhao (spelled Wei-to Chao in 1948, when he graduated from Case Western Reserve University in Cleveland), gave me all their love and were a great source of inspiration. Their encouragement, guidance and support will always be with me.

CHAPTER 1



INTRODUCTION

Road transportation systems have undergone considerable increases in complexity and congestion. In particular, surface vehicle ownership and the use of vehicles are growing at rates much higher than the rate at which roads and other infrastructures are being expanded. One recent study found that, in the major urban areas of the United States alone, the total cost of traffic congestion exceeds \$47.5 billion per year [1]. Traffic congestion wastes more than 14.35 billion liters (3.79 billion gallons) of fuel and 2.7 billion hours of work time per year. These numbers have continued to increase by 5% to 10% per year through the 1990s. Transportation authorities are increasingly turning to existing and new technologies to preserve mobility, improve road safety, and minimize congestion, pollution, and environmental impact. The roads and vehicles of tomorrow will see major changes brought about by the integration of computers, information, and communications technologies. This exciting new field is now known under the name intelligent transportation systems (ITS) (formerly known as intelligent vehicle highway systems (IVHS)).

The goal of ITS is to apply advanced technologies to make transportation operate more safely and efficiently, with less congestion, pollution, and environmental impact. The theory and practice of ITS are currently among the most intensely studied and promising areas in transportation, computer, information, communication, and systems science and engineering, and one that will certainly play a primary role in our future lives. If ITS lives up to its hopes, it will eventually affect virtually

everyone in the world. In working toward this goal, ITS can take many different forms. In this book, we concentrate only on a specific portion of ITS: land-based vehicle location and navigation systems. However, the principles, concepts, and algorithms introduced in the book are applicable to a wide range of location and navigation systems, and form the foundation necessary to understand, design, and implement advanced ITS.

1.1 BRIEF HISTORY

Research on and development of vehicle location and navigation systems have a long history in our civilization, even though systems have only recently started to reach the world market. The earliest vehicle navigation system dates back thousands of years to the time when a south-pointing carriage was first invented in ancient China, around 2600 B.C. according to legend. The invention of this carriage has been written about since the San Guo (Three Kingdoms) period (220–280). During the Song Dynasty (960–1279), because of the invention of movable-type printing, detailed descriptions of this carriage were recorded and have been presented. Reconstructions were built in the San Guo period, 1027 and 1107. This carriage had a two-wheeled cart on which a human figure was mounted. This figure was kept continually pointed toward the south, no matter which way the cart was moving, through the action of a gear train. The basic mechanism of this carriage is similar to that of the modern differential odometer. The late Professor Zhenduo Wang of Beijing Normal University reconstructed this carriage from descriptions in a Yuan Dynasty book [2], as shown in Figure 1.1. For English literature and the working mechanism, refer to [3].

Almost in parallel with the invention of the south-pointing carriage, another interesting invention was the li-recording (distance-measuring) drum carriage. Like the south-pointing carriage, it had a set of gears. As the carriage moved, the gears turned to drive the arms of two wooden human figures. These figures sat on the carriage, one facing the drum and the other facing a gong. During the journey, one figure would strike the drum once every li, and another figure would strike the gong once every 10 li. A li is almost half of a kilometer. The fundamental mechanism of this carriage is similar to that of the modern odometer. From descriptions in the book mentioned above [2], Professor Wang reproduced this carriage, as shown in Figure 1.2.

As discussed in [3,4] and references therein, basic location and navigation technologies such as the odometer, differential odometer, and magnetic compass were invented about 2000 years ago. For the past century, these and other technologies have been gradually incorporated into modern automobiles. To learn more about the world's roads, bridges, and the vehicles that traverse them, refer to [5]. In the United States, the first automotive road map was published in 1895, and road signs were installed and roadways numbered early in this century. Mechanical route

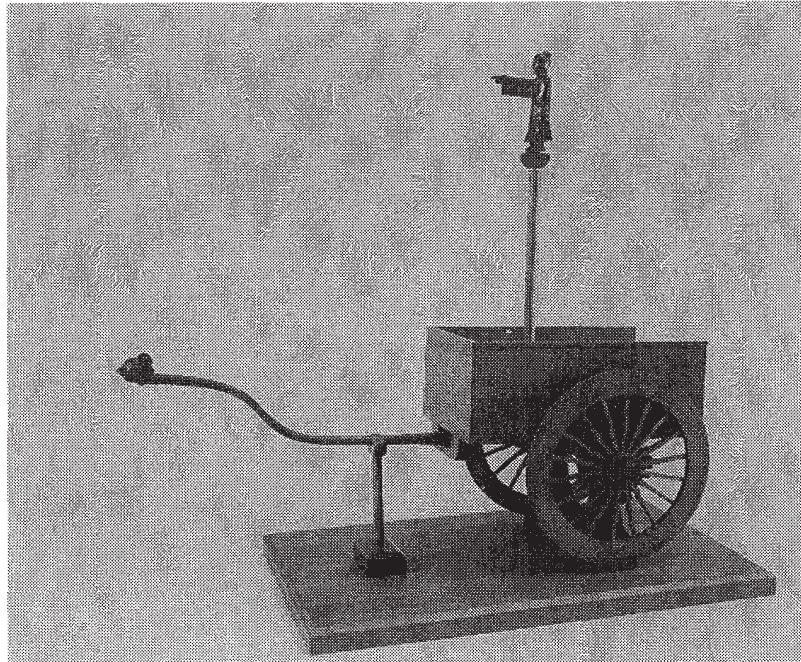


Figure 1.1 South-pointing carriage. (Courtesy of the Museum of Chinese History, Beijing.)

guidance devices, introduced around 1910, incorporated route map information in various forms such as sequential instructions printed on a turntable, punched in a rotating disk, or printed on a moving tape [6]. Each of these devices was driven by an odometer shaft in synchronization with the distance traveled along a route. Many mechanical vehicle route guidance devices patented from 1910 to 1920 provided explicit real-time route instructions automatically.

As roads became better marked and road maps became more accurate, interest in route guidance devices waned. Only a few developments occurred from the 1920s to the 1960s. During World War II, an electronic vehicle navigation system was developed in the United States for jeeps and other military vehicles [7]. This system had a magnetic compass whose needle position was read by a photocell. A servomechanism was driven by the compass output to rotate a mechanical shaft corresponding to the vehicle heading. This shaft was coupled to a mechanical computer that derived travel distance from an odometer and converted it into x and y components. Also driven by the compass shaft, these location components were used to automatically plot the vehicle trajectory on a map of appropriate scale.

In the late 1960s, an Electronic Route Guidance System (ERGS) was proposed by the U.S. Bureau of Public Roads (now the Federal Highway Administration) [8].

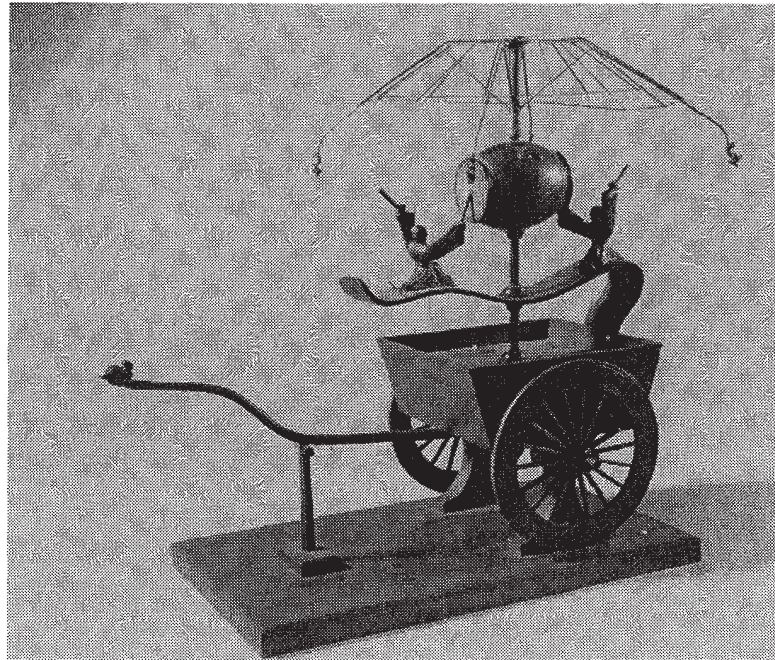


Figure 1.2 Li-recording drum carriage. (Courtesy of the Museum of Chinese History, Beijing.)

The ERGS was a navigation system with wireless route guidance capability to control and distribute the traffic flow. A network of short-range beacons (proximity beacons) was used for two-way communications. An in-vehicle console with thumbwheel switches would permit the driver to enter a destination code. When approaching key intersections, the code would be transmitted from an on-vehicle transceiver to the beacon over a communication loop antenna embedded in the road surface of each approaching lane at the intersection and connected to a roadside controller via a coaxial cable. This controller was also connected to a centralized host for accessing traffic data. After receiving the destination code, the controller decoded it and planned a best route. The route guidance instructions were sent to the vehicle over the beacon before the vehicle left the vicinity of the loop antenna. These guidance instructions were then displayed on the vehicle's head-up display (HUD). Due to limited funding, this project was never fully implemented although successful tests were conducted. Despite never having been implemented, this project introduced the concept of centralized dynamic navigation (route guidance). Similar projects were later developed and tested in Japan and Germany during the 1970s. Note that much of the location and navigation terminology used in this section will be explained briefly in the following section and comprehensively in the following chapters.

In the early 1970s, an autonomous navigation system was introduced in the U.S. [9]. This system used a dead-reckoned positioning module assisted by a map-matching algorithm to locate the vehicle. A second version of the system could display route guidance instructions on a plasma display panel (PDP) once the vehicle's location along the route was confirmed. Similar systems were also developed by other groups in the United States and the United Kingdom. During the past 20 years, many key components and technologies for vehicle location and navigation have begun to mature.

During the past decade, vehicle location and navigation systems and other ITS-related systems have been rapidly gaining momentum worldwide. For detailed information on the projects and systems mentioned below, refer to [6,10-13], the numerous ITS-related World Wide Web (WWW) sites mentioned below, various ITS magazines and journals, as well as the proceedings of conferences organized by IEEE, the European Road Transport Telematics Implementation Coordination Organization (ERTICO), Vehicle, Road and Traffic Intelligence Society, Japan (VERTIS), and the Intelligent Transportation Society of America (ITS America).

In Japan, the ITS movement began in 1973 with the CACS project. The CACS project was similar in concept to the ERGS project. In the 1980s, an autonomous navigation system incorporating a color display monitor with a CD-ROM as the digital map storage device appeared on the market. Since then, a variety of navigation products have evolved, which include many new features such as map matching, use of a global positioning system (GPS) receiver, and voice guidance. More than 1.2 million vehicle navigation units have been sold to consumers. ITS field tests range from RACS and AMTICS in the 1980s to ATIS, VICS, and UTMS today. RACS was developed by the Public Research Institute of the Ministry of Construction and 25 private companies. AMTICS was initiated by the Traffic Management Technology Association in collaboration with private companies under the supervision of the National Police Agency (NPA). Later in 1991, RACS was combined with AMTICS to become the VICS program, with the support of an additional government agency, the Ministry of Posts and Telecommunications. VICS has been conducting tests and began deployment of technology in April 1996 for guiding vehicles to individual destinations, given real-time traffic information, using infrared beacons, microwave beacons, or FM broadcast subcarriers as transmission media (dynamic navigation). ATIS is being promoted by the Metropolitan Police Department, and uses cellular radio for vehicles and telephone lines for personal computers in fleet management centers. UTMS, initiated by NPA, plans to enhance existing traffic control systems by including traffic data collection and traffic information as well as traffic signal control and dynamic navigation.

In Europe, the ITS movement started in the late 1970s with the ALI project. The system was similar to the approach proposed in the ERGS project and tested in the CACS project. Autonomous navigation systems developed in the 1980s included CARIN and EVA. CARIN used dead reckoning with map matching and a color

monitor for map display. It was the first navigation system to use a CD-ROM for map storage. EVA was demonstrated in 1983. In addition to dead reckoning, map matching, and turn-by-turn route guidance, it provided guidance to the driver in the form of both visual displays and synthesized voice output. ITS field tests (programs) known as PROMETHEUS and DRIVE have been operating since the mid 1980s. The former program was initiated by the automotive industry and has concentrated primarily on developments within the vehicle and its immediate environment. The latter program was initiated by the European Community and has been concerned mainly with infrastructure requirements. The two programs have been working together very closely and have many formal and informal links. Recently, PROMETHEUS has evolved into PROMOTE, and DRIVE has become a program under the name Telematics Applications for Transport and Environment.

In the United States, after the navigation systems proposed and developed in the late 1960s and early 1970s, an autonomous navigation system called Navigator was introduced commercially in the mid-1980s. It used a digital map database, dead reckoning with map matching, and a map display with icons showing current location and destination. Another autonomous navigation system named Guidestar, which incorporated a GPS receiver, appeared on the market in 1994. Many ITS field tests were conducted in the early 1990s, such as Pathfinder, TravTek, and ADVANCE. More projects are undergoing operational tests (see Table 10.1 in [10]) and even more are under construction. Most of these projects have been jointly proposed and developed by the public and private sectors. The purpose of these programs is to perform an initial assessment of the feasibility and utility of in-vehicle navigation, dynamic navigation, travel information, and various communication media, as well as other ITS concepts.

1.2 MODERN VEHICLE LOCATION AND NAVIGATION

As seen from the historical developments discussed earlier, human beings have never stopped pursuing the dream of great mobility. Thousands of years of civilization have led to the modern vehicle location and navigation systems of today. There are, and always will be, a range of systems from the very low end to the very high end. The low end might be very simple systems that detect the location of a vehicle or mobile device with human intervention, whereas the high end might be complex systems that navigate the vehicle automatically through the road network, assisted by real-time traffic information provided over a wireless communications network.

This book uses a modular approach to cover a broad range of systems. As we will learn in later chapters, a modern location and navigation system often consists of some or all of the modules shown in Figure 1.3. In other words, sophisticated location and navigation systems require integration of numerous functions and technologies. Therefore, we first focus on important aspects of each individual

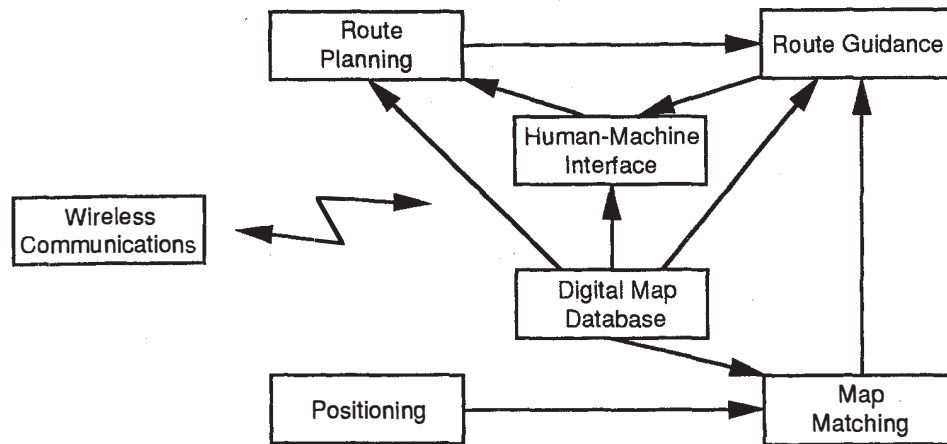


Figure 1.3 Basic modules (building blocks) for a location and navigation system.

module, the principles underlying it, and the implementation of these principles. We then examine how systems can be constructed from these modules, discuss the integration of these modules, and consider the additional complexity that arises when these modules are combined.

The modules depicted in Figure 1.3 can take different forms. Some of the modules were briefly described in the preceding section. These modules can be implemented by different hardware and software components. We now summarize the function of each individual module.

A digital map database contains digitized map information with a predefined format, which can be processed by a computer for map-related functions such as identifying and giving locations, road classifications, traffic regulations, and travel information. Because a map is used to represent the geometry of the Earth's surface, we must understand the relevant reference coordinate systems used by different map databases in order to develop map-related functions properly. To ease the pain of dealing with different digital maps, standards have been developed to guide development and applications. A positioning module fuses different sensor outputs or uses radio signals for automatic determination of the position of the vehicle or a mobile device to identify the road traveled and each intersection approached. A typical stand-alone technique is dead reckoning, and a typical radio-signal-based technique uses GPS receivers. A variety of fusion methods have been developed for integrated sensors. *Map matching* is a method of matching the position (or trajectory) measured or received by a positioning module to a position associated with a location (or route) on a map provided by the map database module. This technique can improve the accuracy of the positioning module, provided the map database is of

reasonable precision. A typical precision requirement for urban areas is within 15m of "ground truth."

Route planning is the process of helping vehicle drivers plan a route prior to or during their journey, based on a given map provided by the map database module, if available, along with real-time traffic information received via a wireless communications network. A commonly used technique is to find a minimum-travel-cost route based on criteria such as time, distance, and complexity. *Route guidance* is the process of guiding the driver along the route generated by the route-planning module. It requires the help of an accurate positioning and map database in order to determine current vehicle position and generate proper real-time guidance instructions, often turn by turn. A human-machine interface allows users to interact with the location and navigation computer and devices. Map displaying, route planning, route guidance, and other activities are directed to the machine and fed back to the user via this module. A wireless communications module further improves the performance and increases the functionality of the system. Through one or more of various communications networks (for which more than 12 different technologies are available), a vehicle and its occupants or transportation management systems can receive updated traffic information or traffic reports, which make the in-vehicle system or whole road network operate more efficiently and safely. Of the several communications technologies discussed in this book, CDPD has been selected by the U.S. Joint ITS Architecture team as the preferred method for providing traffic and route-guidance information to motorists. Furthermore, a centralized facility will even be able to locate and navigate vehicles through its road network. All of these modules are discussed in individual detail in Part I, Chapters 2 through 8.

We will then learn how to integrate the above modules to construct working location and navigation systems in Part II, Chapters 9 through 11. Many practical systems are discussed to help readers understand these chapters. As in Part I, a system can take different forms, each of which may include a variety of different modules. First, autonomous location and navigation systems, such as vehicle location (or tracking) systems and navigation systems with no remote host or centralized computing facilities, are discussed. One of the main characteristics of this type of system is that all location and navigation capabilities are placed solely on the vehicle. Second, centralized location and navigation systems are examined. These systems utilize communications networks, host facilities, and other infrastructure to perform multivehicle location and navigation. For any specific system, three design issues must be considered: placement of location and navigation capabilities (on a host in a fixed infrastructure or on each individual vehicle), location accuracy and update frequency, and selection of wireless communications technologies. Third, a working dynamic route-guidance system is examined in detail. Unlike a static system, a dynamic system uses databases updated by real-time traffic information to guide the vehicle. This study will build on the knowledge and techniques learned in the previous chapters and demonstrate how they are applied in the real world. We will

also discuss several system architecture, design, and implementation issues. Finally, in Chapter 12, we discuss lessons learned from past experience and predict future directions for vehicle location and navigation systems.

As stated, many modern systems consist of some or all of the modules in Figure 1.3. For instance, an autonomous in-vehicle navigation system may exclude only the wireless communications module. It may interact with a map database to plan a minimum-travel-cost route based on a given destination specified by the driver. Knowing the route, it may work with the positioning module and the map-matching module to guide the driver to follow a particular route to the destination. The guidance instructions may consist of voice, audible tones, and/or display text and graphics. Integration with a wireless communications network and a centralized host, which takes over some of the location or navigation functions, produces a centralized navigation system.

So that readers might better understand the various modules and systems, some current or planned products or systems are mentioned. Such information was gathered via extensive literature searches, direct phone interviews, or proofreadings by manufacturers. This information is believed correct at the time of publication. However, it may be changed or updated anytime during any phase of product or system development. In addition, to concentrate on the main theme of the book, we emphasize the modules used in the vehicle and systems employed in the vehicle. Application to other system architectures should not pose too much of a challenge once the reader has mastered the fundamental principles discussed here.

Despite our modular approach, readers must keep in mind that a top-down approach is generally used in product development. A top-down approach typically includes identifying system requirements, determining functions and system architecture, specifying appropriate modules, selecting hardware components and software tools, and designing, implementing, integrating, and testing the system. To facilitate our study, we will introduce the various individual location and navigation modules first, but this does not mean we can ignore issues of system architecture and integration. For any location and navigation system to be successful, it must be based on a good architecture. This system architecture must provide a stable basis on which the system can further evolve, through adding or upgrading individual modules in response to changing requirements and operating environments over the normal lifetime of the system. Sophisticated systems, in particular, are expected to have complex system integration. Without proper processes and management, system integration can be delayed considerably, even when the individual modules function properly.

Because of explosive technological advances, modern vehicle location and navigation systems have improved greatly during the last decade. A variety of communication tools has been used to educate the public and to promote further technical advances. Lately, dozens of Internet WWW pages have emerged to share information in this exciting new field. Interested readers can browse sites worldwide for instant

research and development project status, the latest news, transportation statistics, real-time road traffic conditions, and much more. A few sample sites are listed at the end of this chapter [14–26].

Market studies have predicted great potential for vehicle location and navigation equipment and systems. One study estimates that by the year 2000 the world market for vehicle navigation units integrated with GPS receivers will be \$3 billion (Fig. 1.4). Similar studies have been conducted in their respective regions by the Japan GPS Council, Nikko Research Center, Elsevier Advanced Technology, Allied Business Intelligence, Frost & Sullivan, NevTech, SRI, the University of Michigan, and many other organizations. One Japanese forecast predicts a total sales of 3 million units [27]; one European forecast estimates a total sales of 2.5 million units [28]; and one American forecast projects a total sales of 22.5 million units [29] for in-vehicle navigation systems in the year 2005 alone. Assuming that the unit cost will be \$500 by that time, this leads to a total sales volume of \$1.5 billion, \$1.25 billion, and \$11.25 billion, respectively, in each of these regions.

Vehicle location and navigation systems have already gained great user acceptance worldwide. In Japan, sales of vehicle navigation systems to consumers reached 530,000 units in 1995, up 73.2% from 1994, as reported by the Japan Electronic Industries Association. The ERTICO study, based on field tests in Europe, has confirmed high levels of user acceptance for in-vehicle route guidance and RDS-TMC (Radio Data System-Traffic Message Channel) systems [30]. Consumers are

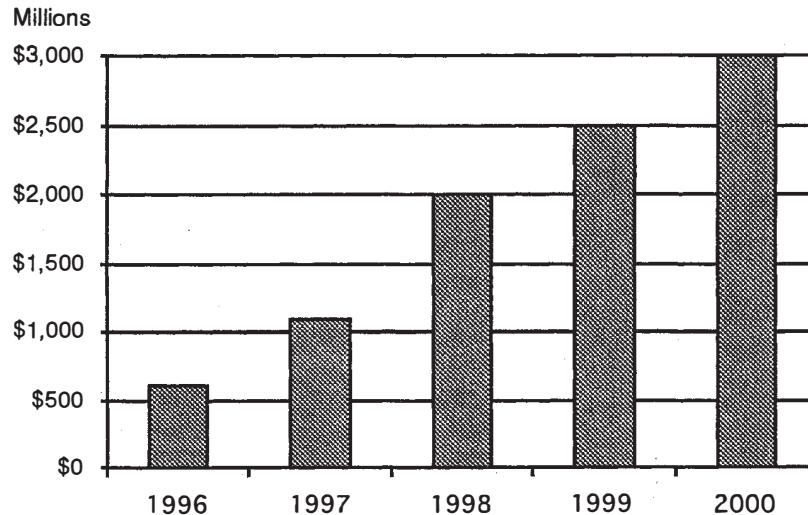


Figure 1.4 Worldwide market projections of GPS-based in-vehicle navigation systems. (Source: U.S. GPS Industry Council.)

willing to pay \$950 to \$1900 for the route-guidance unit and \$133 for the RDS-TMC equipment. In the United States, consumers are also willing to use vehicle location and navigation systems, as indicated by a recent study conducted in five main American metropolitan areas with 10 focus groups in October 1995 [31]. The groups were composed of cellular phone subscribers over age 25 from households with annual incomes of more than \$30,000. This study found that, as shown in Figure 1.5, interest was strongest in the so-called "mayday systems," which allow vehicle occupants to put themselves in instant wireless connection with a service center for emergency attention or assistance, while automatically reporting the vehicle location (see detailed discussions and the recently debuted products in Section 10.4).

All these market studies indicate that the vehicle location and navigation field has a promising future and should be economically rewarding. With this historical background and basic information on vehicle location and navigation systems in mind, we now proceed to a detailed description of these systems with the aim of

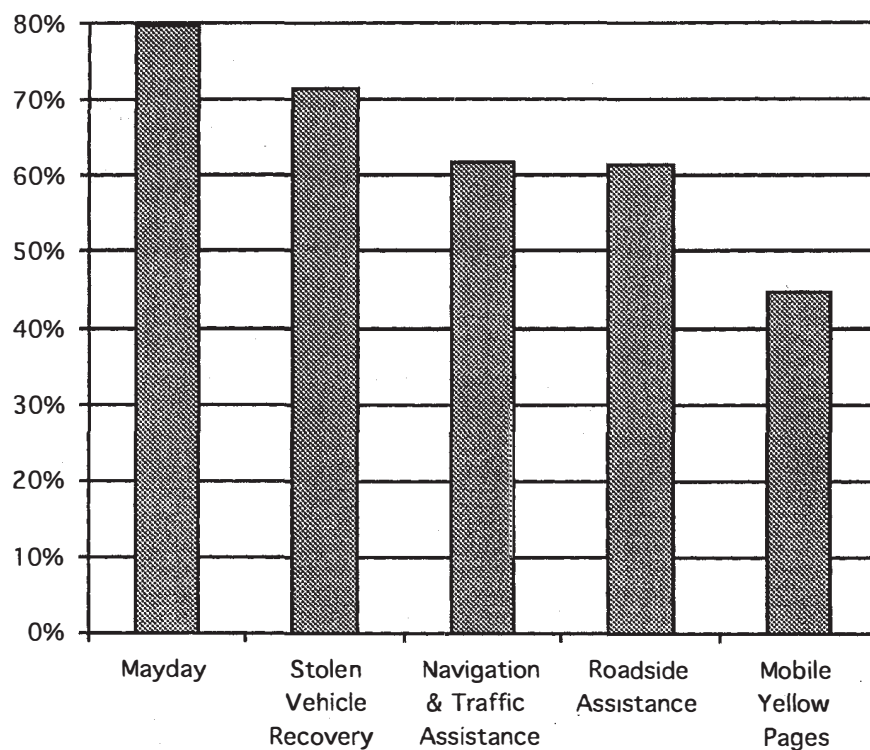


Figure 1.5 American consumer interest in location-based mobile services.

providing the reader with a better understanding of the underlying principles and practices.

References

- [1] Texas Transportation Institute, *Urban Roadway Congestion 1982-1992, Volume 1: Annual Report*, College Station, TX: Texas Transportation Institute, Sep. 1995.
- [2] T. Tuo and T. Alu (Eds.), *Song Shi* [History of The Song Dynasty, in Chinese], Yuan Dynasty, China, 1345.
- [3] J. Needham, *Science and Civilization in China*, Vol. 4, Part II, Cambridge, UK: Cambridge University Press, 1965.
- [4] J. Needham, *Science and Civilization in China*, Vol. 4, Part I, Cambridge, UK: Cambridge University Press, 1962.
- [5] M. G. Lay, *Ways of the World: A History of the World's Roads and of the Vehicles That Used Them*, New Brunswick, NJ: Rutgers University Press, 1992.
- [6] R. L. French, "From Chinese Chariots to Smart Cars: 2,000 Years of Vehicular Navigation," *Navigation: Journal of the Institute of Navigation*, Vol. 42, No. 1, Spring 1995, pp. 235-257.
- [7] D. J. Faustman, "Automatic Map Tracer for Land Navigation," *Electronics*, Vol. 17, No. 11, Nov. 1944, pp. 94-99.
- [8] D. A. Rosen, F. J. Mammano, and R. Favout, "An Electronic Route Guidance System for Highway Vehicles," *IEEE Trans. Vehicular Technology*, Feb. 1970, pp. 143-152.
- [9] R. L. French and G. M. Lang, "Automatic Route Control System," *IEEE Trans. Vehicular Technology*, May 1973, pp. 36-41.
- [10] I. Catling (Ed.), *Advanced Technology for Road Transport: IVHS and ATT*, Norwood, MA: Artech House, 1994.
- [11] A. M. Parkes and S. Franzen (Eds.), *Driving Future Vehicles*, Washington, DC: Taylor & Francis, 1993.
- [12] J. Walker (Ed.), *Mobile Information Systems*, Norwood, MA: Artech House, 1990.
- [13] R. Whelan, *Smart Highways, Smart Cars*, Norwood, MA: Artech House, 1995.
- [14] Bureau of Transportation Statistics, *Transportation Statistics*, <http://www.bts.gov/>, 1996.
- [15] P. David, *Subway Navigator*, <http://metro.jussieu.fr:10001/bin/cities/english>, 1996.
- [16] Georgia State DOT, *ATMS Home Page*, <http://www.georgia-traveler.com/>, 1996.
- [17] Institute of Transportation Engineers, *ITE on the Web*, <http://www.ite.org/>, 1996.
- [18] Intelligent Transportation Society of America, *Access ITS America*, <http://www.itsa.org/>, 1996.
- [19] Intelligent Transportation Systems Program, Princeton University, *Transportation Resources*, <http://dragon.princeton.edu/~dhb/>, 1996.
- [20] Sytadin, *Real-Time Road Traffic Conditions* [in French], <http://www.club-internet.fr:80/sytadin/map.html>, 1996.
- [21] The United States Department of Transportation, *DOT Home Page*, <http://www.dot.gov/>, 1996.
- [22] University of Leeds, *Institute for Transport Studies*, <http://its02.leeds.ac.uk/>, 1996.
- [23] University of Valencia, *Spanish National Road Network*, http://carpanta.eleinf.uv.es/dgt/ap_autol.html, 1996.
- [24] Victorian State Government, *VicRoads Home Page*, <http://www.vicnet.net.au/~vicroads/>, 1996.
- [25] Washington State DOT, *WSDOT Home Page*, <http://www.wsdot.wa.gov/>, 1996.
- [26] J. Werner, *ITS Online*, <http://www.itsonline.com/>, 1996.
- [27] T. Tachibama, *Car Navigation Forecast Data* [in Japanese], Autoparts Group, Nikko Research Center, Nov. 1994.
- [28] T. Long, *Advanced Transport Telematics (ATT): A European Forecast Study for the In-Car System Market to 2005*, Oxford, UK: Elsevier Advanced Technology, 1996.

- [29] P. Samuel, "Unit Prices Key to US Market for In-Vehicle Guidance," *ITS International*, June 1996, pp. 57-58.
- [30] ERTICO, "ITS Studies Confirm User Acceptance," *ITS International*, Mar. 1996, pp. 41-42.
- [31] C. J. Driscoll and D. Wolfe, *Location-Based Mobile Communications Services Marketing Research Study*, C. J. Driscoll & Associates and Wolfe & Company, May 1996.

PART I



BASIC MODULES

Part I is devoted to a discussion of subsystems of vehicle location and navigation systems. As mentioned in Chapter 1, to make our study easier, we will treat these subsystems as separate modules. After studying these basic modules, we discuss how to integrate these modules into complete working systems in Part II. In other words, the intricacies of systems engineering will be dealt with later in the book.

In this part, we look at digital map databases in Chapter 2, positioning subsystems in Chapter 3, and map-matching algorithms in Chapter 4. We then discuss route-planning algorithms in Chapter 5, route-guidance methods in Chapter 6, human-machine interfaces in Chapter 7, and various existing communications technologies in Chapter 8.

CHAPTER 2



DIGITAL MAP DATABASE MODULE

2.1 INTRODUCTION

A digital map database is an indispensable module for any vehicle location and navigation system that involves map-related functions. Without a map, it is very difficult for a traveler to explore an unfamiliar area and make correct decisions concerning the route. With a map as a medium, complex information can be communicated very easily.

A digital map database can provide many important functions for a vehicle location and navigation system. In short, it can help the system with the following tasks:

1. Display a map;
2. Locate an address or destination using a street address or nearby intersection;
3. Calculate a travel route;
4. Guide the driver along a precalculated route;
5. Match the sensor-detected vehicle trajectory with the known road network to determine more accurately the actual position of the vehicle;
6. Provide travel information such as travel guides, landmarks, hotel and restaurant information.

In general, there are two methods by which a computer can present a map to a traveler. One is to digitize a paper map (or photographs) using a scanner so it can be stored in and retrieved from computer memory as a digitized image or a raster-encoded structure (raster encoding). The other is to convert a paper map into a data structure or a vector-encoded structure so that it can be stored in and retrieved from computer memory easily, and presented in different ways to the user based on the functions or features required (vector encoding). Raster-encoded (scanned) maps can be easily produced and provide all the information contained in the original sources. They look exactly like the paper maps from which they are derived. On the other hand, vector-encoded maps require less storage and have faster access times. They are more flexible, easier to manipulate, and intrinsically relational in nature, since the data structure creates an implicit relationship between all of the various map elements. In the first (raster-encoded) method, values of each parameter of interest are provided for each cell (pixel) in a matrix over space. This type of map requires a large amount of storage and is difficult to use in many vehicle location and navigation applications that make use of mathematical models or calculations. Therefore, in this chapter we discuss only the second method, vector-encoded maps. Although we do not discuss the first method in detail, readers should be aware that this type of map still has applications despite its restrictions. For instance, it can be used for display purposes, for example, in vehicle location (tracking) systems where vehicle location or other information is superimposed on a map.

2.2 BASIC REPRESENTATIONS

Vector encoding is the representation of road network features using Cartesian geometry. A feature is denoted as an existing or planned item in the real world. Features are modeled by associating each feature with one or more primitives: points, lines, areas. A point is represented by two numbers with respect to a reference coordinate system. A line is represented by two connected points. An area is represented by a closed series of lines. The encoding is done in such a way that processing of the information may be automated using computers.

The purpose of the digital map database module is to define a network (Figure 2.1) of roads and associated attributes based on digital cartographic information, and then to compile it into a file or set of files that can be readily accessed by other modules in the location and navigation system. Digital cartographic information is geographic information that has been encoded in digital form. In other words, it is data digitized from existing maps or aerial photographs based on the primitives discussed earlier (and variations on these primitives). The attributes might include types of roads, street names, address ranges, expected driving speeds, connectedness, signs and signals, turn restrictions, points of interest, etc.

The digitized road network typically represents the road data using line segments whose endpoints (nodes) and shapes are defined in terms of latitude, longitude,

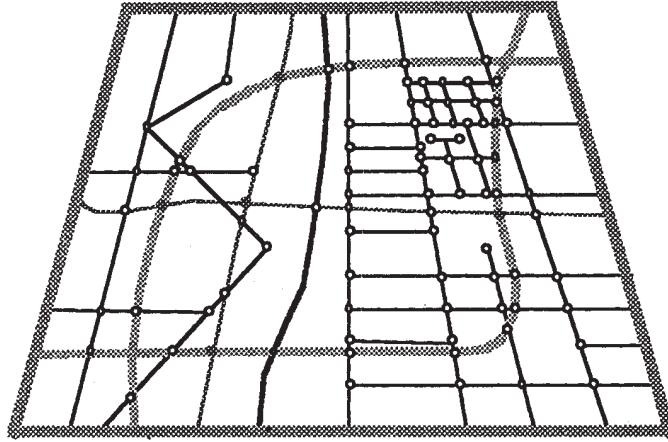


Figure 2.1 A typical road network.

and sometimes relative altitude. To understand the digital map database better, we need to define nodes, segments, and shape points:

- A *node* is a cross point or an endpoint of a street/road and is used to represent an intersection or a dead-end of a road. It is commonly represented in map databases by a latitude and a longitude.
- A *segment* is a piece of roadway between two nodes and is used to represent fragments of roadways and other features. A node can be the endpoint of a segment, but not the endpoint of a shape segment (see the definitions of shape point and shape segment next).
- *Shape points* are ordered collections of points which map the curved portion of a given segment (excluding the endpoints) to a series of consecutive straight-line pieces (called *shape segments*) in such a way as to make the calculated segment distance close to the actual length.

Samples of segments, nodes, and shape points are shown in Figure 2.2. From these definitions, we now know that a road can typically be represented by a sequence of straight lines (vectors if travel directions are considered) selected to approximate the real curvature of the road.

A computer program, usually referred to as a *database compiler*, takes an input file containing raw digital cartographic information provided by a vendor and produces a file or a set of output files in a compact format suitable for rapid access by the location and navigation system (see Section 2.6 for a high-level description of this process). In addition to integrating and formatting segments and nodes over

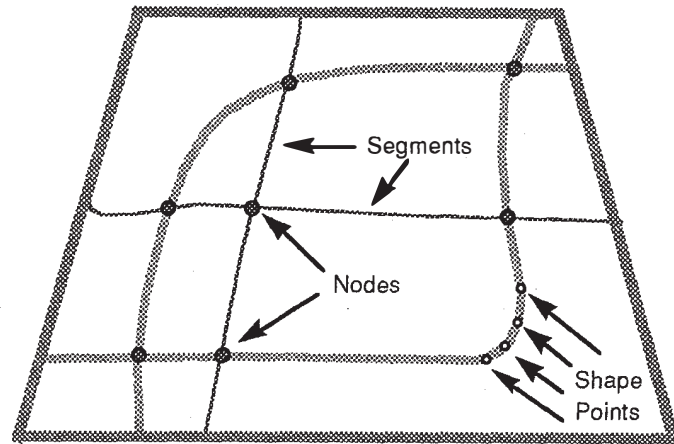


Figure 2.2 Segments, nodes, and shape points in a map.

a preselected area, the compilation process can also involve attributes such as road names, road segment lengths, segment direction and speeds, address ranges for roads, and various other relevant additional data for use by the target navigation system. The final output data files must be in a format suitable for access by the run-time system on the target computer, the dedicated computer which manages all the activities of the location and navigation system.

A digital map database is a very important component of any location and navigation system. Once the position (coordinates) of a vehicle has been determined, the map provides all location-related features. The on-board digital map database can be used as a reference to visualize and locate vehicles. Map-matching algorithms can use the geometry and topology of road networks to help determine vehicle location (as discussed in Chapter 4). Therefore, digital map databases are more than aids for pictorial representation of location, they themselves can be a means of providing vehicle location.

2.3 REFERENCE COORDINATE SYSTEMS

A digital map represents the geometry of the Earth's surface along with the attributes (such as the coordinates) of various geographical features. A review of the terminology and coordinate systems used should be helpful, and will assist in explaining various map references used by different organizations. We will also explain why we cannot blindly mix maps with different references, how to transform between the various coordinate systems, and where to find good sources for further research.

First, we define some of the terms. The *geoid* is the surface along which the gravity potential is constant at all points and for which the gravity vector is normal to all points, that coincides with mean sea level extended through the continents (Figure 2.3). A *meridian* is a great circle on the surface of the Earth passing through the poles at the same longitude as the north-south reference line for the coordinate system. An *azimuth* is a horizontal angle measured through 360 degrees, from the north and reckoned clockwise from the meridian. An *ellipsoid* is a smooth closed surface whose plane sections are ellipses or circles. A *rotational ellipsoid* is the shape formed when an ellipse is rotated around one of its axes. The geometric parameters for an ellipsoid are as follows:

- Semi-major axis: a ;
- Semi-minor axis: b ;
- Flattening: $f = (a - b)/a$;
- Eccentricity: $e = (a^2 - b^2)^{1/2}/a$.

The Earth is an oblate rotational ellipsoid that is slightly flattened at the poles. There are many mathematical descriptions and coordinate systems for the Earth. To map a local geographical area accurately onto a map, many regional authorities will adopt a special regional coordinate system. The geodetic coordinates (latitudes, longitudes, and heights, described shortly) of these systems depend on the shape, size, and orientation of the reference ellipsoid. These parameters (along with others) serve as a reference or basis for the calculation of other quantities and, therefore, are called the *datum* of the coordinate system. It is desirable to have the rotational ellipsoid and the geoid match as closely as possible within the area under consideration. This explains why there are so many datums available. Some important ones are listed in Table 2.1. The full name of the datums given in Table 2.1 are listed below:

- WGS 84 stands for World Geodetic System 1984.
- NAD 83 stands for North American Datum 1983.

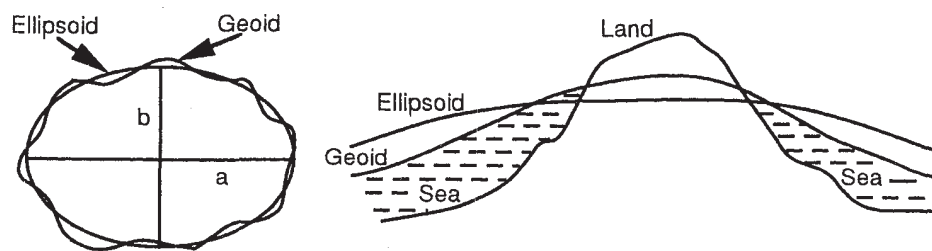


Figure 2.3 Geoid and ellipsoid.

Table 2.1
Important Datums for the Earth

<i>Datum</i>	<i>Reference Ellipsoid</i>	<i>Ellipsoid Parameter</i>	<i>Datum Origin</i>	<i>Coverage</i>
WGS 84	WGS 84	$a = 6378137\text{m}$ $f = 1/298.257223563$	Geocenter (center of the Earth)	Global
WGS 72	WGS 72	$a = 6378135\text{m}$ $f = 1/298.26$	Geocenter	Global
NAD 83	GRS 80	$a = 6378137\text{m}$ $f = 1/298.257222101$	Geocenter	Regional
ED 50	International Ellipsoid	$a = 6378388\text{m}$ $f = 1/297$	Helmert's Tower, Potsdam, Germany	Regional
Pulkovo 42	Krasovskiy 1940	$a = 6378245\text{m}$ $f = 1/298.3$	Center point of the Round Hall at Pulkovo Observatory, Russia	Regional
NAD 27	Clarke 1866	$a = 6378206.4\text{m}$ $f = 1/294.9786982$	Meades Ranch, Kansas, USA	Regional
Tokyo Datum	Bessel 1841	$a = 6377397.2\text{m}$ $f = 1/299.1527052$	Center of the transit circle at the old Tokyo Observatory, Japan	Regional

- GRS 80 stands for Geodetic Reference System 1980.
- WGS 72 stands for World Geodetic System 1972.
- ED 50 stands for European Datum 1950.
- NAD 27 stands for North American Datum 1927.

Note that the origin of the datum is a point fixed by definition and not necessarily the same as the origin of the coordinate system involved.

We will discuss two coordinate systems and one map projection: the geocentric coordinate system, the geodetic coordinate system, and the Universal Transverse Mercator (UTM) conformal map projection. Understanding these systems is important for incorporating map databases in this module as well as other modules to be discussed in other chapters. For instance, different map database vendors may supply databases defined by different reference ellipsoids or datums. The digital map database generally uses one reference ellipsoid as its coordinate reference while the positioning data received from a Global Positioning System (GPS, Section 3.4.2) receiver uses another reference ellipsoid (WGS 84) as its coordinate reference. Without proper conversions between these different coordinate systems, how can one expect the system to work as planned? Therefore, we must learn these different coordinate systems and know how to convert between them.

A geocentric coordinate system is a coordinate system with its origin at a specific point defined as the "center of the Earth," such as the center of mass (CM)

(Figure 2.4). It uses a polar (spherical) coordinate system in which the coordinates of a point are defined by the geocentric latitude ϕ , the longitude λ , and the distance r from the origin. In other words, it designates a point by means of the angles between a line running from the center of the Earth to the point, the plane of the celestial equator and the plane of a selected initial geodetic meridian. The celestial equator is an imaginary great circle on the sky, lying directly above the Earth's equator. Its plane is perpendicular to the Earth's axis of rotation and contains the Earth's CM. Because the geocentric coordinate system does not agree with the nature, it has not been used in any vehicle location and navigation system. Some literature may use the term geocentric system to describe a reference coordinate system for either a digital map or a GPS receiver, which actually means a geodetic coordinate system with the origin of the reference ellipsoid at the geocenter.

A geodetic coordinate system is a coordinate system consisting of an ellipsoid, the equatorial plane of the ellipsoid, and a meridional plane through the polar axis of the ellipsoid (Figure 2.5). The coordinates of a point in this system are given by the perpendicular distance of the point from the ellipsoid (the ellipsoidal height h), by the angle between that perpendicular (the normal) and the equatorial plane (the geodetic latitude ϕ), and by the dihedral angle between the meridional plane and a plane perpendicular to the equatorial plane and containing the normal (the geodetic longitude λ). This system designates a point with respect to the reference ellipsoid and with respect to the planes of the geodetic equator and a selected geodetic meridian. The geodetic equator is an ellipse on the reference ellipsoid midway

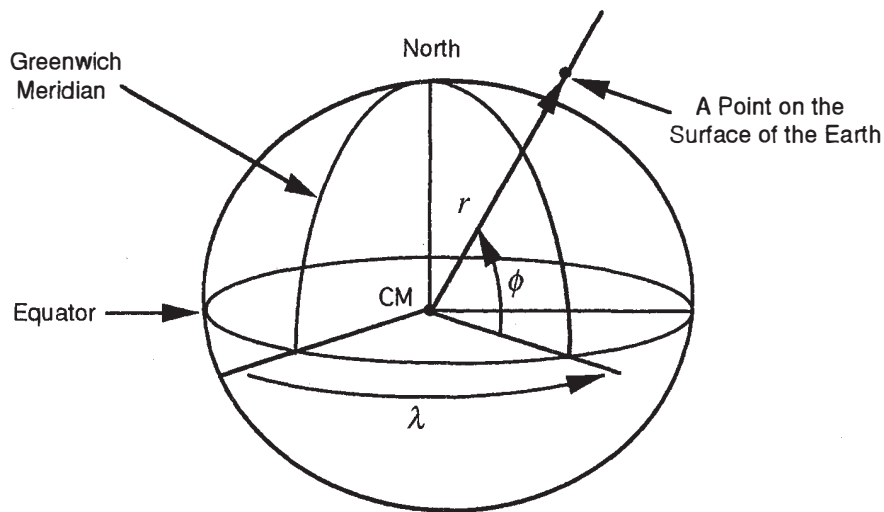


Figure 2.4 Geocentric coordinate system.

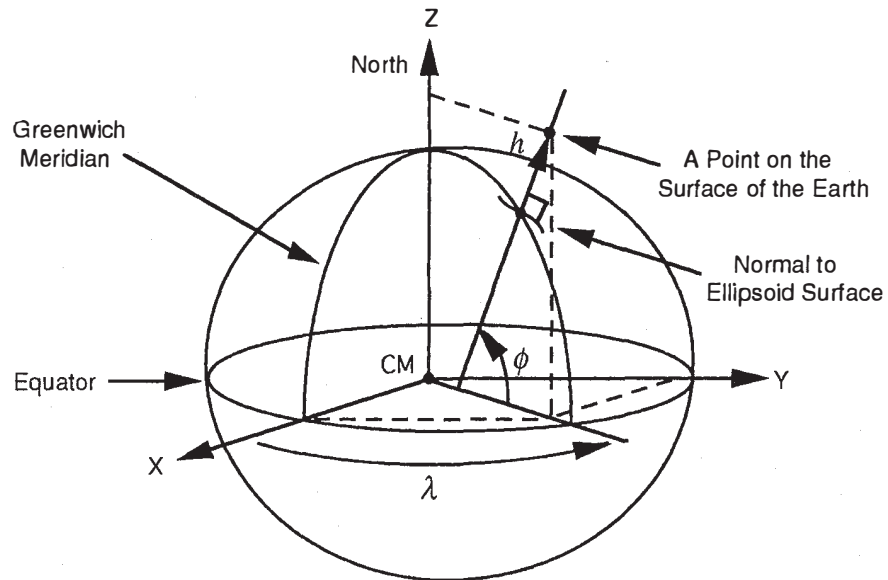


Figure 2.5 Geodetic and Cartesian coordinate systems.

between its poles of rotation. This equator is the line on which geodetic latitude is 0 degrees, and from which geodetic latitudes are reckoned, north and south, to 90 degrees, at either pole. If the minor axis of the reference ellipsoid is parallel to the rotational axis of the Earth, the geodetic equator will coincide with the Earth's equator.

Three-dimensional (3D) Cartesian coordinates (X,Y,Z) are sometimes used. One such coordinate system is shown in Figure 2.5. The dotted lines indicate the coordinates of a selected point on the Earth's surface. The main advantage of the Cartesian coordinate system is that it is completely defined by the origin, CM in this case, and the three axes. It is much simpler than the other coordinate systems. However, it is not very convenient for location and navigation. For instance, except at the North Pole, moving northeast a certain distance will not result in an identical increase in the Z value. Therefore, this type of coordinate system is rarely used in surface vehicle location and navigation. The geodetic coordinate system referenced to the WGS 84 geocentric ellipsoid (Table 2.1) is commonly known as a conventional terrestrial or Earth-centered-Earth-fixed (ECEF) coordinate system. A Cartesian coordinate system centered at CM can also be considered an ECEF coordinate system. Any point in a geodetic ellipsoidal coordinate system can be translated to a point in Cartesian coordinates provided the origin is the same. Refer to Appendix

A for the exact transformation between 3D Cartesian coordinates and ellipsoidal coordinates.

The following equation describes the relationship between geocentric latitude ϕ_c and geodetic latitude ϕ_d (Figure 2.6).

$$\tan \phi_c = (1 - e^2) \tan \phi_d$$

From Figures 2.4 and 2.5, we see that the geocentric longitude and geodetic longitude are identical.

Many geodetic coordinate systems are not geocentric, that is, their origins do not coincide with the center of the Earth. This is because offsets may improve the match between the surface of the ellipsoid and the geoid in the relevant region. Latitudes and longitudes referenced to different ellipsoids are different coordinate systems and cannot be mixed. If mixed, this may lead to a displacement of up to 1,500m for a point on the Earth's surface depending on the datum, reference ellipsoid, and the map projection used. This explains why the output of a GPS receiver may not necessarily match a certain map.

Geodetic coordinates are commonly used in modern location and navigation systems. There are three problems with geodetic coordinate systems (as discussed in [1]). First, there are a wide variety of different geodetic systems (Table 2.1). When switching between different systems or integrating portions of a vehicle location and navigation system that use different coordinate system, care must be taken to ensure that a proper transformation is done. Second, geodetic coordinates are inconvenient for 3D air and space navigation, because the third dimension (height above the reference ellipsoid) provided by this coordinate system is different from the height

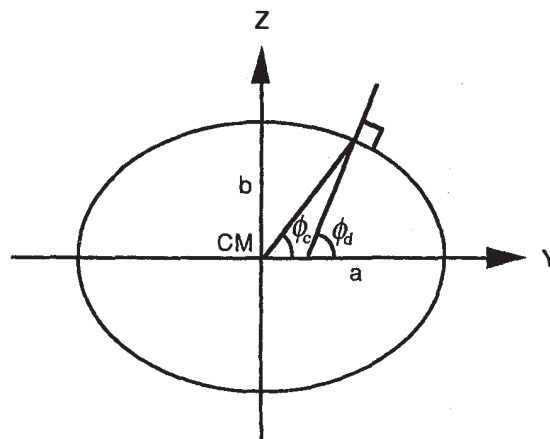


Figure 2.6 Geocentric and geodetic latitudes from the cross section of the Earth.

above the geoid and other heights relative to different reference ellipsoids. Third, the latitude and longitude cannot be used straightforwardly in a location and navigation system based on geodetic coordinates. Although exact or approximate mathematical formulas can be used to calculate the distance and azimuth, they are not as straightforward as plane trigonometry in 2D Cartesian coordinates. These considerations led to the development of the third method described below, in which the distance and azimuth between two points of known coordinates can be computed easily by plane trigonometry.

To transform geodetic latitudes and longitudes into a planar Cartesian coordinate system, projection equations are used. One valuable projection is the UTM projection. It is often used for map projections involving large countries or continents because a line of constant azimuth is represented by a straight line in this projection.

When integrating a GPS receiver with a dead-reckoned positioning module (Section 3.2) or with a map-matching (Chapter 4) module, one often needs to convert the WGS 84 coordinate system used in the GPS receiver to a UTM coordinate system, so that all relevant observations can be aligned to the surface of the Earth.

$$x = f_1(\phi, \lambda)$$

$$y = f_2(\phi, \lambda)$$

where x and y are the northing and easting in the UTM projection, respectively, and ϕ and λ are the latitude and longitude in the geodetic coordinate system, respectively.

The UTM projection is coupled with a certain datum that references a geodetic coordinate system. We can transform geodetic coordinates referenced to WGS 84 directly to coordinates in a UTM mapping plane referenced to any ellipsoid. (The 2D transformation equations can be found in Appendix B.) Alternatively, a simple square root equation can be used to calculate the distance between two points in any coordinate system with latitude and longitude as their axes, since the travel area for a particular vehicle location and navigation system is often relatively flat, even though the Earth's surface is curved. The errors of calculation are negligible only if the region is not too big, as is typically the case for most metropolitan areas. However, any such computation must be scaled properly, that is, we must multiply by the appropriate number of meters per degree along each axis. We now discuss the basic principles of the UTM projection.

Representation of the 3D spherical Earth on a 2D mapping plane requires a transformation or projection. Either the angles are preserved (conformal projection, where any angle on the map is exactly the same as on the Earth), or the areas are preserved (equivalent projection, where the ratio between any area on the map and the corresponding area on the Earth is constant). There is no projection that can preserve both.

The Mercator projection is a conformal map projection based on a cylindrical projection. In this projection, the equator is a straight line true to scale, the meridians are straight lines of equal length, and the parallels are straight lines of unequal length. The cylinder is tangent to the Earth along the equator. Turning the cylinder 90 degrees, we have the transverse Mercator projection (Figure 2.7), where the central meridian is a straight line true to scale, and meridians 90 degrees from the central meridian and the equator are also straight lines. Other meridians and parallels are complex curves. Imagine that by wrapping a cylinder around a globe, the cylinder is tangent along a meridian, or secant along lines parallel to this meridian. The meridian and parallel lines are conceptually placed on the cylinder. When the cylinder is cut along a line perpendicular to this meridian, unrolled, and laid flat, the projection is obtained. The ellipsoid (or Earth) is partitioned into zones of equal longitude where the central meridian is in the center of each zone.

The UTM projection is a specific transverse Mercator projection. First, the ellipsoid is partitioned into 60-degree zones, each of which is 6 degrees wide in longitude. Second, a scale factor of 0.9996 (k in Appendix B) is applied to the 2D coordinates in order to avoid large distortions in the outer areas of each zone. As noted earlier, UTM coordinates in the north and east directions are referred to as the "northing" and "easting" directions (Figure 2.7). Each zone is divided into smaller squares where the coordinates are given in meters referred to the bottom left boundary of each individual square. In other words, the UTM coordinates are based on a system of zones of longitude, each with its own central meridian. Although the UTM coordinate system makes the life of vehicle location and navigation engineers easier, navigation across UTM zones is difficult. For more information on Earth coordinate systems and map projections, refer to [1-3].

The UTM projection was adopted by the U.S. Army Map Service in 1947 for their use in worldwide mapping. Many of the paper maps used in the United States

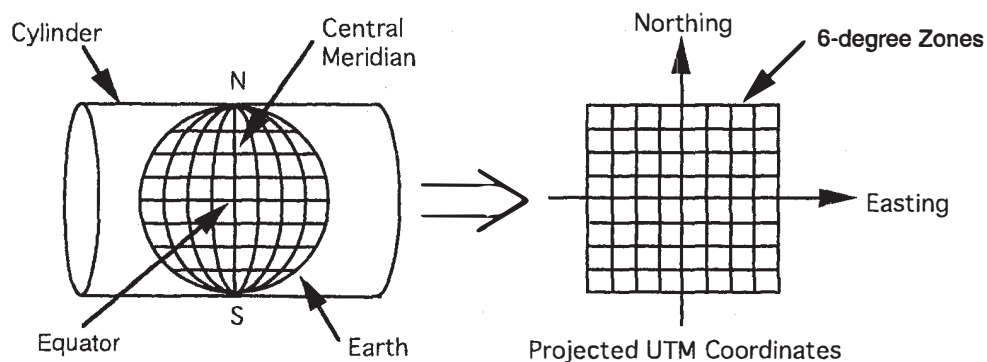


Figure 2.7 UTM projection.

for local surveying and other mapping operations are also based on transverse Mercator projections. For projection methods used by individual states in the United States, the reader may refer to [4].

There are no widely accepted standards available for digital maps. It is very common for a map database vendor and its customers to agree on a software interface standard. In this way, the interface provides flexibility in the storage and retrieval of data. In the following sections, some of the emerging map database standards are discussed. Two proprietary digital maps are presented as examples of how software interface approaches are used. An introduction to the map compilation process is presented in the last section.

2.4 STANDARDS

Recently, the International Organization for Standardization (ISO)/TC 204 has approved a new task on an application program interface (API) for digital map databases, in addition to the existing task to create a map database physical storage format (PSF). Some believe that this will lead to an open global standard in digital map databases. One difficult work lying ahead is to make sure that the industry will actually use these standards once they are developed. The API task group held its first meeting in December 1996. It plans to create a standard method for navigation software to link with different data formats used by different digital map database vendors. The PSF task group aims to standardize the location of data on the storage medium so the system knows where to look for it. In short, the purpose of these activities is to provide a standard mechanism for navigation software to interact with map databases stored on different media as mentioned in Section 2.6.1. These standardization activities will certainly boost the market for vehicle location and navigation systems.

2.4.1 Geographic Data Files

The first version of the draft Geographic Data Files (GDF), which is a product of the EUREKA project DEMETER undertaken by Philips and Bosch in Europe, was released in October 1988. This standard was designed to specify the data content, the means by which data representation and structure of data supplied for vehicle navigation systems. The second version of the GDF (GDF 2.0) was developed under work package 3532 of the European Digital Road Map (EDRM) Task Force. The GDF 2.0 Working Group consisted of representatives of EDRM Task Force partners Daimler Benz (prime contractor), Bosch, Philips, Renault, Tele Atlas and Intergraph, and MVA Systematica, a participant in the PANDORA project. The partners in the PANDORA project are Philips, Bosch, the Ordnance Survey, and the Automobile Association. GDF 2.0 differs from GDF 1.0 both in its treatment of the architecture

of certain data structures and in its use of terminology. It is also a far more extensive specification. It defines standards for describing, classifying, and encoding features of the road environment, which will support a broad family of application areas.

GDF 2.0 includes the following content: The Feature Catalogue provides a clear definition of "real-world objects" such as roads, buildings, administrative areas, and settlements that are the concern of the broad area of application for the road environment. The Attribute Catalogue defines the characteristics of each feature. Dedicated attributes of particular features are cross referenced. The Relationship Catalogue describes relations between features that can be used to convey information in a realistic manner. The Feature Representation Scheme specifies how real-world objects should be represented in terms of GDF logical structures. The Data Content Specification makes clear which data are required for different applications. The Global Data Catalogue classifies geodetic requirements and other information related to the geometric and geographic characteristics of the data. Finally, the information needed to supply GDF 2.0 data as physical data records is provided in the Media Record Specification.

GDF 2.0 has three major parts: First, the specification of data content (SDC) part lists the information required for navigation systems. Second, the specification of data acquisition (SDA) part describes a set of features, attributes, and relationships for representation of the nonspatial aspects of information. Third, the exchange format (EF) part defines the records and fields.

One important aspect of this standard is the data model in which features, attributes, and relationships are formally defined, along with the concept of a *layer*. Layers are groupings of features that are strongly related such as roads and ferries, administrative areas, settlements, buildings, bridges and tunnels, railroads, and waterways. For example, roads consist of features such as road elements, intersections, and the road itself. In turn, each feature has attributes. For instance, the road element has attributes of traffic direction, name, class, speed limit, impedance value, etc. Relationships represent driving-related information and are a function of several features such as posted sign information and prohibited turns. The geometry and topology of the road network and related geographic features are described in terms of geodetic coordinates, relative location to neighboring features, and shape where appropriate.

After the release of GDF 2.0, GDF 2.1 was released in October 1992, GDF 2.2 was released in November 1994, GDF 3.0 was submitted to Comité Européen de Normalisation (CEN) or the European Committee for Standardization TC 278 for official approval as ENV in October 1995 [5]. The main changes contained in these new versions involve introduction of the concept of a segmented attribute and the concept of time domain, of new information items, and the incorporation of review comments. The revisions and refinements will continue to meet international requirements. There are also several national standards for map transfer, which are moving toward compatibility with the international version of GDF. Some anticipate

that the ISO version of GDF will form the foundation for a single international family of transfer standards for ITS map data [6].

GDF is an interchange database standard. Users must develop their own compilers to translate it into *navigable* databases. A navigable database is a database that provides information which can be used to navigate a vehicle through a road network. For instance, in order to provide turn-by-turn route guidance, the database must contain some important attributes such as one-way streets and turn restrictions. Most digital map database suppliers can provide GDF maps for their European customers. Major companies currently involved in the production of map databases are Etak, Inc., Intergraph Corporation, Navigation Technologies Corporation, Tele Atlas International BV, etc.

2.4.2 Digital Road Map Association

The Japan Digital Road Map Association (DRM or JDRMA) was established by the Public Works Research Institute of the Japanese Ministry of Construction in 1988 [7,8]. Their product is both a digital map standard and a map database. The association supplies digital road map databases to member companies on magnetic tapes. DRM started digital map database production using 1/50,000 scale topographical maps and is now in the process of refining the database using 1/25,000 scale topographical maps, which offer more detailed and precise features. The topographical maps are issued by the Geological Research Institute and the Ministry of Construction. In addition, they have also used other urban planning maps and information provided by road administrators. They use custom sources, but project all the data onto a reference ellipsoid based on the Tokyo datum. This datum is defined by the location and azimuth on the 1841 Bessel spheroid, with the origin of the datum at the center of the transit circle of the old Tokyo Observatory (Table 2.1).

Three design principles are used by DRM. First, the location and configuration of features on the map are measured on two topographic maps (1:25,000 and 1:50,000) and filed according to a standard regional grid and mesh code system. Second, separate links are established for the road network indicating road configuration, interchanges, and ramps and lanes, with route searching and map matching being integral parts of the system. Third, the digital map consists of two layers of road networks, with the basic road network (main roads) used for the best route computations and the full road network for detailed operations such as map matching and zoom-in graphic displays.

The two layers of the road network are defined as follows. The basic road network consists of the national highways, prefectural highways, and roads 5.5m or wider (two lanes or more). The full road network consists of all roads covered by the basic road network plus all roads 3m or wider (one lane and more). Location is expressed using a mesh code number (with each mesh covering an area of

approximately 10×10 km) and normalized coordinates (X: 0-10,000, Y: 0-10,000) within each mesh. The road network is defined in terms of a combination of nodes and links (similar to the segments defined earlier). The basic road network uses four-digit node numbers, while the full road network uses five-digit node numbers. Each link number is a combination of the numbers for the nodes at each end. In this way, all roads can be identified by mesh and link numbers. The location and shape of each link is expressed by a series of coordinate values for the interpolating points. The database is organized as a series of files.

DRM has already covered all of Japan and version 3.0 of the map has recently been released. DRM provides the map to their users on magnetic tapes and its data structure is open to public. However, the file format, which is the key to reading the magnetic tapes properly, is only open to DRM users. The DRM does not provide any database interface to users who need to design and implement their own interface to access the map database. Furthermore, DRM does not contain information such as turn restrictions or one-way streets for route planning and guidance. Users need to further expand the database or include smaller scale information (1:10,000) for vehicle navigation.

2.4.3 Spatial Data Transfer Standard

The Spatial Data Transfer Standard (SDTS) was begun by the U.S. National Committee for Digital Cartographic Data Standards in 1982. It is designed for transfer of a wide variety of data structures that are used in the spatial sciences. These sciences include cartography, geography, geology, geographic information systems (GIS), and many other related sciences. In 1985, the Standards Working Group of the Federal Interagency Coordinating Committee on Digital Cartography also began work on spatial data exchange standards. During 1987, the results of these parallel efforts were merged by the Digital Cartographic Data Standards Task Force into the proposed Digital Cartographic Data Standards, published as a special issue in 1988 [9].

SDTS is an open and general standard for the exchange of spatial data [10]. It is also a Federal Information Processing Standard (FIPS). Effective February 1994, U.S. federal agencies are required to provide data to other federal agencies and the general public in SDTS format. SDTS consists of three primary parts:

- The first part is the SDTS logical superstructure, which describes the organization and structure of the SDTS transfer mechanism.
- The second part prescribes definitions for spatial features and attributes.
- The third part describes the implementation of the ISO 9211 data transfer standard (i.e., encoding method).

SDTS intends to provide a solution to the problem of transferring spatial (i.e., geographic and cartographic) data from the conceptual level to the details of physical

file encoding. Transfer of spatial data involves modeling spatial data concepts, data structures, and the logical and physical structure of the files. To be useful, the data to be transferred must also be meaningful in terms of data content and data quality. SDTS addresses all of these aspects for both vector and raster data structures.

The ITS America (Intelligent Transportation Society of America) Map Database and Information Systems Subcommittee is working on a vector profile for SDTS, which would be a standard for the North American highway database. This project is being worked on in conjunction with the Federal Highway Administration (FHWA).

2.4.4 Truth-in-Labeling Standard

A labeling standard, the Truth-in-Labeling Standard, has been completed after several years of work by the Map Database Standards Committee of the Society of Automotive Engineers (SAE) [11]. The purpose of this standard is to define consistent terminology, metrics, and tests for describing and comparing the content and quality of navigable map databases for vehicle location and navigation. The standard will allow users to determine whether a given map database meets the needs of an application. However, it does not specify minimum performance standards or physical database format as the other standards do.

The truth-in-labeling standard contains definitions and detailed descriptions of map database attributes, as well as metrics and tests for each map database object (i.e., physical representation of a concept such as "node" or "segment" in a digital map database) defined. Note that this standard uses "link" in place of "segment." The objects defined include the following:

- Nodes and junctions;
- Location and existence of attached links;
- Road names;
- Address ranges;
- Road classifications;
- Restricted maneuvers.

The present focus of this standard is on in-vehicle navigation applications such as address location, route determination, route guidance, vehicle positioning, and display. Later versions of the standards during the next 3 to 5 years will also be applicable to other ITS map-related applications. In addition, SAE plans to establish a testing institute to test database labels using this standard. Database vendors will specify the content, currency, accuracy, coverage, etc., of their databases in the format specified by the standard. The institute will verify their claims. Based on the verification results, users will be able to quickly and accurately select the database best suited for their applications. As a result, users can determine the usability of a

particular database for their applications in advance, rather than finding out later, after the shipment of the database product. In short, the standard allows users with different needs to select the most appropriate database for their applications.

Despite these standards activities, most location and navigation products for automobiles in the United States are based on proprietary digital map databases provided by private companies. The next section covers this topic in detail.

2.5 PROPRIETARY DIGITAL MAP DATABASES

2.5.1 Etak

The EtakMap databases developed by Etak, Inc., use many sources, and project everything onto a two-dimensional representation of the Earth's reference ellipsoid [12]. The sources used include aerial photography, U.S. Geographical Survey (USGS) maps, local government maps, the U.S. census bureau's TIGER '92 database, cadastral maps, tax assessor information, cross reference directories, ZIP+4 files, field data capture information, customer files, and map reports.

- The USGS produces Quad Sheets, which are accurate 7.5- by 7.5-min contour maps of the United States. Many these maps are 20 to 30 years old and lack highway information.
- TIGER is a standard for vector-encoded data in which features are represented using Cartesian geometry. Features are modeled using combinations of primitives such as points, lines, and areas. TIGER maps were developed by the Census Bureau for taking Census data. They are more complete than USGS maps but they lack accuracy, geometry, and highway data.
- Cadastral maps are maps showing the boundaries of subdivisions of land, usually with bearings and lengths, and the areas of individual tracts, for purposes of describing and recording ownership.

In North America, the reference ellipsoid is based on the North American Datum of 1927 (NAD 27) defined by the location and azimuth on the Clarke spheroid of 1866, with origin at Meades Ranch, Kansas. Clearly, NAD 27 is a geodetic coordinate system. NAD 27 is a horizontal control datum specifying the coordinate system in which coordinates of points on the Earth can be calculated using a set of constants.

As mentioned before, Etak provides interface formats for users to access their map databases. The MapBase format is an ASCII format compatible with the major GIS formats. The MapAccess format is based on Etak 0-track, a proprietary method that relies heavily on the topology of the map, in particular, the carrier topology of the map, rather than geometric coordinates (see Table 2.2). It is said that faster map

Table 2.2
Sample Data Structures for Digital Map Databases

<i>Data Structure</i>	<i>Principle</i>	<i>Usage</i>	<i>Comment</i>
B-trees	Partition data into a hierarchical index tree in which each node occupies a block of memory.	Text indexing	Conventional indexing
Carrier-block-trees	Partition data into carrier blocks, each one is the smallest topological closed set.	Spatial indexing	Used by Etak
k-d-trees	Partition data recursively into a binary search tree where odd and even levels have different keys to determine branching directions.	Spatial indexing	Popular in GIS but difficult to update compared with quad-trees
Oc-trees	Partition data recursively into octants, with possible duplications for features crossing octant boundaries.	Spatial indexing	For 3D data
Quad-trees	Partition data recursively into quadrants, with possible duplications for features crossing quad boundaries.	Spatial indexing	Very popular in GIS for 2D data
R-trees	Partition data recursively into nested rectangular areas, which may overlap.	Spatial indexing	More flexible than quad-trees or k-d-trees

data retrieval can be achieved using this method [12]. It is available only by using MapAccess development tools (C language libraries) to retrieve and display maps and geocode addresses. *Geocoding* is the method of applying a latitude and longitude coordinate to an address. Other formats are also available for users with different applications and interface requirements.

EtakMap premium maps are offered in standard and enhanced versions. The enhanced maps will eventually cover 80% of the U.S. population. These maps include address, zip code, road classification, graphic features, political and statistical geography, major landmarks, one-way street encoding for selected urban areas, data overlay attributes, and compliance with National Map Accuracy Standards. In other words, these enhanced maps are navigable. They provide road networks covering more than 43 million metropolitan people. By 1998, Etak plans to have a map database covering major metropolitan areas with a total population of 130 million people. In addition, its standard maps will fill the rural gaps to provide 100% national coverage. These maps include only address and zip code information and are therefore not navigable.

Etak overseas coverage includes Japan, Hong Kong, and Europe, some with complete coverage and some without. Etak has entered into a data-sharing agreement

with Tele Atlas BV, a new joint venture formed by Robert Bosch GmbH and Tele Atlas International BV. Etak is focusing its efforts on mapping Great Britain and the United States, and Tele Atlas BV is continuing to map continental Europe, including Switzerland and Austria.

2.5.2 Navigation Technologies

The databases developed by Navigational Technologies Corporation (NavTech) also use many sources, and project everything onto a two-dimensional representation of the Earth's reference ellipsoid. The basic database concepts by NavTech contained in NavTech's early interface standards include database content, data organization, data relationships, and data access methodology [13]. Their sources include topographical base maps, aerial photography, and local ground research (field data), as well as information from a variety of additional sources. Each database area is continuously updated and supported by a local field staff and a global customer support organization. Information in the database includes road network geometry, roadway classification, roadway characteristics such as directionality, turn and time-of-day restrictions, cartographic and geopolitical boundaries, and points of interest, landmarks, and services. NavTech databases come with a written warranty of at least 97% accuracy and completeness based on "ground truth." Its guaranteed precisions are within 15m of ground truth in detailed city areas and 100m in areas covered by intertown maps.

In North America, the reference ellipsoid mentioned earlier is based on the North American Datum of 1983 (NAD 83) as defined in the Geodetic Reference System 1980, with its origin at the center of the Earth. From Section 2.3, we know that NAD 83 is a geodetic coordinate system based on a geocentric reference ellipsoid.

The NavTech databases contain information on the road network and environment including roads, intersections, and points of interest [13]. The data are logically organized so that certain information can be accessed and manipulated, and physically organized in such a way as to provide maximum flexibility. The data relationships include a variety of relationships between data entries in the same layer and zone, which means that road names can be superimposed on the correct roadway segment. The data access methodology involves a transparent user interface, which resides between the application and the database itself, thereby allowing users to make high-level logical requests without having to know the structure of the database.

The early NavTech proprietary map-data interchange interface standard, the Standard Interchange File (SIF), consists of a set of record types, in which each record type contains a portion of the database for navigation and digital cartography. Depending on the custom request, either all record types or only a subset of the record types can be created. Record types can be divided into two groups: interchange file (data description directory) records and link records (link information).

Interchange file records contain information about the data format and meaning of each field. Link records contain detailed information about each link. A NavTech link is a specific segment or subsegment in the road network or an area/polygon that is a physical feature or administrative boundary. The interchange file is primarily link oriented and all information about each link is grouped. NavTech requires user applications to connect the endpoints of intersecting links.

In the United States, NavTech has navigable digital databases available for numerous major metropolitan areas. NavTech has completed detailed city coverage for a total area containing more than 100 million people and intertown map databases for less populated areas. These intertown map databases include all roads that are contained on American Automobile Association (AAA) state maps. By the year 2000, NavTech plans to have a map database covering an area with a total population of more than 210 million people.

In Europe, NavTech Europe (formerly European Geographic Technologies (EGT)) has completed a database for more than 80 cities in Germany, the greater Paris area, Lyon, and Marseille in France, Torino and Milano in Italy, and Coventry in the United Kingdom. Additional cities throughout France, Italy, the United Kingdom, and Germany have also been completed. In addition, intertown map databases now include all roads connecting municipalities and settlements larger than 250 km².

2.6 DIGITAL MAP COMPILATION

As mentioned, a digital map database can be encoded from existing nondigital maps or aerial photographs. The coordinates of nodes are usually encoded using special computers that can record the coordinates of any given point. The conversion or digitizing is done for a point by placing the crosshair of an instrument over the point and pressing a button. In some cases, paper maps are scanned and then converted to vector form by software. Once the basic road geometry has been encoded and the road topology has been checked, the database is further expanded by entering attributes such as road types, road names, speed limits, lengths, turn restrictions, and other details as discussed earlier. Gradually, all these data form an intricate data mesh, a virtual reality of the road network. Although the whole process has been automated to varying degrees, it still requires tedious human involvement and intervention. For instance, one quad sheet for San Antonio, California, took a week to input into a vendor's database. Although a section of the database can be as accurate as the sources, field engineers may still need to actually drive vehicles on the road to verify and update the map for final completion. The final product typically takes the form of raw cartographic data files. The accuracy of these data files is very important to systems relying on map-related functions. Significant errors (greater than 15m in city areas) may lead to poor overall system performance.

Besides providing raw cartographic data files in one of the interchange formats, all the map database vendors also compile their databases in some internal format.

Users can use their map retrieval and drawing libraries to easily access the databases. At this writing, none of the database vendors provides compatible and independent formats for customers. Most location and navigation product companies compile their own navigable databases (files) based on the raw interchange cartographic database files provided by vendors as shown in Figure 2.8. Besides the vendor database files, additional files collected from other sources can also be used as inputs for the map compiler. One example is discussed in Section 6.4.

2.6.1 Data Structures

Many different data structures and implementations could be used to store the digital map database [14–16]. Table 2.2 lists several popular data structures, some of which were summarized in [12]. It is not uncommon for a simple map database to use a combination of these and other data structures. The database can be stored either in the main memory of the location and navigation computer or in a secondary storage area such as a hard disk, a CD-ROM, or a PCMCIA memory card (or PC card). Typically, around 10 to 100 MB are required to store an uncompressed metropolitan area map; the size of the map is a function of the data structures selected, the complexity of the attributes, the geographic area covered, the road topology, and other factors. Compression techniques can be used to reduce the size of the map, with the side effect that decompression will be required for applications to access the map database; this often requires additional design considerations and the development of proper mechanisms for dealing with real-time applications. Depending on the system implementation, the user may design a database manager that will provide services to a set of applications requiring access to the database.

The data structures shown in Table 2.2 typically contain, at a minimum, node records and segment records. As an example, the following list contains three possible records and their attributes:

- *Node (point) record*: node ID, latitude, longitude, connectivity, etc.;
- *Segment (line) record*: segment ID, left node, right node, length, speed limit, directionality, address range, road name, city name, state name, zip code, shape points, road type (classification), drivability, etc.;
- *Area (polygon) record*: area ID, area name, city range, city name, state name, zip code, etc.

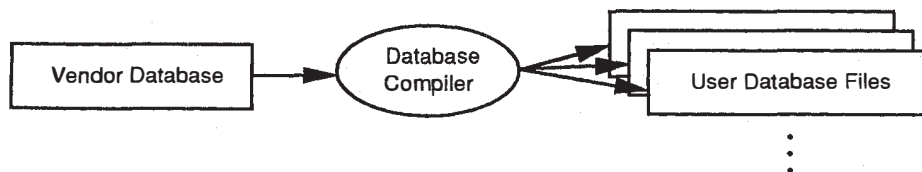


Figure 2.8 Database compilation.

2.6.2 Compiler Structure

The database compiler has a structure identical to that of a conventional language translator. At the top level it consists of two parts: a front end and a back end. The purpose of the front end is to read the vendor data in the interchange format, analyze it, and output the results of this analysis as a set of intermediate files. The analysis process includes determining the connectivity of the road network, finding unique names of objects like streets, cities, countries, points of interest, etc. The purpose of the back end is to take the intermediate files and encode them efficiently as navigable databases (database files) for storage and retrieval by an application or another module. The encoding process includes the creation of B-tree indexes for retrieval of names of streets and cities, the creation of blocks of data containing spatially correlated objects (parcels of land), and the creation of indexes for access to these objects (quad trees, etc.).

If handled properly, the compiler structure and its compilation process should have no effect on map quality. As mentioned earlier, mixed reference coordinate systems in a map-based system can lead to systemic errors as large as 1,500m for every point on the map. On the other hand, errors in the data files themselves will not have such consistency and often occur randomly in several different attributes. For instance, one node on a street might be off 20m of "ground truth," one local street might be missing, a one-way street might be marked as two-way, a particular road might be classified incorrectly, and so on. Such errors frequently occur before compilation of the database and will adversely affect performance of a location and navigation system.

2.6.3 Hierarchical Maps

To efficiently solve the complex problem of vehicle location and navigation, the system should at first ignore low-level details and concentrate on the essential features of the problem, filling in the details later. This idea easily leads to generalization, that is, multiple hierarchical layers of abstraction, each focused on a different level of detail. This technique has proven to be very effective in reducing the complexity of large problems (see Section 5.5 for detailed analysis).

As an example, a hierarchical map database and compiler developed by Motorola are briefly discussed. A key feature of this database is that the segments stored in the higher layer are equivalent to a combination of several segments stored in the layer immediately below. Notice that this storage method leads to a considerable reduction in the number of nodes required to represent the exact same road network in a higher layer. As we will learn from the running time analysis of route-planning algorithms in Chapter 5, this reformatting of the road data to create combined road segments for the higher layer results in improvement of the route planning running

time because there will be fewer nodes that correspond to segments in the higher layer. The number of nodes to be visited during the planning process is a major factor that affects the running time for the algorithms. An alternative method is to construct a database in which the lower layer segments are reproduced almost exactly in the higher layer. Unlike the previous method, no combination of the low-layer segments is required. Because this method simply removes certain local roads when moving up to the higher layer, the remaining road segment data will be reproduced exactly from the lower layer to the higher layer. This will require more memory, but reduces the compilation time and makes the database design less complicated. However, route planning is relatively time consuming using this type of database, because there is no substantial reduction in the number of nodes visited during the decision-making process.

The Motorola hierarchical map database is a collection of layers. The full set consists of all four layers, layer 0 (the bottom layer), layer 1, layer 2, and layer 3.

- Layer 0 includes all the roads in the network and related navigation information.
- Layer 1 includes collector roads, arterials, and highways.
- Layer 2 includes arterials and highways.
- Layer 3 includes highways only.

Collector roads are semi-major roads, which may lead to arterials. Although a four-layer map is used, the database compiler can support up to eight layers. A graphical representation of a four-layer sample map is illustrated in Figure 2.9. It is possible to consider maps that are composed of a single layer, such as layer 2 only, or multiple layers such as layers 1 and 2. Such maps would cover large areas and contain fewer segments, thus requiring less storage space than fully populated maps. These maps are direct analogs of the country and state maps that we use when traveling from city to city.

As discussed before, the database compiler, with its front-end and back-end structure, compiles the raw cartographic data into navigable hierarchical data files that include map parcel files and index files. These files can then be stored on CD-ROM as shown in Figure 2.10, for access by other modules via the application interface (API). A parcel is analogous to a portion of a map area, which is a collection of geographical objects (such as segments) falling within a rectilinear region defined by minimum and maximum latitudes and longitudes. The map index data help to locate multiple maps stored on the CD-ROM (such as maps for different states). The map parcel data contain detailed information on the map for each area. Index data are used to locate the parcel which stores vector-encoded map representations. A similar CD-ROM storage method has been used by Philips [17] and others [18,19].

Another feature of the Motorola hierarchical database is the addition of a dynamic database (see Figure 2.10). The dynamic data are used to store dynamically updated information for each *link* received from a traffic information center. In this

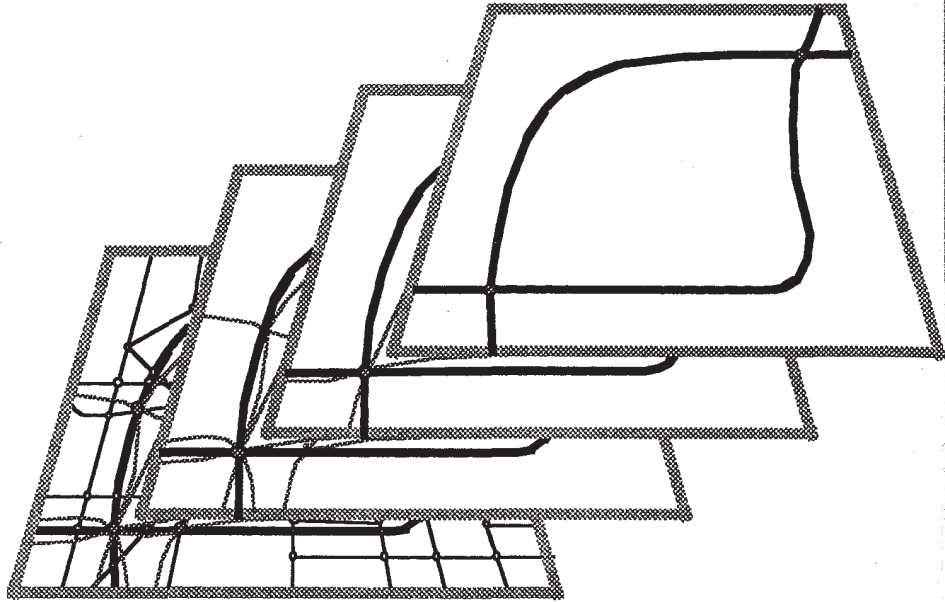


Figure 2.9 Hierarchical map database.

terminology, link refers to a segment with turning delay information. A turning delay is the normal time required for a vehicle to finish a turn associated with a particular segment. Note that this definition of a link is different from the one used in our discussion of the DRM and NavTech databases. Depending on the number of turns ahead, a specific segment may need more than one link to distinguish the delays for different turns. The database module and its application modules can incorporate these real-time updates for each segment and use the turn delays for specific applications. Applications for the dynamic database are discussed in further detail in Section 6.4.

One of the key benefits for developing a hierarchical map database is to support hierarchical route-planning algorithms in an environment where a slow secondary database storage device such as a CD-ROM is being used. All modules work at layer 0, except for the route-planning module. The database has detailed data at layer 0 and minimal data at other layers. In this way, for route planning, the system can fetch data that cover a larger area during a single secondary-memory read for the higher layer than for the bottom layer. Section 5.5 will contain an additional discussion of the benefits of hierarchical map databases for route planning. Another application of hierarchical storage is in map display. Depending on the scale of the map requested by the user, the map display application will select the appropriate

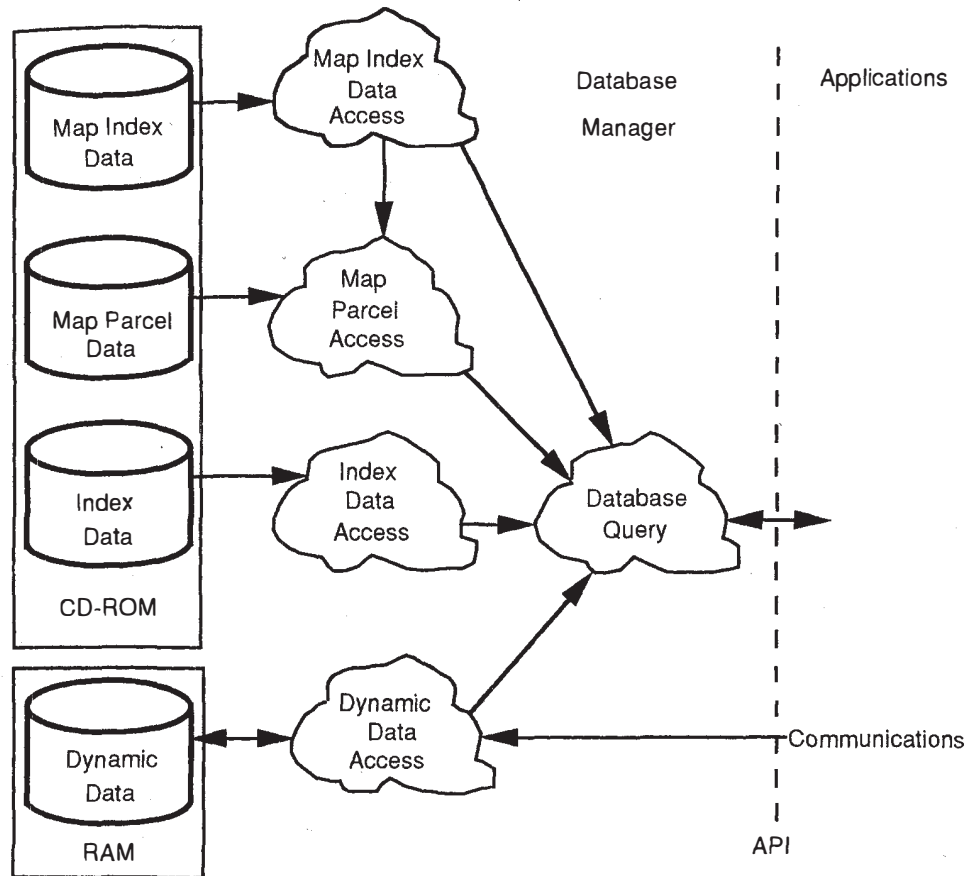


Figure 2.10 Database access.

map layer needed to draw the map. This will reduce the amount of data read from the CD-ROM and reduce the processing time required to filter unwanted information.

In summary, digital map database design should take into account the following four considerations: content, organization, relationships, and access methodology. In other words, a vehicle location and navigation system needs a map database that can support map matching, route planning, route guidance, map display, and the display of associated points of interest. To compile a vendor database into a customized database (data files), database compiler designers must consider the specific application domains involved. For vehicle navigation systems, a navigable database

should at a minimum provide accurate map-related location information, the attributes required to calculate a best route and to guide the vehicle, as well as road-related facilities and information. The database must be easy to access. For an on-board navigable database, the run-time access speed must be carefully addressed.

Before turning to the next chapter, let us consider how street addresses can be correlated with digital map coordinates (geocoding) assuming that the map does not contain the coordinates of every possible address. Given the map database concepts we learned in this chapter, how can we derive the coordinates for a known address?

References

- [1] V. Ashkenazi, "Coordinate Systems: How to Get Your Position Very Precise and Completely Wrong," *The Journal of Navigation*, Royal Institute of Navigation, Vol. 39, No. 2, May 1986, pp. 269-278.
- [2] L. M. Bugayevskiy and J. P. Snyder, *Map Projections: A Reference Manual*, Bristol, PA: Taylor & Francis, 1995.
- [3] J. P. Snyder, *Map Projections—A Working Manual*, U.S. Geological Survey, 1987.
- [4] A. Leick, *GPS Satellite Surveying*, 2nd ed., New York: John Wiley & Sons, 1995.
- [5] European Committee for Standardization (CEN), *Geographic Data Files*, Brussels: CEN, Oct. 1995.
- [6] R. Pearlman and S. Scott, "IVHS Map Database Transfer Standards: Current Status," *Proc. IEEE-IEE Vehicle Navigation and Information Systems Conference (VNIS '95)*, July 1995, pp. 368-373.
- [7] Japan Digital Road Map Association, *Digital Road Map Database Standards*, Version 2.0, Jan. 1991.
- [8] M. Shibata and Y. Fujita, "Current Status and Future Plans for Digital Map Database in Japan," *Proc. IEEE-IEE Vehicle Navigation and Information Systems Conference (VNIS '93)*, Oct. 1993, pp. 29-33.
- [9] J. Morrison, "Proposed Standard for Digital Cartographic Data," *The American Cartographer*, Vol. 15, No. 1, Jan. 1988, pp. 9-140.
- [10] U.S. Geological Survey, *Spatial Data Transfer Standard*, 1992.
- [11] Society of Automotive Engineers, *Truth-in-Labeling Standard for Navigation Map Databases*, SAE J1663, Aug. 1995.
- [12] M. S. White, "Digital Maps—A Fundamental Element of IVHS," *IVHS Journal*, Vol. 1, No. 2, 1993, pp. 135-150.
- [13] D. C. Marsh, "Database Design, Development, and Access Consideration for Automotive Navigation," *Proc. IEEE-IEE Vehicle Navigation and Information Systems Conference (VNIS '89)*, Oct. 1989, pp. 337-340.
- [14] O. Gunther, *Efficient Structures for Geometric Data Management*, Berlin: Springer-Verlag, 1988.
- [15] H. Samet, *The Design and Analysis of Spatial Data Structures*, Reading, MA: Addison-Wesley, 1990.
- [16] M. S. White and G. E. Loughmiller, "Apparatus Storing A Representation of Topological Structures and Methods of Building and Searching the Representation," *Canada Patent No. 1277043*, Nov. 1990.
- [17] M. L. G. Thoone, "CARIN, A Car Information and Navigation System," *Philips Technical Review*, Vol. 43, No. 11/12, Dec. 1987, pp. 317-329.
- [18] H. J. G. M. Benning, "Digital Maps on Compact Disc," *SAE Paper No. 860125*, Society of Automotive Engineers, 1986.
- [19] K. Ishikawa, M. Ogawa, T. Tsujibayashi, and Y. Shoji, "Digital Map on CD (Called CD Information)," *SAE Paper No. 880221*, Society of Automotive Engineers, 1988.

CHAPTER 3



POSITIONING MODULE

3.1 INTRODUCTION

The positioning module is a vital component of any vehicle location and navigation system. To either help users obtain vehicle location or provide users with proper maneuver instructions, the vehicle location must be determined precisely. Therefore, accurate and reliable vehicle positioning is an essential prerequisite for any good vehicle location and navigation system.

Positioning involves determination of the coordinates of a vehicle on the surface of the Earth. Location involves the placement of the vehicle relative to landmarks or other terrain features such as roads. These tasks are accomplished by the positioning module in conjunction with other modules. In this chapter, we introduce the basic positioning technologies and sensors and then study sensor fusion technologies for integrating the sensor inputs used to estimate position and location.

Three positioning technologies are most commonly used: stand-alone, satellite based, and terrestrial radio based. Dead reckoning is a typical stand-alone technology. A common satellite-based technology involves equipping a vehicle with a global positioning system (GPS) receiver. Dead reckoning and GPS technologies have been used widely in vehicles. Therefore, we will concentrate on these two commonly used and easily accessed technologies in this chapter and leave the terrestrial radio technology to a later section (Section 9.2) following our discussion of radio

communications in Chapter 8. Although some of these radio technologies, like GPS, do not require users to construct central host facilities or complicated communications systems and infrastructures, this arrangement will quickly direct readers' attentions to the more popular technologies first. Readers may refer to Table 10.1 for a performance comparison of the main positioning technologies discussed in the book.

As we will learn shortly, no single sensor is adequate to provide position and location information to the accuracy often required by a location and navigation system. The common solution (frequently the only way of obtaining the required levels of reliability and accuracy) is to fuse information from a number of different sensors, each with different capabilities and independent failure modes. Therefore, a positioning module typically integrates multiple sensors, which compensate for one another to meet overall system requirements. This in turn requires that we study a variety of sensors (Figure 3.1), fusion methods, and algorithms. Because a GPS receiver functions like an absolute sensor, for convenience of study we discuss it together with the compass. Bear in mind that this is a satellite-based technology and is based on an operating principle quite different from that of other sensors (as we will learn shortly).

The positioning module is the most fundamental of the various modules needed for vehicle location and navigation. As seen from Figure 3.1, it is based on a variety of different positioning sensors. Because of the key roles played by these sensors, we will discuss commonly used low-cost sensors in depth. Readers who are not interested in the details can skip the in-depth discussions but still should be able to grasp the basic concepts in this chapter. We do not pretend to cover all possible sensors in this chapter. With the rapid advances of technology, it is difficult to cover even a particular type of sensor completely in a limited space since a variety of implementation methods can be used to manufacture a sensor based on the same theoretical principle. A similar difficulty also occurs with sensor integration and fusion technologies. However, the emphasis placed on sensors in current use and

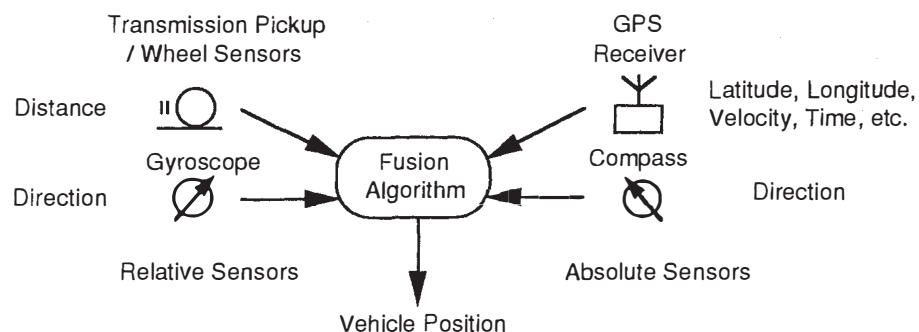


Figure 3.1 A generic positioning module.

the fundamental fusion technologies available should be sufficient for anyone interested in vehicle location and navigation to have a basic understanding of this particular area. For available linear and angular automotive sensor technologies, see the two tables summarized in Appendix C. Reviews of the latest automotive sensor development and automotive sensor technologies can be found in [1–4]. A detailed discussion of sensor technologies, sensor principles, and sensor interface circuits can be found in [5,6]. In-depth coverage of sensor integration and fusion can be found in [7].

3.2 DEAD RECKONING

A very primitive positioning technique is dead reckoning. For vehicles traveling in a two-dimensional planar space, it is possible to calculate the vehicle position at any instance provided that starting location and all previous displacements are known. Dead reckoning incrementally integrates the distance traveled and direction of travel relative to a known location. In short, it is a technique that determines the vehicle location (or coordinates) relative to a reference point.

The sensors discussed in the following sections can be used to measure the direction of the vehicle θ and the distance traveled d . After the sensor data have been sampled and integrated and sensor fusion has been performed, the vehicle position (x_n, y_n) and orientation (θ_n) at time t_n can be calculated from the equation

$$\begin{aligned} x_n &= x_0 + \sum_{i=0}^{n-1} d_i \cos \theta_i \\ y_n &= y_0 + \sum_{i=0}^{n-1} d_i \sin \theta_i \\ \theta_n &= \sum_{i=0}^{n-1} \omega_i \end{aligned}$$

where (x_0, y_0) is the initial vehicle position at time t_0 , d_i is the distance traveled or the magnitude of the displacement between time t_{n-1} and time t_n , θ_i is the direction (heading) of the displacement vector, and ω_i is the angular velocity for the same time period. A drawing outlining this method is shown in Figure 3.2.

When the sampling period is constant (and short relative to the time scale for changes in vehicle velocity), the above equation can be written as

$$\begin{aligned} x_n &= x_0 + \sum_{i=0}^{n-1} v_i T \cos(\theta_i + \omega_i T) \\ y_n &= y_0 + \sum_{i=0}^{n-1} v_i T \sin(\theta_i + \omega_i T) \end{aligned}$$

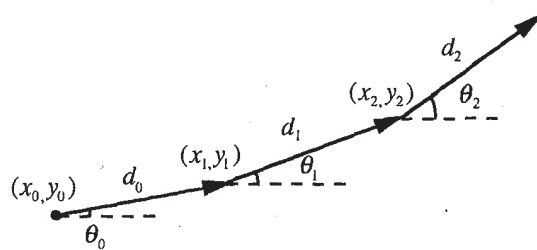


Figure 3.2 Dead-reckoning method.

where v_i is the velocity measured over the sampling period T . During each sampling period, v_i and ω_i are essentially constant. Mathematically speaking, this method basically amounts to a continuous integration of successive displacement vectors.

As seen from the preceding equations, the current calculated vehicle position during each sampling period (cycle) depends on the previous calculation cycle. It is difficult to eliminate errors associated with the previous cycle or the current measurement (due to sensor inaccuracy and the assumption that the heading remains constant over the sampling period). If not compensated or not properly compensated, these errors will generally accumulate as the vehicle continues to travel, and the calculated position of the vehicle will become less and less accurate. We will discuss algorithms (Chapter 4), short-range beacon networks (Section 8.3.7), and other technologies (Section 9.2) for eliminating these cumulative errors.

Position sensors can be designed to detect various components (distance, direction, or angular velocity) of the position of vehicular mechanical systems. They are either directly coupled to a shaft or linkage, or indirectly coupled to these or other vehicle parts in the case of a noncontact or proximity sensor. Many factors affect the choice of a sensor for a particular situation in the vehicle design, such as frequency of vibrations, temperature range, or the presence of dirt and grease. For a designer of a location and navigation system, position sensors and their output signals are often already given. The task is to make them function as an integral part of the positioning module. Therefore, a basic understanding of common sensor technologies becomes necessary.

3.3 RELATIVE SENSORS

A relative sensor is a device that can measure the change in distance, position, or heading based on a predetermined or previous measurement. Without knowing an initial position (or previous reference) or heading, this sensor cannot be used to determine absolute position or heading with respect to the Earth.

3.3.1 Transmission Pickups

Transmission pickup sensors are used to measure the angular position of the transmission shaft. Different technologies, such as variable reluctance, the Hall effect, magnetoresistance, and optically based technologies, are used to convert the mechanical motion into electronic signals. Although some transmission pickup sensors may have analog output, it is not difficult to design an interface to convert the signal to pulses indicating fractional revolution. Knowing the number of pulse counts per revolution and a proper conversion scale factor, one can convert the output of the sensor into distance traveled. To avoid mechanical wear and corresponding changes in accuracy of the measurement, it is desirable to measure the angular position of the shaft with a noncontact sensor. Because there is always an air gap between the sensor and the component being monitored, a noncontact sensor is not subject to friction wear. Magnetic fields and optical methods are the two most common methods used for noncontact coupling to a rotating shaft. In this section, we discuss only the magnetic field method.

There are two common places in the vehicle where the signal from the transmission pickup sensor can be obtained. One is from a connector behind the speedometer. The other is from a connector on the transmission housing. In general, a two-wire connector indicates a variable-reluctance position sensor and a three-wire connection indicates a Hall-effect position sensor.

These variable-reluctance and Hall-effect sensors are not limited to transmission pickups. They can be used to detect the position and speed of rotating toothed or notched wheels in crankshaft-, camshaft-, and wheel-monitoring applications. Some of these applications are discussed in the next section.

Variable-Reluctance Position Sensor

The principle of the variable-reluctance sensor is that the total electromagnetic force (emf) induced in a closed circuit is equal to the time rate of decrease in the total *magnetic flux* linking the circuit. The total magnetic flux (number of lines of magnetic force) is inversely proportional to total reluctance. Any change in the reluctance of the magnetic circuit will cause a change in the magnetic flux. If the changing flux lines pass through a coil, the change in flux will induce an emf on the coil. The output emf is measured as a voltage proportional to the rate of change of the reluctance for the path as shown in the following equation:

$$e = - \frac{d\Phi}{dt}$$

where $d\Phi/dt$ is the rate of change in the magnetic flux in webers per second, and e is the emf in volts. As a result, the output voltage goes to zero as the rate of change

approaches zero. Therefore, variable-reluctance sensors cannot be used at speeds near zero. Additional detail on magnetic field measurements and magnetic flux is provided in Section 3.4.1.

The variable-reluctance position sensor consists of a permanent magnet with a coil of wire wound around it (Figure 3.3). A ferrous wheel or iron/steel disk mounted on the rotating shaft has teeth or tabs that pass between the pole pieces of this magnet. The voltage output from the sensor is typically a sine wave. When a tooth on the ferrous wheel begins to pass between the magnet pole pieces, the coil voltage starts increasing from zero, reaches a maximum, and falls to zero again when the tooth is exactly between the pole pieces. At this point the rate of change in the magnetic flux is zero (even though the magnetic flux is at its maximum value). Therefore, the induced voltage across the sensing coil goes to zero. As the tooth continues its movement, the voltage output from the coil once again increases from zero, reaches a maximum, and falls to zero again as the tooth passes out of the gap between the pole pieces, and thus finishes a complete sine cycle.

Despite their simple, rugged construction and low cost, variable-reluctance sensors do have drawbacks [2,8]. First, it is difficult to output a constant voltage envelope. As the speed of the ferrous wheel changes, the voltage level and frequency change as well. Second, the signal-to-noise ratio can be seriously degraded by vibrations or resonance. Third, the output of the sensor is sensitive to the gap between the sensor and the disk. In other words, the output is inversely proportional to the gap size. Proper alignment and a rigid sensor mounting are required. Fourth, there is no sensor output when the disk rotation speed is below a certain threshold. This means that no output is available at very low speeds. Fifth, the range of this sensor tends to flatten at high speeds. The extent of this effect varies based on the sensor design. Finally, electromagnetic interference or radio-frequency interference can

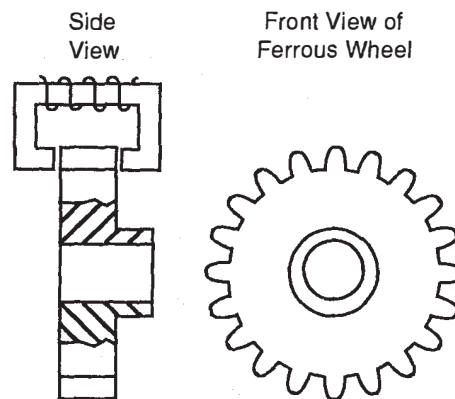


Figure 3.3 Variable-reluctance position sensor.

introduce false signals if the sensor is not properly shielded or packaged, since the coil in the reluctance sensor has a rather high impedance. Signal processing circuitry can be added to the sensor package to obtain a high signal level and to minimize the interference.

Hall-Effect Position Sensor

The Hall effect is the effect by which a voltage is generated in a conductor moving in a magnetic field. It is based on the interaction between moving electric carriers in a conductor and an external magnetic field. As a conductor is moved through a magnetic field, the current (which consists of moving electric charges) generated in the conductor will experience a force in a direction perpendicular to both the direction of motion and the magnetic field. A voltage due to the movement of these charges can be detected. Hall-effect devices can be manufactured in silicon or semiconductor materials.

A Hall-effect position sensor with a ferrous wheel or iron/steel disks is shown in Figure 3.4. The Hall-effect device is assembled into a probe with a biasing magnet and a circuit board. Figure 3.4 does not show the configuration inside the probe. The Hall-effect device actually lies at the end of the probe, close to the wheel, so that it always lies between the magnet and the wheel. When an external current is passed through this device, a voltage develops across the device perpendicular to the direction of current flow and to the direction of magnetic flux. As the ferrous wheel rotates, the reluctance of the magnetic field changes as the teeth or tabs pass the probe.

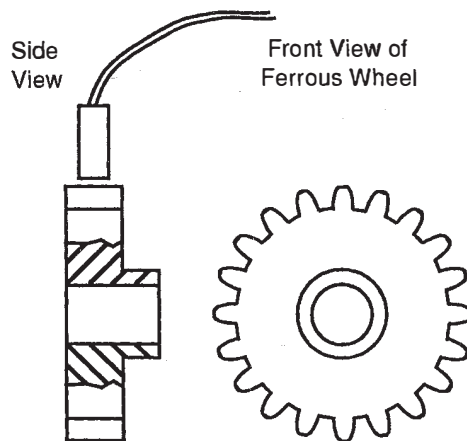


Figure 3.4 Hall-effect position sensor.

Unlike the variable-reluctance sensor, the Hall-effect sensor is a zero-speed sensor. No matter what the velocity, once the Hall-effect device lines up with the strongest magnetic field, there will be a maximum waveform output. Therefore, the voltage output (square wave) from this sensor is independent of the speed of the rotating shaft.

Sensors that utilize Hall-effect technology also have some limitations. The response degrades as the upper and lower limits of operating temperature are approached. They cannot be used in severe temperatures, lower than -40°C (-40°F) or higher than 65.6°C (150°F). The presence of other close, strong magnetic fields can cause errors. As in the case of the reluctance sensor, Hall-effect sensors require careful alignment and mounting because of sensitivity to the gap size between sensor and disk. Unlike the passive reluctance sensor, the Hall-effect sensor is an active sensor which requires an external electrical input. In addition, unlike the variable reluctance sensor, the output voltage of the Hall-effect sensor remains constant as the disk rotation frequency is varied. This sensor can easily be combined with a signal processing circuit to output a high-level voltage. In general, it is more desirable to use a Hall-effect sensor than a variable-reluctance sensor for transmission pickup sensors or other position sensors used in vehicle location and navigation systems. Further discussion of these and other automotive sensors can be found in [2,4].

3.3.2 Wheel Sensors

Wheel Sensors for Anti-Lock Braking Systems

A typical anti-lock braking system (ABS) consists of wheel speed sensors, an electronic control unit, a brake pressure modulator, wiring, relays, hydraulic tubing, and connectors to link the whole system together. Most ABS systems currently in use offer four-wheel control. The basic components used for the wheel speed measuring systems include a sensing element, a ferrous wheel (exciter wheel, target wheel, tone wheel, or iron/steel disk), and the mechanical hardware required to mount the above elements with the appropriate spacings to ensure correct operation.

The variable-reluctance sensor is still the most popular for ABSs because of its low cost and relative reliability. As discussed earlier, this sensor is not reliable at low speeds, especially below 0.45 to 1.34 m/s (1 to 3 miles/hr); this makes accurate vehicle positioning very difficult when using these sensors for input to a positioning module. Hall-effect position sensors can also be used as ABS speed sensors. We have learned that both sensors are electromagnetic pulse pickups using toothed wheels (exciter rings) mounted directly on the rotating components of the drive train, shaft, or wheel hubs. They provide a digital signal whose frequency is proportional to the rotational velocity of the wheel. Furthermore, any technology with a sensing element that converts mechanical motion into an electrical signal can also be used as a wheel

sensor. Table 3.1 compares several different wheel sensor technologies (most of the items in this table are from [9]). From this table we notice that only the variable-reluctance sensor is fundamentally incapable of measuring at zero speed (another exception is that a Hall-effect sensor cannot function as a zero-speed sensor if its feedback circuit has a capacitor).

Differential Odometer

Differential odometry is a technique to provide both distance traveled and heading change information by integrating the outputs from two odometers, one each for a pair of front or rear wheels. An odometer is a relative sensor that measures distance traveled with respect to an initial position. A wheel odometer typically measures the number of rotation counts (pulses) generated by a rotating wheel. Averaging the left and right wheel rotation counts and multiplying by a proper scale factor enables the distance traveled by the vehicle to be determined. The difference between the counts for the left and right wheels multiplied by the same scale factor and divided by the axle length can be used to obtain the change in the heading of the vehicle. Both the distance traveled and the heading change can be used in calculating the vehicle position.

One example of a differential odometer is shown in Figure 3.5. A ferrous wheel or iron/steel disk (toothed gear) is attached to each of the nondriven wheels of the vehicle (because acceleration and deceleration have less effect on the output of sensors on nondriven wheels). A Hall-effect sensor is located close to the toothed gear. This sensor outputs a pulse for each tooth that passes across its sensing tip. Two counters, one for the left wheel and one for the right wheel, count these pulses. This left or right tooth count can be translated to the distance each wheel has traveled by multiplying the count by the scale factor (which is a predetermined value for the distance per count).

Table 3.1
Wheel Speed Sensor Comparison

<i>Sensor</i>	<i>Frequency Range</i>	<i>Magnet</i>	<i>Ferrous Wheel</i>	<i>Signal/Noise</i>
Eddy current	0–500 kHz	No	No	High
Hall effect	0–1 MHz	Yes	Yes	Moderate
Magnetic transistor	0–500 kHz	Yes	Yes	Moderate/high
Magnetoresistive	0–5 MHz	Yes	Yes	Moderate
Optical	0–10 MHz	No	No	Very high
Reed switch	0–600 Hz	Yes	Yes	High
Variable reluctance	1–100 kHz	Yes	Yes	Low
Wiegand effect	0–20 kHz	Yes	Yes	High

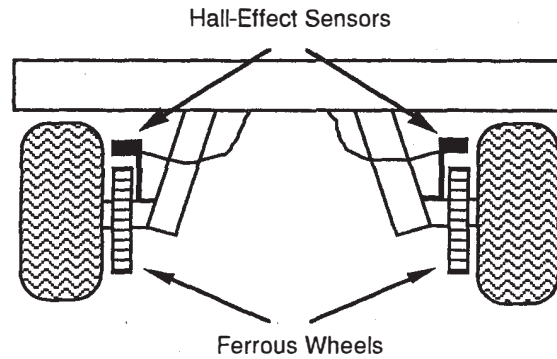


Figure 3.5 Differential odometer.

After resetting the counters and driving the vehicle along a known straight distance D while recording the counter numbers, the left and right scale factors can be calculated from

$$K_L = \frac{D}{C_L}$$

$$K_R = \frac{D}{C_R}$$

where C_L and C_R are the number of counts for the left wheel and right wheel, respectively. Knowing the scale factor for each wheel and the previous measurement at $t - 1$, we can easily derive the distance traveled up to time t as follows:

$$d(t) = d(t - 1) + \frac{K_L C_L + K_R C_R}{2}$$

When the vehicle heading is changed, the outer wheel travels farther than the inner wheel by $|K_L C_L - K_R C_R|$. Knowing the distance L between the left and right wheels (axle length), the vehicle heading can be obtained from this equation:

$$\theta(t) = \theta(t - 1) + \frac{|K_L C_L - K_R C_R|}{L}$$

Note that as long as the ratio $K_L/K_R \approx 1$, we know that the vehicle is traveling on a straight-line segment road (provided K_L and K_R are accurate). On the other hand, if we know that the vehicle is traveling on a straight-line segment based on a given digital map, we can use this information to calibrate these two scale factors

dynamically. Moreover, the outputs from other sensors (such as GPS, see Section 3.4.2) can be used to calibrate the scale factor. For instance, when the vehicle travels above a certain speed and the tracked satellite geometry used for the position fix is good, the velocity and heading information received from the GPS sensor is very good. The travel distance extracted from the velocity can then be used to calibrate the scale factor. Now we can see the benefit of using multiple positioning sensors in a location and navigation system.

Depending on the technology used, an odometer or differential odometer often has an operational error similar to that experienced by variable-reluctance and Hall-effect sensors in general. In addition, the differential odometer is subject to systematic and nonsystematic errors such as unequal wheel diameter, misalignment of wheels, the finite sampling rate and count resolution, running over objects on road, slips or skids involving one or more wheels, etc. A slip or skid occurs if the vehicle accelerates or decelerates too rapidly or travels on a snowy/icy/wet road. These errors may lead to incorrect calculated distance and heading data. Furthermore, in sharp turns, the contact point between each wheel and the road can change, so that the actual distance L between the left and right wheels will be different from the one used in the preceding equation to derive the heading. The tire pressure in each wheel may change over time, as may the wheel diameter due to the tire conditions or change in temperature. Therefore, the scale factors used to calculate position may change after initial calibration. These factors contribute to the error sources of the differential odometer, which lead to accumulative errors discussed in Section 3.2.

3.3.3 Gyroscopes

Rate-sensing gyroscopes measure angular rate, and rate-integrating gyroscopes measure attitude. At the present time, most location and navigation systems use gyroscopes to measure the angular rate.

Gyroscopes include mechanical, optical, pneumatic, and vibration devices. A comparison of these gyroscope types is presented in Table 3.2. (The gas-rate gyroscope is classified as a pneumatic gyroscope for lack of a better word.) As the explosive development of design and manufacturing technologies continues, the

Table 3.2
Gyroscope Comparison

Type	Performance	Cost	Size
Mechanical	Very good	Expensive	Large
Optical	Very good	Moderate	Moderate
Pneumatic	Good	Moderate	Moderate
Vibration	Good	Cheap	Small

characteristics of each type of gyroscope in Table 3.2 will improve over the years, and other new technologies may mature to the point where they can be used to further expand this table.

The vibration gyroscope is currently the device of choice for many vehicle location and navigation systems. Discussions of other techniques can be found in [5,10,11]. There are four different types of vibration gyroscope based on the bar (or beam) shape: tuning fork, rectangle, triangle, and cylinder as shown in Figure 3.6. A small electronic circuit is commonly used to control the vibration, convert measurements to an angular velocity output, and provide an interface to the user.

A vibration gyroscope measures the Coriolis acceleration—a phenomenon discovered by G. G. de Coriolis. An object moving perpendicular to a rotating frame will cause an observer on the rotating frame to see an apparent acceleration of the object. The Coriolis acceleration is generated by the angular rotation of a vibrating bar or fork. A change in the direction around one axis of a driving element (transducer) induces a vibration in a detection element (transducer) on another axis. Vibrating piezoelectric ceramics are often used as transducers in a vibration gyroscope and are attached to the bars shown in Figure 3.6.

In general, the apparent Coriolis acceleration a is given by

$$a = 2\omega v \sin\phi$$

where ω is the angular velocity of the rotating axes relative to a fixed set of axes, v is the velocity of the bar, and ϕ is the angle between the vectors ω and v . The

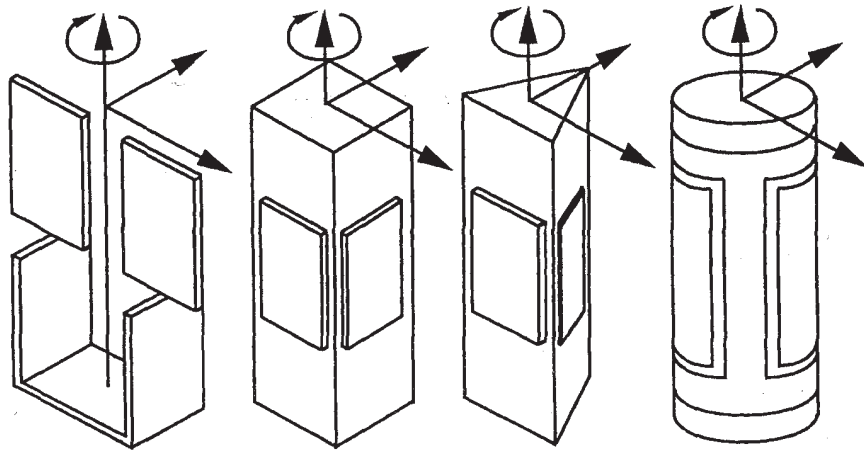


Figure 3.6 Vibration gyroscopes.

acceleration is at right angles to both ω and v . Assuming the bar has mass m and $\sin \phi = 1$, we derive the Coriolis force as

$$F = 2m\omega v$$

Notice that the Coriolis force is proportional to ω in the preceding equation. This Coriolis force is detected and converted to a voltage by the detector element and a simple circuit. Since the mass of the bar and the vibrational velocity v of the bar excited by the driving element (and the associated oscillator circuit) are both known quantities, the angular velocity ω may easily be derived and provided to the user as the output of the gyroscope. This angular velocity must be sampled and integrated to produce the change in the vehicle direction θ :

$$\theta = \int \omega dt$$

Gyroscope performance is measured by a variety of factors. Scale factor and drift (bias and offset) are major parameters used to characterize gyroscope accuracy. The scale factor describes the capability of the gyroscope to accurately sense angular velocity at different angular rates. This value is used to convert the voltage output from the gyroscope into an angular rate or direction. System imperfections can cause small variations in the scale factor. These variations appear as a factor at the output of the gyroscope. Gyroscope drift characterizes the ability of the gyroscope to reference all rate measurements to the nominal zero point. It appears as an additive term on the gyroscope output, so any actual drift away from a given or predetermined value will cause the angular errors to accumulate. The gyroscope drift model varies considerably between different sensor designs. For instance, the following error model, with adjustable parameters C_1 , C_2 , and T , has been suggested for a triangle-bar vibration gyroscope [12] (t is the time since initialization):

$$\epsilon(t) = C_1(1 - e^{-t/T}) + C_2$$

Recommended methods for reducing the gyroscope errors caused by temperature or other factors include using an analog-to-digital (A/D) converter and filters and measuring the offset voltage at zero angular velocity, just before the output of the sample-and-hold circuit [13,14].

In addition to the methods mentioned, users can also build compensatory models to dynamically calibrate a gyroscope. For instance, one method of calibrating the scale factor involves using other sensors to sense the angular rate and modify the scale factor. Another approach involves recalibrating the scale factor whenever a known turn maneuver occurs based on the digital map or other sensors. One method of calibrating the drift involves calibrating the predetermined or manufacturer-given

value whenever the vehicle is stationary. Another method is to correct the value whenever the vehicle is not turning, as determined by other sensors or the map-matching algorithm (Chapter 4).

3.4 ABSOLUTE SENSORS

Absolute heading and position sensors are very important in solving location and navigation problems. A relative sensor alone cannot provide an absolute direction or position with respect to a reference coordinate system. In contrast, an absolute sensor can provide information on the position of the vehicle with respect to the Earth. The most commonly used technologies for providing absolute position information are the magnetic compass and GPS.

3.4.1 Magnetic Compasses

A magnetic compass measures the Earth's magnetic field. When used in a positioning system, a compass measures the orientation of an object (such as a vehicle) to which the compass is attached. The orientation is measured with respect to magnetic north. Magnetic field sensors are also known as magnetometers. The intensity of a magnetic field can be measured by the magnetic flux density B . The units of measurement are the tesla (T), the gauss (G), and the gamma (γ), where $1 \text{ tesla} = 10^4 \text{ gauss} = 10^9 \text{ gamma}$.

The Earth's magnetic field has an average strength of 0.5 G. It can be represented by a dipole that fluctuates in both time and space. The best-fit dipole location is approximately 440 km off center and is inclined approximately 11 degrees to the Earth's rotational axis [5]. The difference between true north and magnetic north is known as *declination*. This difference varies with both time and geographical location. To correct the difference, declination tables are provided on maps or charts for any given locale. For a vehicle location and navigation system, this does not cause a problem as long as all directions used in the system are referred to the same (magnetic or geographic) north.

Because of portability, high vibration durability, and quick response, the electronic compasses have many advantages over conventional compasses. Among the electronic compasses, the fluxgate compass is the most popular one. The sensitivity range of this sensor is from 10^{-6} G to 100 G and the maximum frequency is about 10 kHz [15]. The term *fluxgate* is obtained from the gating action imposed by an ac-driven excitation coil that induces a time-varying *permeability* in the sensor core. Permeability is the property of a magnetizable substance that determines the degree to which it modifies the magnetic flux of the magnetic field in the region occupied by the magnetizable substance. This property in the magnetic field is analogous to conductivity in an electrical circuit. The absolute permeability μ of a given material

is a measure of how well it serves as a path for magnetic flux. A magnetic field force H applied to a high-permeability material induces a magnetic flux density

$$B = \mu H$$

where the magnetic flux density B is analogous to the current density in an electrical circuit. A magnetization curve can be plotted from the preceding equation, where the permeability μ is the slope (Figure 3.7).

High-permeability materials have two interesting properties: hysteresis and saturation. Hysteresis refers to the fact that the magnetic flux through the material lags behind the change in the magnetic field. Due to hysteresis, μ depends both on the current value of H , and the history of previous values for H , as well as the sign of dH/dt . This causes the curve to follow a different path as B increases and decreases to form a hysteresis loop (Figure 3.7). Materials saturate (in other words, further increases in magnetic flux become impossible) at high magnetic flux densities. If a material such as a core is forced into saturation by some additional magnetizing force, the flux lines of the external field will be relatively unaffected by the presence of this saturated core. The fluxgate compass uses the phenomenon of core saturation to directly measure the strength of a surrounding static magnetic field.

The permeable core is driven in and out of saturation by a signal applied to a drive coil (drive winding) on the same core. As the core is driven into and out of saturation, the magnetic flux lines from the external magnetic field being measured are also drawn into and out of the core. These varying flux lines induce positive

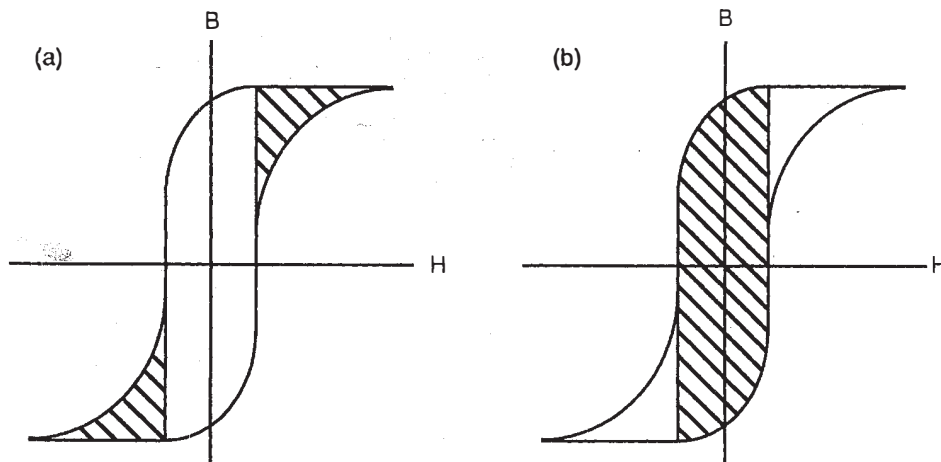


Figure 3.7 Magnetization (hysteresis) curves for high-permeability materials: (a) in saturation and (b) out of saturation.

and negative electrical current spikes in a sense coil (sense winding). Whenever the flux lines are forced out of the core, positive spikes are generated, and vice versa. The amplitude of these spikes is directly proportional to the intensity of the flux vector. The flux vector, whose direction is determined by the polarity of the pulse, runs parallel to the axis of the sense coil. The greater the difference between the saturated and unsaturated states, the more sensitive the compass will be.

Two sensing coils at right angles to each other have been developed to measure all 360 degrees of the Earth's magnetic field. These two orthogonal pickup coils can be configured in a symmetrical fashion around a common core. A toroidal drive winding is used to excite the permeable core into and out of saturation. More complete flux linkage throughout the core can be achieved using a ring geometry, which in turn requires a smaller drive excitation power and reduces interference in the sense coil(s). For these reasons, as well as ease of manufacture, the many low-cost fluxgate compasses use toroidal ring-core configuration. A sketch of such a compass is shown in Figure 3.8.

The integrated dc output voltages V_x and V_y from the orthogonal coils vary as sine and cosine functions of θ . The instantaneous value of θ can be easily derived by performing two successive A/D conversions on these voltages, and taking the arctangent of the quotient:

$$\theta = \arctan \frac{V_y}{V_x}$$

where θ is the angle of the sensor unit relative to the Earth's magnetic field.

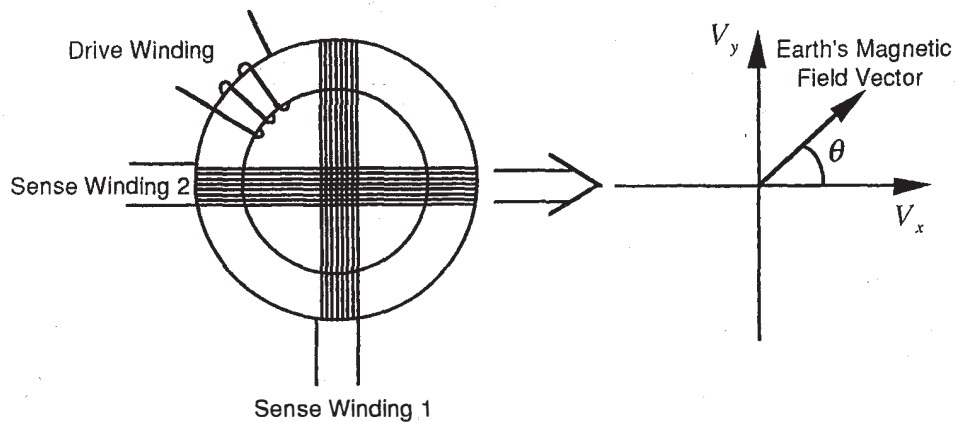


Figure 3.8 Fluxgate compass.

When the compass is installed in a vehicle it measures the orientation of the vehicle relative to the magnetic field. Any vehicle partially made of iron-based materials generates its own magnetic field, and compasses cannot distinguish between the Earth's magnetic field and other magnetic fields present. In short, compasses measure the vector sum of all magnetic fields present (Figure 3.9 shows the situation if we ignore the effects of any magnetic fields other than the two larger vectors.) Therefore, the equation just given cannot be directly used to derive the direction of the vehicle. The Earth's magnetic field vector may be extracted from the compass measurement by subtracting the magnetic field vector of the vehicle from the compass measurement.

Determining the magnetic field vector of the vehicle is simple. As the vehicle moves and changes its orientation relative to the Earth's magnetic field vector, the vehicle's own magnetic field vector moves with the vehicle. That means that regardless of vehicle orientation, the compass will "see" the sum of the vehicle vector (which remains constant) and a changing Earth vector. If the vehicle is moved through a complete 360 degrees closed loop, the compass output will trace a circle. The coordinates of the center of the circle will describe the vehicle magnetic field vector.

Several methods are available for finding the center of the circle. The simplest one is to drive the vehicle along a closed-loop path and record the maximum and minimum values measured by the x and y coils. Figure 3.10 shows the values generated from an actual trial. The two compass coils might not have identical gain, so the individual gains should be calculated using the following equations:

$$K_x = \frac{X_{\max} - X_{\min}}{2}$$

$$K_y = \frac{Y_{\max} - Y_{\min}}{2}$$

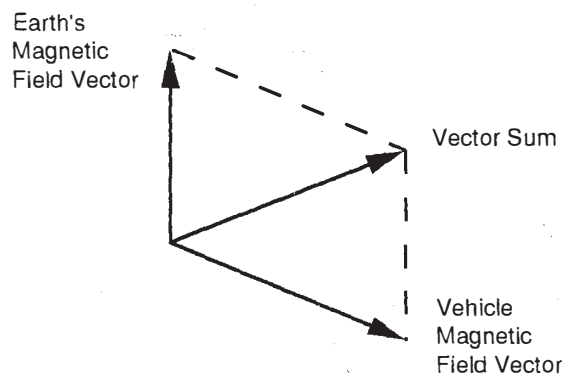


Figure 3.9 Measurement of vector sum.

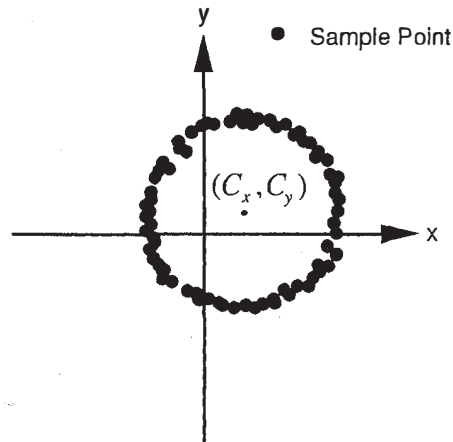


Figure 3.10 Constructing a magnetization circle to determine the magnetic field vector of the vehicle.

The coordinates of the center of the circle are midway between the maximum and minimum points and can be derived from the following equations:

$$C_x = K_x + X_{\min}$$

$$C_y = K_y + Y_{\min}$$

The direction of the vehicle may be computed using:

$$\theta = \arctan \frac{K_y(V_y - C_y)}{K_x(V_x - C_x)}$$

This method is sensitive to magnetic anomalies or noise. For example, if a truck is positioned along the calibration path such that the magnetic field of the truck affects the maximum or minimum values, the calculated coordinates determined for the center of the circle will contain these errors. Therefore, calibration should be performed in a magnetically quiet area.

In general, a fluxgate compass is very sensitive to magnetic anomalies. Figure 3.11 shows direction change data collected in a very noisy magnetic environment for a vehicle traveling in a straight line. For comparison, we also show the direction change data collected simultaneously using a differential odometer. The driver has tried to follow a straight line. However, it is very difficult to control the steering wheel to force the wheels to follow a perfectly straight line. This fact is reflected in the data collected by the odometer. For the compass data, these direction

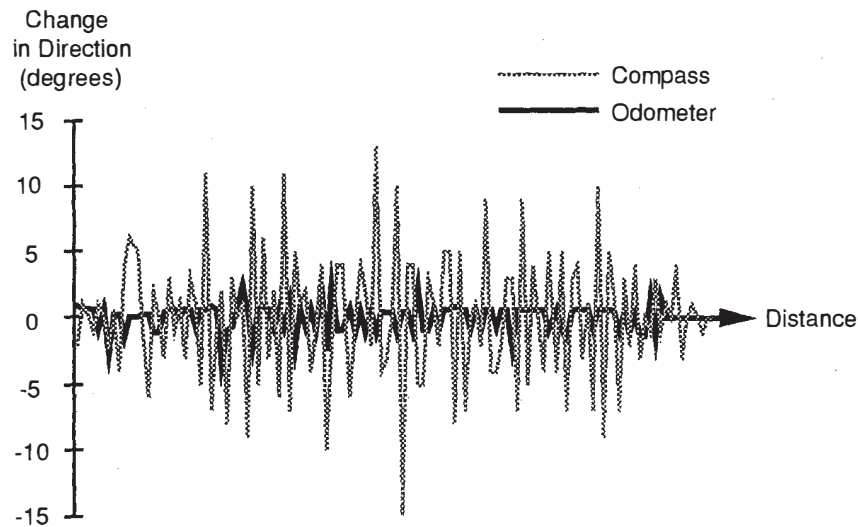


Figure 3.11 Effect of noisy magnetic environment on direction measurements.

errors result in positioning errors. Therefore, minimization of measurement errors is crucial for any location and navigation system using a fluxgate compass.

Fluxgate compass measurements generally contain two types of errors: short-term magnetic anomalies and long-term magnetic anomalies. Deviations caused by nearby power lines, big trucks, steel structures (such as freeway underpasses and tunnels), reinforced concrete buildings, and bridges are typical examples of short-term magnetic anomalies. Long-term magnetic anomalies result from inaccuracies in calibration, electrical/magnetic noise, or magnetization of the vehicle body.

Filters such as the low-pass filter or complementary filter discussed in Section 3.5 and other valid methods can be used to detect and eliminate short-term magnetic anomalies. These methods include checking the heading change reported by the compass against the physical operating envelope of the vehicle and comparing the measured vector magnitude against the expected range of the Earth's magnetic field. When the external field is strong, it produces a measurement vector that falls far from the curve defined by the circle (Figure 3.10). If we use only samples that fall close to the magnetization circle curve to determine the heading, this source of errors will be eliminated. Relative sensors can also be used to detect blunders (anomalies). When a digital map is available, road segment direction is also a very good reference for detection and correction of compass blunders. Relative sensor data and segment direction are not likely to change dramatically. These blunders are not difficult to detect using if-then rules with proper direction change thresholds relative to the abnormal compass changes associated with a blunder [16].

Errors in calibration are deterministic, and can be eliminated by introducing correction offsets. It is important to realize that calibration errors can change over time. For example, when the vehicle is run through a car wash, the automatic car wash brushes can cause the magnetic field of the vehicle to change for few days so that the center of the circle is shifted (Figure 3.12), which will in turn severely affect the performance of the positioning module. In addition to large moving or stationary steel objects, electronic devices in the vehicle may also cause errors. For instance, when a rear window defroster is in operation, the large current generated may affect a compass mounted nearby. Therefore, care must be taken when deciding where the compass should be installed in the vehicle.

Errors due to certain long-term magnetic anomalies, such as the magnetization of the vehicle body, are difficult to detect. A heading reference that does not use magnetic effects (e.g., GPS) can be used to detect and remove such errors. Electrical or magnetic noise can be averaged over distance and time, and the white noise variance can be estimated *a priori* or estimated in real time when an external heading change reference is available (e.g., differential odometer). As discussed before, the actual magnetic field direction measured by a compass is determined by the combined magnetic field from the Earth and the vehicle itself. If the vehicle is not magnetized, the normalized coil voltages form a circle with the center at the origin. If it is magnetized, the center of the magnetization circle will drop away from the origin (Figure 3.10). A shift curve to 10% of the circle radius will cause a maximum heading error of 6 degrees. The heading error due to the shift in the center of the circle is a function of the actual center and the true heading.

Two methods can be used to correct for the error due to a shift in the center of the magnetization circle. One method is to continuously search for new maximum

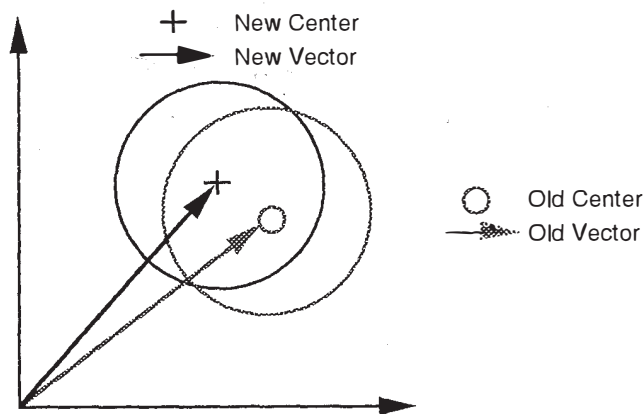


Figure 3.12 Shift of the vehicle magnetic field center.

and minimum points V_x and V_y to reconstruct the circle center dynamically. Another method is to introduce a correction table. The correction table contains the correction information for any heading. For instance, in Figure 3.13 the compass has an error of -4 degrees at 100 degrees, thus the table will contain a $+4$ -degree correction in the cell for a 100 -degree heading. Adding the correction to the compass measurement will eliminate the error. Independent sources of heading information (i.e., GPS or a digital map) can be used for determining the correction and generating the table. These heading sources are often more accurate than the compass but are not available on a continuous basis.

When the vehicle is traveling over a hilly road, the compass plane will not be parallel to the plane of the Earth's surface. The compass measures only the projection of the vector components. This type of error can be detected and removed via a check on the magnitude of the magnetic field.

3.4.2 Global Positioning System

The GPS is a satellite-based radio navigation system. It provides a practical and affordable means of determining position, velocity, and time around the globe. GPS was designed and paid for by the U.S. Department of Defense (DOD). Civilian access is guaranteed through an agreement between the DOD and the Department of Transportation (DOT).

GPS consists of three main parts: the space segment (satellites), the user segment (receivers), and the control segment (management and control). From the location

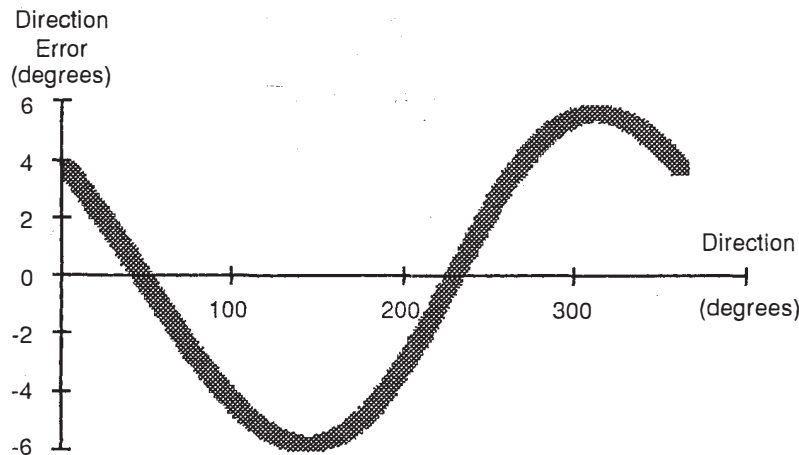


Figure 3.13 Errors due to the shift of the vehicle magnetic field center.

and navigation point of view, only the first two segments are of interest. For more detailed descriptions of all the segments, as well as various theoretical and practical aspects of GPS, readers may consult [17-19].

The observation of at least four satellites simultaneously will permit determination of the 3D coordinates of the receiver (Figure 3.14) and the time offset between the receiver and GPS time. The 3D coordinates can be computed in an ECEF Cartesian coordinate frame or in a geodetic coordinate system such as WGS 84. Using transformation equations, positions can be computed in other geodetic frames or mapping planes (see Section 2.3).

The GPS constellation consists of 24 satellites arranged in six orbital planes with 4 satellites per orbital plane as shown in Figure 3.15. It is designed to provide worldwide coverage 24 hours per day. The characteristics of GPS are summarized in Table 3.3 (some items in Table 3.3 are from [20]). The design life span of each GPS satellite is 7.5 years. The on-board time and frequency standards are the most likely functions to fail, so each satellite carries four atomic clocks on board.

In Table 3.3, SPS stands for Standard Positioning Service, PPS stands for Precise Positioning Service, and 2dRMS stands for two times the root mean square distance error (twice the standard deviation). The term 100m 2dRMS means that 95% of the time the horizontal positions will be within a circle of a 100m radius. Both the C/A code and P code are pseudorandom noise (PRN) ranging codes. PRN is a sequence of digital 1's and 0's that appear to be randomly distributed like noise, but which can be exactly reproduced. The navigation message broadcast includes satellite orbital position data and satellite health status.

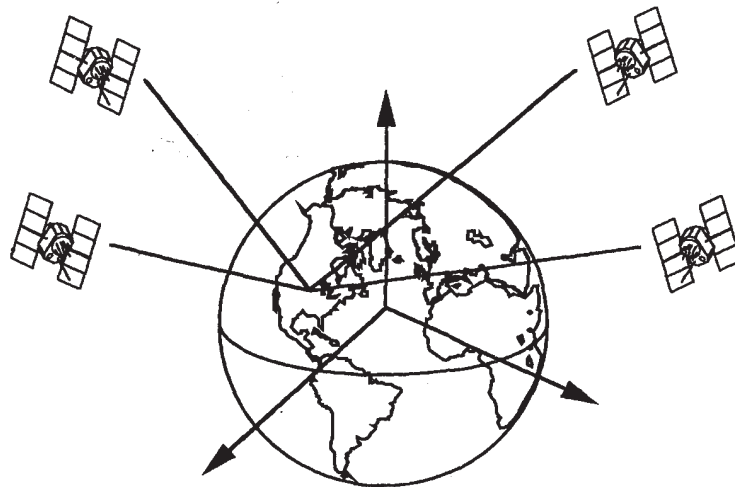


Figure 3.14 ECEF Cartesian coordinate system for GPS.

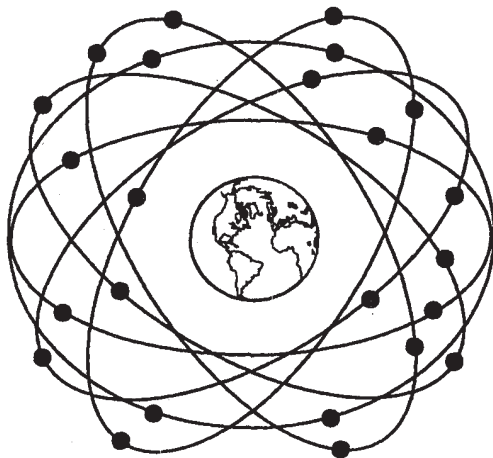


Figure 3.15 Satellite constellation (space segment).

Table 3.3
GPS Characteristics

<i>Item</i>	<i>Characteristic</i>
Satellites	24 satellites broadcast signals autonomously
Orbits	Six planes, at 55-degree inclination, each orbital plane includes four satellites at 20,231-km altitude, with a 12-hr period
Carrier frequencies	L1: 1575.42 MHz; L2: 1227.60 MHz
Digital signals	C/A code (coarse acquisition code): 1.023 MHz P code (precise code): 10.23 MHz Navigation message: 50 bps
Position accuracy	SPS: 100m horizontal (2dRMS) and 140m vertical (95%) PPS: 21m horizontal (2dRMS) and 29m vertical (95%)
Velocity accuracy	SPS: 0.5–2 m/s observed PPS: 0.2 m/s
Time accuracy	SPS: 340 ns (95%) PPS: 200 ns (95%)

The SPS is based on the C/A code and is subject to selective availability (SA). SA is an accuracy degradation scheme to reduce the accuracy available to civilian users to a level within the national security requirements of the United States. It is achieved by a combination of orbital parameter degradation and satellite clock parameter dithering. This results in an intentional error in the navigation message regarding the location (ephemeris) of the satellites and the intentional error in the

timing source (carrier frequency) that creates an uncertainty in the Doppler velocity measurements. For civilian users, only SPS is available, therefore, SA degrades the performance of civilian GPS receivers. A simple way to detect whether SA is on or off is to record stationary GPS position and velocity data over a period of time. If SA is off, the 2D position error trace will fall within a circle with a 25m radius (versus 100 m), and the peak velocity error will be about 0.25 m/s (versus 1 m/s). The satellites also broadcast a parameter known as user range accuracy (URA) as part of the navigation message, which can be used as an indication of the status of SA. Recently, the U.S. government has decided to conditionally phase out SA placed on the civilian users (without charging fees) starting in the year 2000, subject to yearly review by the president. This is certainly good news for GPS civilian users.

Differential GPS (DGPS) techniques can improve GPS performance significantly. The technique involves two GPS receivers. One is a master receiver at a reference station with known (surveyed) coordinates and the other is a receiver at a remote location whose coordinates are to be determined (Figure 3.16). The calculated position solution for the master receiver can be compared to the known coordinates to generate a differential correction for each satellite. In one such commonly used technique, the correction information can then be transmitted to the remote receiver for calculating the corrected local position. This effectively reduces the position error

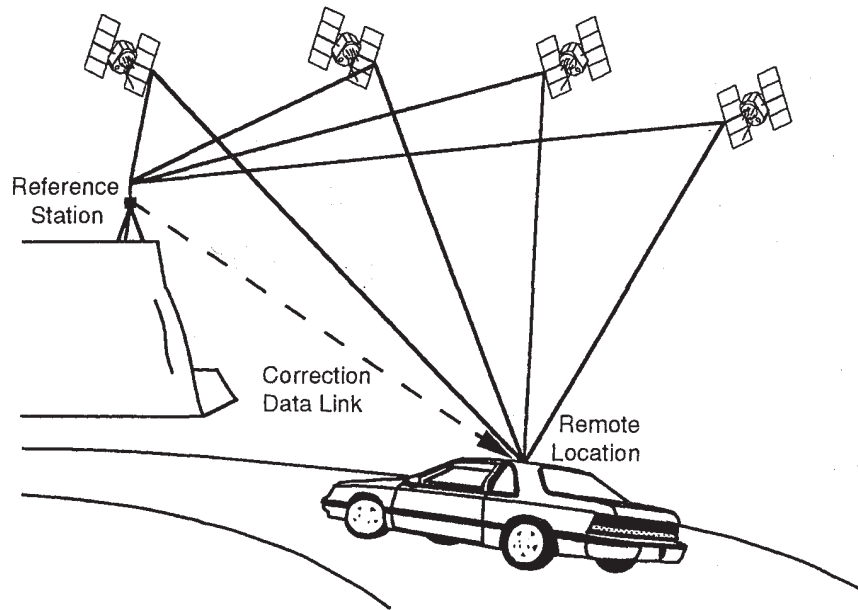


Figure 3.16 Differential GPS.

for commercial systems to under 15m even when SA is in effect. DGPS technology assumes that the range errors for the remote receiver are the same as for the master receiver. It is easy to see that this assumption may not be true all the time, especially when the two receivers are far apart. An accuracy level of 5m can commonly be achieved for receiver separations of up to 50 km when low-noise receivers are used.

Typical DGPS characteristics are listed in Table 3.4. Comparison with Table 3.3 indicates that DGPS is much better than SPS and even (surprisingly) better than PPS. A more detailed description of DGPS techniques follows the presentation of the basic positioning concepts.

We now briefly discuss the measurement principles used by the GPS receiver. As shown in Figure 3.15, a constellation of satellites is maintained in Earth orbit. Radio receivers at or near the surface of the planet are used to decode the transmitted satellite signals. These receivers then calculate position, velocity, and time from the signals. This is a one-way broadcast system, so receivers do not need to transmit any signals back to the satellites. Since it is a passive system, GPS can support an unlimited number of users, much like a television broadcast system.

Position measurement is based on the principle of time of arrival (TOA) ranging. The time interval taken for a signal transmitted from an emitter (in our case satellite) at a known location to reach a receiver is multiplied by the speed of the signal to obtain the emitter-to-receiver range (distance). Multiple signals received by a receiver from multiple emitters at known locations are used to determine its location. Because of clock offset between satellite and receiver, propagation delays, and other errors, it is impossible to measure the actual range, so a pseudorange is measured. The clock offset is the constant difference in the time reading between the satellite clock and receiver clock. To determine the position of the receiver, the receiver needs to know the pseudoranges to the satellites being tracked and the positions of those satellites. Pseudoranges are obtained by multiplying the apparent signal propagation time by the speed of light. The signal propagation time is determined by measuring the time offset required to match the received satellite code to an internally generated replica of the code. This is known as "correlation." As in all surveying situations, the number of independent observations required depends on how many unknowns need to be solved.

Table 3.4
DGPS Characteristics

<i>Item</i>	<i>Real-Time Characteristic</i>
Receiver separation	<50 km
Data link update rate	5–10 sec
Position accuracy	15m (2–5m for low-noise receivers)
Velocity accuracy	0.1 m/s
Time accuracy	100 ns

In the 2D case, at least two satellites are required to derive the receiver position. If the positions of the satellites are known and the pseudorange from the receiver to each satellite can be measured, the receiver position is constrained to be one of two points of intersection between circles having radii equal to the pseudoranges (Figure 3.17).

The unknown receiver position (x, y) can be calculated using the following equations. Note that these equations use a 2D Cartesian coordinate system for the reference coordinate frame.

$$p_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2}$$

$$p_2 = \sqrt{(x - x_2)^2 + (y - y_2)^2}$$

where (x_1, y_1) and (x_2, y_2) are the known coordinates of the satellites, and p_1 and p_2 are the measured pseudoranges. After solving these two simultaneous equations, a rough initial estimate (or a third satellite) can be used to determine which of the two solutions should be used.

For the 3D case, three satellites are required (Figure 3.18). Now instead of circles, the pseudoranges yield spheres of position. Clearly, a unique receiver position (x, y, z) of the receiver can be determined by solving the following equations:

$$p_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}$$

$$p_2 = \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2}$$

$$p_3 = \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2}$$

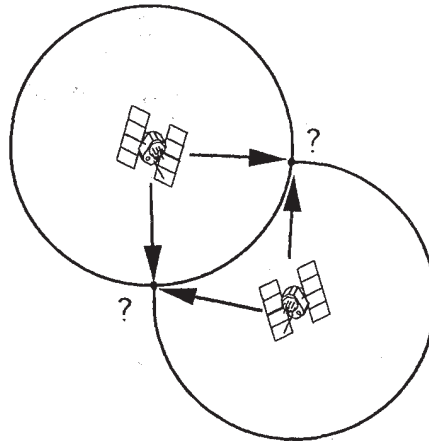


Figure 3.17 Two-pseudorange intersection.

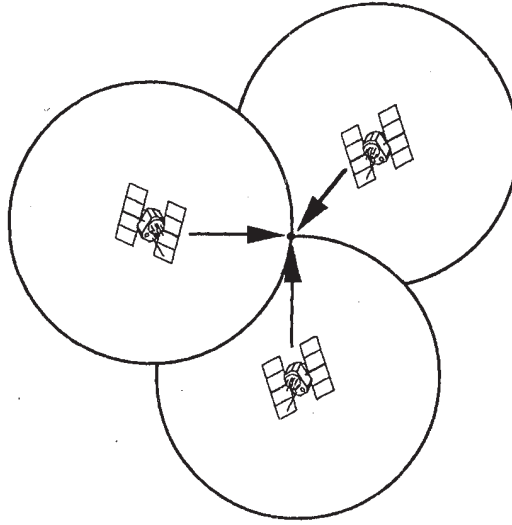


Figure 3.18 Three-pseudorange intersection.

where (x_1, y_1, z_1) , (x_2, y_2, z_2) and (x_3, y_3, z_3) are the known positions of the satellites, and p_1 , p_2 and p_3 are the measured pseudoranges.

The receiver clock used to measure the signal propagation times is not synchronized to GPS time. Instead, the clock offset between receiver time and GPS time must be determined. This parameter can be calculated by adding a fourth satellite. By design, all of the satellite clocks are synchronized using very precise atomic clocks. If the receiver clock could be precisely synchronized with the satellite clocks, time measurement would be simple. However, it is economically impractical for receivers to use atomic clocks, so instead, inexpensive crystal oscillators are used. These clocks introduce a time offset (clock bias) between the receiver and the GPS clocks so we must consider their effects in the computation. The receiver clock bias is the time offset of the receiver, and it is the same for each satellite. Thus both the receiver position and clock offset can be derived from the following equations:

$$p_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} + c(dt - dT_1)$$

$$p_2 = \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} + c(dt - dT_2)$$

$$p_3 = \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} + c(dt - dT_3)$$

$$p_4 = \sqrt{(x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2} + c(dt - dT_4)$$

where (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) , and (x_4, y_4, z_4) are the known satellite positions, p_1, p_2, p_3 , and p_4 are measured pseudoranges, c is the speed of light, dT_1, dT_2, dT_3, dT_4 are the known satellite clock bias terms from GPS time, and dt is the unknown receiver clock offset from GPS time. The satellite clock bias terms are calculated by the receiver from the broadcast navigation message. In the preceding equations, several error terms have been left out for simplicity. For instance, the range errors due to ionospheric delay and tropospheric delay can both be estimated using atmospheric models. However, receiver noise, multipath propagation error, satellite orbit errors, and SA effects remain. The square-root term represents the geometric range between the receiver and satellite, and all other terms contribute to the measurement being a pseudorange.

The preceding equations can be solved by standard techniques such as the Newton-Raphson method. If more than four satellites are used for a solution, a least squares algorithm (Appendix D) or a multiple-state Kalman filter algorithm (Appendix E) can be used. The algorithm used and how it is used have a large impact on the accuracy of position determination and the smoothness of the receiver trajectory.

Let us use four satellites as an example to explain briefly how to solve these equations and to derive measurements of the satellite geometry with respect to the receiver. After linearization, the above equations can be written in a vector-matrix form as

$$Y = AX$$

where Y is the known vector of measurements and X is the unknown vector. Linear algebra can be used to solve for the unknown vector X . A by-product of the computation is the cofactor matrix

$$P = (A^T A)^{-1}$$

where T stands for the matrix transpose operation and -1 stands for matrix inversion. For a 3D GPS solution, the cofactor matrix is a 4×4 matrix where three components are contributed by the receiver position (x, y, z) and one component by the receiver clock. The matrix can be written as

$$P = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xt} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yz} & \sigma_{yt} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 & \sigma_{zt} \\ \sigma_{tx} & \sigma_{ty} & \sigma_{tz} & \sigma_t^2 \end{bmatrix}$$

To use a GPS receiver's output properly, users must understand various potential GPS errors such as the uncertainty in satellite positions and the effects of

propagation conditions. In addition, the geometrical position of the receiver with respect to the satellites being tracked plays a considerable role in GPS receiver measurement errors. Various dilution of precision (DOP) measures are used to characterize the geometric contribution to the receiver error. The calculation of these DOPs is based on the square root of the sum of the variances of the errors in the estimated position along the three coordinate axes, and of the estimated time as shown below. We usually consider the GPS geometry good if the GDOP (geometrical DOP) is less than 3 or PDOP (position DOP) is less than 12.

$$\text{GDOP} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2 + \sigma_t^2}$$

$$\text{PDOP} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}$$

Velocity measurements are based on the Doppler frequency shift principle. The Doppler shift in the frequency of each satellite is a direct measure of the relative velocity of the receiver and the satellite along the line between them. Each satellite has a very high velocity relative to a stationary receiver, both because of the orbital motion of the satellite and because of the motion of the receiver associated with the Earth's rotation. The velocity solution can be calculated by differentiating the four-dimensional navigation solution (pseudorange equations) introduced earlier with respect to time.

To have a general idea of the composition of a GPS receiver, Figure 3.19 depicts a block diagram of a Motorola Oncore GPS receiver, where ASIC stands for application specific integrated circuit and EEPROM stands for electronically erasable programmable ROM. See [21] for more information about the receiver design and some practical receiver examples.

Readers who are interested in evaluation and selection of GPS receivers for specific applications should consult [22] and [23]. For location and navigation applications, GPS receiver performance under heavy foliage and in urban canyon areas is particularly important. Overhead foliage may cause significant attenuation of satellite signals, and high-rise buildings in urban canyons may block and reflect satellite signals. Therefore, a receiver with a position filter, fast reacquisition time, and parallel channel tracking (versus multiplexing channel) capabilities is much better than one without such features (multiplexing is the practice of rapidly sequencing any or all of the available channels among the visible satellites). For performance test reports under these circumstances, see [24,25]. Another feature that enhances GPS receiver performance is receiver autonomous integrity monitoring (RAIM). It provides detection, isolation, and removal of faulty satellite signals. Finally, research indicates that an eight-channel low-cost receiver provides an excellent balance of cost and performance needed to track satellite signals reliably.

As mentioned earlier in this section, when determining the receiver position, several errors are associated with the pseudorange measurement. Table 3.5

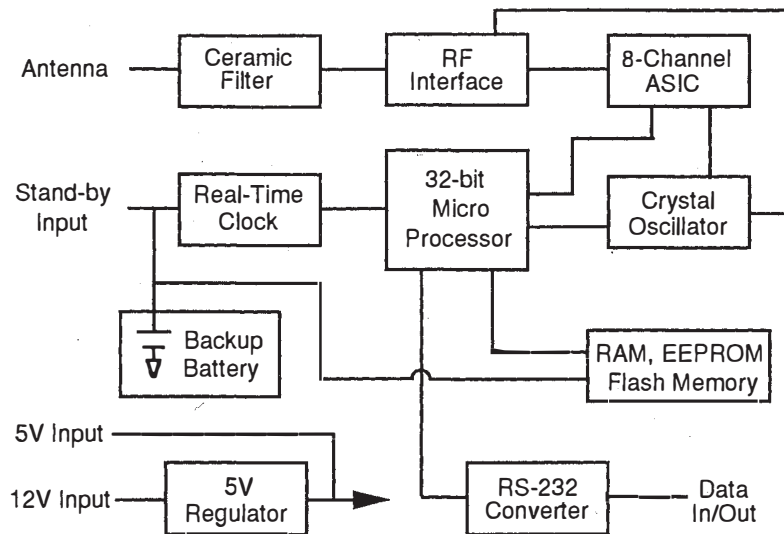


Figure 3.19 Motorola Oncore GPS receiver diagram.

Table 3.5
GPS Receiver Errors

Error	Typical GPS (m)	Worst-Case GPS (m)	Typical DGPS (m)
Satellite clock	2-3	25	<0.5
Satellite orbit	1-2	5	<0.5
SA	30-50	100	<0.5
Ephemeris prediction	3-5	15	<0.5
Ionospheric delay	10-15	100	<0.5
Tropospheric delay	3-5	30	<0.5
Multipath	10	300	10
Receiver noise	5	15	5
Total effect	100 (95%)	300 (99.99%)	15 (95%)

summarizes the typical effects of these errors on the position output by the receiver. From the table, we see that there are other sources of error in addition to SA. The first six errors in Table 3.5 are virtually zero for typical DGPS (receiver separation less than 50 km) because they are highly correlated.

Other techniques can also be used when designing the receiver algorithm to reduce the errors further. For civilian receivers, two methods can be used to calculate receiver position at first lock-on to satellites. One is based on the code phase (1.023 MHz, wavelength $\approx 293,000$ m). The other is based on the carrier phase

(1575.42 MHz, wavelength ≈ 0.19 m). Both the code and the carrier phases are transmitted by the satellite as shown in Table 3.3. The receiver generates its own code or carrier to correlate with the one received from the satellite in order to decode the navigation message. Both methods have associated range ambiguities because the receiver needs to decide the number of cycles transmitted from the satellite in order for the receiver to initially lock on. For the code phase tracking, the ambiguity is easily resolved.

Sophisticated algorithms are required to resolve the carrier phase tracking ambiguity. A combination of these two observations, as is done in carrier-aided code tracking (carrier-smoothed code) produces better results. For instance, typical Motorola Oncore DGPS can reduce both the multipath and receiver noise errors down to 1 to 2m. One study shows that by postprocessing the raw Oncore output data, DGPS accuracy can be made as good as 2 cm [26]. The postprocessing uses carrier phase differential processing and ambiguity resolution techniques.

GPS receivers can be used for time synchronization and frequency stabilization. Recall that the satellites have very expensive on-board atomic clocks with picosecond accuracy (10^{-12}) compared with the nanosecond accuracy (10^{-9}) of low-cost GPS receivers. Furthermore, the clock time-keeping ability of the satellites is better than other technologies because all satellite clocks are time synchronized by the control segment to GPS time. The time difference between any two GPS receivers is typically within 60 ns. Therefore, it is very economical to use low-cost GPS receivers to synchronize time to either GPS time or universal time coordinated (UTC) time. Examples of such applications are to synchronize time between adjacent cellular telephone cells to smooth the transition of a caller from one cell to another, and to synchronize time among three radio-frequency reference sites to locate a mobile user (Sections 9.2.2 and 10.2). Because GPS receivers can generate steady pulses (one pulse per second or 1 Hz), these pulses can be used for comparison with oscillator frequencies. Using a correction circuit, the frequency of the oscillator can be stabilized very precisely.

GPS receivers have many different applications. In geodetic surveying, a reference station receiver with few mobile receivers at selected surveying locations can collect raw data in the field, and the resulting data can be used to calculate the distances between the receiver locations in the office. This is called a *static postprocessing survey*. A reference station receiver with one mobile receiver can collect data while the mobile receiver is moved to different surveying locations in the field, and the distances are calculated in the office. This is known as a *semi-kinematic* or *rapid static survey*. A reference station receiver or a mobile receiver can broadcast its raw measurements to the other receiver, so that carrier phase differential processing can be done in real time. This method is called a *real-time kinematic (RTK) survey*.

DGPS has different implementations. For instance, if both the satellite positions and the receiver positions are known, the range correction vector can be derived by subtracting the known range vector from the measured range vector. The correction

vector can then be transmitted from the reference station to the mobile receiver. The data transmission rates for the corrections are determined by the rates of variation in the errors. For autonomous vehicle location and navigation, one-way communication of corrections is sufficient. For vehicle tracking, two-way communication may be required if the system needs both the user in the vehicle and a central dispatcher to have corrected positions. VHF (very high frequency) radio, two-way radio, FM subcarrier radio, pager signals, cellular telephone, or satellite-based communications, etc., can be used for this communications link. We discuss RF communication in more detail in Chapter 8.

One cost-effective technique for vehicle tracking is inverse DGPS. In this situation, the vehicle does not have the corrected positions; only the reference station does. This technique requires the vehicle to transmit satellite numbers and ephemeris data for the satellites it is tracking to the reference station along with the vehicle position. The software in the reference station uses these data and other data collected by the reference station using the same satellites to compute a position fix. It then derives the correction vector for that vehicle and applies it. If the system needs to track many vehicles, the reference station needs a very powerful processor or processors because of the computationally intensive nature of the task.

Wide-area DGPS (WADGPS) is another cost-effective method of bringing differential techniques to many users. Recall that DGPS needs to have a receiver separation of less than 50 km to provide very good accuracy. In other words, the main disadvantage of DGPS is the high correlation between the reference-station-to-user distance and the positioning accuracy achieved. To cover a wide area, a substantial increase in the number of reference stations is required. In contrast, WADGPS requires very few reference stations. For example, 10 sparsely located reference stations may be enough to cover the continental United States. Using the GPS receiver data collected by these 10 stations, a national map can be calculated for the entire country. Whenever any local user needs the service, this national map can be used to derive corrections for a virtual reference station in that particular area. Since this virtual reference station is nearly equivalent to a real reference station in the same area, similar position accuracy can be achieved. Research has also been very active in applying similar techniques to determine the exact satellite orbits. If real-time computation can be achieved, this *precise orbits* method should be able to improve the positional accuracy even more. Currently, precise orbits and clocks are available for postmission use only.

In addition to the Global Positioning System developed by the United States, other satellite systems have been developed that could be used for location and navigation, such as GLONASS, which was developed by the former USSR. NAVSAT has been developed by the European Space Agency (ESA), LOCSTAR is being promoted by the French space organization (CNES), and STARFIX has been developed by John E. Chance and Associates of Louisiana for the Gulf of Mexico and continental United States. More information about these systems can be found in [21,27] and

the references cited therein. We discuss other satellite communication technologies in Section 8.3.8; most of the satellite systems in operation or in planning also offer position determination as a feature. In general, the positional accuracy provided by these technologies is worse than that of GPS.

3.5 SENSOR FUSION

From the preceding discussion, it is clear that no single sensor can provide completely accurate vehicle position information. Therefore, multisensor integration is required in order to provide the on-vehicle system with complementary, sometimes redundant information for its location and navigation task. Many fusion technologies have been developed to fuse the complementary information from different sources into one representation format. The information to be combined may come from multiple sensors during a single period of time or from a single sensor over an extended period of time. Integrated multisensor systems have the potential to provide high levels of accuracy and fault tolerance.

Multisensor integration and fusion provide a system with additional benefits. These may include robust operational performance, extended spatial coverage, extended temporal coverage, an increased degree of confidence, improved detection performance, enhanced spatial resolution, improved reliability of system operation, increased dimensionality, full utilization of resources, and reduced ambiguity. However, we must keep in mind that although there are many advantages in sensor fusion, most fusion methods make certain assumptions, either explicitly or implicitly. If the assumed sensor model does not adequately describe the data from the real sensor, a perfect sensor modeling and estimation theory may not produce the expected result.

3.5.1 Simple Filters

Because of error propagation, many different error filtration methods are used. One such method is the digital filter. A low-pass filter can be used to reduce the disturbing effects of other magnetic fields on the compass. A high-pass filter can be used to eliminate errors due to odometer drift. The output of these two filters can then be integrated to obtain the output from the positioning module. From the basic deterministic methods of linear system analysis, a low-pass filter can be written in the form of a transfer function:

$$G(s) = \frac{1}{1 + \tau s}$$

where τ is the time constant of the low-pass filter. If a hardware filter is used, τ could be the product of a resistance and a capacitance. By substituting s with $1/s$, we can derive the transfer function for the high-pass filter:

$$G(s) = \frac{s}{s + \tau}$$

Both filters can be implemented in either hardware or software. One such implementation is discussed in [28].

If the sensor outputs come from two independent noisy measurements of the same signal, a complementary filter can be used to reduce the noise without distorting the signal [29,30]. The filter is shown in Figure 3.20, where $x(t)$ is the measured signal, $n_1(t)$ and $n_2(t)$ are the noises on each signal, $y(t)$ is the output of the complementary filter, and $G(s)$ is the transfer function, which represents the dynamic characteristics of the underlying sensor system. This technique is very effective when the spectral content of the noise is mainly at low and high frequencies, respectively.

The purpose of this filter is to perform noise cancellation and to obtain an output $y(t)$ as close as possible to the original signal $x(t)$. Similarly, from the basic deterministic methods of linear system analysis and the block diagram in Figure 3.20, it is not difficult to see that the Laplace transform of the filter output is

$$Y(s) = X(s) + N_1(s)[1 - G(s)] + N_2(s)G(s)$$

Clearly, the signal term $X(s)$ is not affected by $G(s)$. Therefore, we can select a $G(s)$ that attenuates both noise components.

3.5.2 Kalman Filters

Kalman filtering uses the statistical characteristics of a measurement model to recursively estimate fused data. These data are optimal in a statistical sense. If a system can be described using a linear model and both the system and sensor errors can be modeled using Gaussian white noise, the Kalman filter will provide statistically optimal estimates for the fused data. The filter can be used without storing any historical data because of its recursive nature. It can also be used to blend location determined by the positioning and the map-matching modules (Chapter 4) to yield a best estimate for the position.

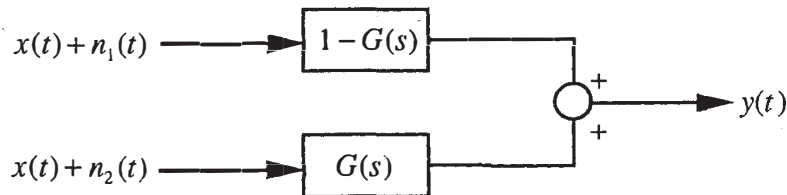


Figure 3.20 Complementary filter.

A Kalman filter can be used to fuse measurements from multiple sensors and provide both an estimate of the current state of a system and a prediction of the future state of the system. Two approaches can be taken: a centralized (standard) Kalman filter or a decentralized Kalman filter. A centralized filter takes all the sensor input and outputs a complete solution while a decentralized filter uses local filters for each sensor and a master filter to fuse all the sensors (Figure 3.21). When applied to multisensor systems and/or systems with embedded local filters, research results show that a centralized filter may result in heavy computation loads, poor fault tolerance, and an inability to correctly process prefiltered data in a cascaded filter structure. During the last several years, many decentralized Kalman filters have been introduced in an effort to improve fault tolerance (see [31] and its references).

Many good specialized textbooks have been published that discuss the underlying principles and implementation of Kalman filters [29,32,33]. The recursion equations for discrete Kalman filters are given in Appendix E. As mentioned earlier, the fusion method generally makes certain assumptions. This is certainly the case for a Kalman filter. It assumes that all measurement errors not modeled are uncorrelated and that the variations in these errors are uncorrelated (white noise). Therefore, if these assumptions are not valid, a theoretically perfect filter algorithm might lead to unacceptable results. Since the discrete Kalman filter is recursive, round-off error may lead to problems as the number of steps increases. Sometimes there may be one or more unobservable state variables. All of these situations may cause the system to diverge.

Many other Kalman filters can be used for sensor fusion. In some applications, a linear Kalman filter can be used, provided the nominal trajectory can be linearized in the state space of the system (i.e., the system model is approximately linear) and this trajectory does not depend on the results of the measurement. An extended Kalman filter can be used if the estimated trajectory can be linearized under the assumption that this trajectory can be estimated from the measurements [12]. If the estimated trajectory of the extended Kalman filter is worse than the normal one, this could lead to further error in the trajectory, which causes further errors in the estimates, and so on, leading to the eventual divergence of the filter. Therefore, caution must be used, especially in the situation where the initial uncertainty and measurement errors are both large. An adaptive Kalman filter can be used if all the parameters of the filter are not known initially or if the filter parameters change over time. When numerical stability such as round-off error is of special concern for possible divergence of the filter, a U-D covariance factorization filter can be used [34]. U is an upper triangular matrix (ones along the major diagonal, nontrivial elements in the upper triangular part, and zeros in the lower triangular part) and D is a diagonal matrix. In practice, many systems have no divergence problems, because they are completely observable and there is adequate process noise feeding into all system states.

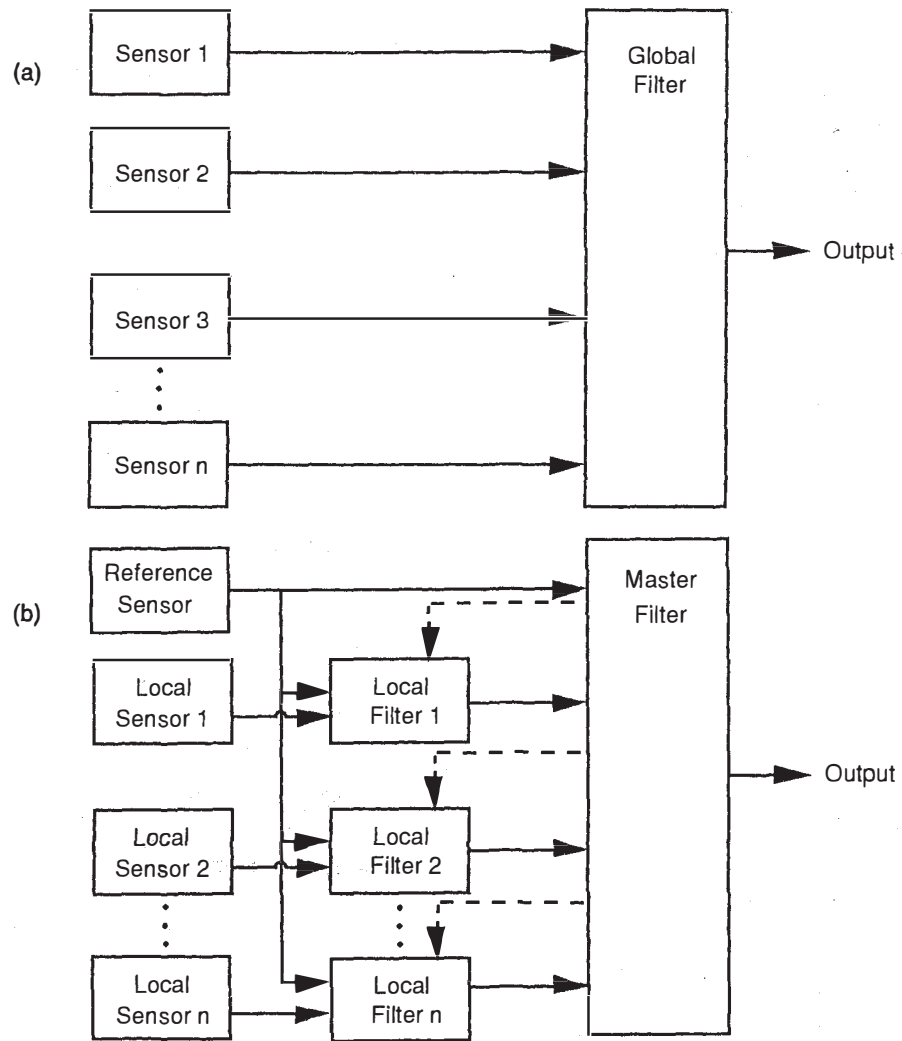


Figure 3.21 (a) Centralized and (b) decentralized Kalman filters.

3.5.3 Other Fusion Methods

Many other techniques have been devised that can be used for sensor fusion [7,35]. Although the names of some of these methods may not be familiar, we introduce

them here for interested readers to explore further. Many techniques not mentioned here are often used for pixel processing and image recognition and could be very useful for enhancing drivers' vision or even allowing automatic driving. Some techniques have already been proposed and used in practice.

Fuzzy logic can be used to represent uncertainty in the inference (or fusion) process to make a crisp decision. A distributed blackboard architecture can be used to ease the communication between various sensors in the fusion process. Weighted averaging can be employed to average redundant information. Statistical decision theory can be used to model sensor noise as probability distributions for removal of outliers. Bayesian estimation (Bayesian inference) can be applied to combine sensor information based on the rules of probability theory. Dempster-Shafer evidential reasoning can be used to extend the Bayesian estimation. Production rules can be used to symbolically represent the relation between sensory information and an attribute that can be inferred from the information as part of an expert system, etc. A neural network can be trained for fusing sensors without knowing their characteristics. Neural networks consist of many layered nodes (neurons) that are interconnected. A neural network is a dynamic system and must be trained to rearrange layers and interconnections in order to model the real-world application. One important feature of neural networks is that they simulate a dynamic system without a mathematical model of how the outputs depend on the inputs. Neural networks learn from numerical (and sometimes even linguistic) sample data.

Although various sensor integration and fusion technologies are available, we must keep in mind that any technique used in a vehicle location and navigation system must be efficient enough for real-time derivation of the correct position. Otherwise, the technique is not suitable for actual implementation in an embedded real-time system.

In general, conventional vehicle positioning techniques such as dead reckoning tend to accumulate errors; absolute positioning systems such as GPS are free from error accumulation. However, as discussed, the accuracy and availability of GPS are affected by SA and the vehicle environment. Consequently, the raw vehicle position derived from sensor measurements as described earlier will not be error free. In Chapter 4, we discuss how to use the map-matching module to enhance the vehicle positioning system.

Before turning to the next chapter, can you list the main error sources for each sensor discussed and the compensation methods learned that can eliminate or reduce these errors? Are there any other approaches that can compensate for the errors you listed?

References

- [1] K. Jost, "Sensor Technology Review," *Automotive Engineering*, Vol. 103, No. 9, Sep. 1995, pp. 39-49.
- [2] R. Jurgen (Ed.), *Automotive Electronics Handbook*, New York: McGraw-Hill, 1995.

- [3] C. O. Nwagboso (Ed.), *Automotive Sensory Systems*, London/New York: Chapman & Hall, 1993.
- [4] W. B. Ribbens, *Understanding Automotive Electronics*, 4th ed., Carmel, IN: SAMS, 1992.
- [5] J. Fraden, *AIP Handbook of Modern Sensors: Physics, Designs, and Applications*, New York: American Institute of Physics, 1993.
- [6] L. Ristic (Ed.), *Sensor Technology and Devices*, Norwood, MA: Artech House, 1994.
- [7] R. C. Luo and M. G. Kay (Eds.), *Multisensor Integration and Fusion for Intelligent Machines and Systems*, Norwood, NJ: Ablex Publishing Corp., 1995.
- [8] D. A. Foster, "Developments in Wheel Speed Sensing," *ABS Traction Control*, SAE Paper 880325, Society of Automotive Engineers, Aug. 1988, pp. 81-87.
- [9] R. F. Wells, "Non-Contacting Sensors: An Update," *Automotive Engineering*, Vol. 96, No. 11, Nov. 1988.
- [10] S. Oho, H. Kajioka, and T. Sasayama, "Optical Fiber Gyroscope for Automotive Navigation," *IEEE Trans. Vehicular Technol.*, Aug. 1995, pp. 698-705.
- [11] N. Yumoto, "Onboard Equipment," in *Advanced Technology for Road Transport: IVHS and ATT*, Ian Catling (Ed.), Norwood, MA: Artech House, 1994, pp. 338-342.
- [12] B. Barshan and H. F. Durrant-Whyte, "Inertial Navigation Systems for Mobile Robots," *IEEE Trans. Robotics Automation*, Vol. 11, No. 3, June 1995, pp. 328-342.
- [13] J. Miyazaki, R. Higashimae, T. Kuirhara, and S. Ishino, "Vibratory Gyroscope and Various Applications," *Murata Electronics North America*, 1995.
- [14] T. Oikawa, Y. Aoki, and Y. Suzuki, "Development of Vibrational Rate Sensor and Navigation System," *SAE Paper No. 870215*, Society of Automotive Engineers, 1987.
- [15] L. E. Lenz, "A Review of Magnetic Sensors," *Proc. IEEE*, Vol. 78, No. 6, June 1990, pp. 973-989.
- [16] Y. Zhao, L. G. Seymour, and E. M. Kozikaro, "Absolute Heading Sensor Blunder Detection Using Relative Heading Sensor and Road Segment," *Technical Developments*, Motorola, Vol. 29, Nov. 1996, pp. 104-107.
- [17] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, 3rd ed., Berlin: Springer-Verlag, 1994.
- [18] A. Leick, *GPS Satellite Surveying*, New York: John Wiley & Sons, 1995.
- [19] B. W. Parkinson, J. J. Spilker Jr., P. Axelrad, and P. Enge (Eds.), *Global Positioning System: Theory and Applications*, Washington, DC: American Institute of Aeronautics and Astronautics, 1996.
- [20] DoD/DoT, 1994 *Federal Radionavigation Plan*, Springfield, VA: National Technical Information Service, Report No. DOT-VNTSC-RSPA-95-1/DOD-4650.5, May 1995.
- [21] B. Forssell, *Radionavigation Systems*, New York: Prentice Hall, 1991.
- [22] J. F. McLellan and J. P. Battie, "Testing and Analysis of OEM GPS Sensor Boards for AVL Applications," *Proc. IEEE Position Location and Navigation Symposium*, Apr. 1994, pp. 512-519.
- [23] Y. Zhao, "Evaluation of GPS Receivers for Automotive Applications," *Proc. Second World Congress on Intelligent Transport Systems*, Nov. 1995, pp. 568-572.
- [24] G. Lachapelle and J. Henriksen, "GPS Under Cover: The Effect of Foliage on Vehicular Navigation," *GPS World*, Mar. 1995, pp. 26-35.
- [25] Motorola, *Navigation Performance Test Report*, Position and Navigation Systems Business, Jan. 1994.
- [26] A. Tabsh, H. Martell, and D. Cosandier, "High Precision Real-Time Positioning Using Low Cost OEM Sensors," *Proc. ION GPS-95*, Sep. 1995, pp. 1433-1441.
- [27] E. Kaplan (Ed.), *Understanding GPS: Principles and Application*, Norwood, MA: Artech House, 1996.
- [28] M. L. G. Thoone, "CARIN, A Car Information and Navigation System," *Philips Tech. Rev.*, Vol. 43, No. 11/12, Dec. 1987, pp. 317-329.
- [29] R. G. Brown and P. Y. C. Huang, *Introduction to Random Signal Analysis and Applied Kalman Filtering*, New York: Wiley, 1992.

- [30] W.-W. Kao, "Integration of GPS and Dead-Reckoning Navigation Systems," *Proc. IEEE-IEE Vehicle Navigation & Information Systems Conference (VNIS '91)*, Society of Automotive Engineers, Oct. 1991, pp. 635-643.
- [31] H. A. Carlson, "Federated Kalman Filter Simulation Results," *Navigation: J. Institute of Navigation*, Vol. 41, No. 3, Fall 1994, pp. 297-321.
- [32] A. Gelb (Ed.), *Applied Optimal Estimation*, Cambridge, MA: MIT Press, 1974.
- [33] P. S. Maybeck, *Stochastic Models, Estimation and Control*, New York: Academic Press, 1979.
- [34] E. J. Krakiwsky, C. B. Harris, and R. V. C. Wong, "A Kalman Filter for Integrating Dead Reckoning, Map Matching, and GPS Positioning," *Proc. IEEE Position, Location and Navigation Symposium (PLANS)*, IEEE, pp. 39-46, 1988.
- [35] W. Waltz and J. Llinas, *Multisensor Data Fusion*, Norwood, MA: Artech House, 1990.

CHAPTER 4



MAP-MATCHING MODULE

4.1 INTRODUCTION

The map-matching module plays an important role in vehicle location and navigation systems. It employs a digital map to make the positioning system more reliable and accurate.

To provide drivers with proper maneuvering instructions or to correctly display the vehicle on a map in an error-free fashion, the vehicle location must be precisely known. Therefore, an accurate vehicle location is considered a prerequisite for good system performance. In Chapter 3, we discussed how a vehicle location and navigation system can rely on sensors to determine its position. One such technique is called dead reckoning. Dead reckoning can track the position of a vehicle relative to another position, such as an origin. Typically, vehicle heading and distance traversed are used to determine incremental changes (Section 3.2) in the position of the vehicle relative to the origin. As discussed in Sections 3.2, 3.3, and 3.4, these sensory systems have errors associated with their operation. These errors (especially cumulative errors) detract from the accuracy of the dead-reckoning process. Even with very good sensor calibration and sensor fusion technologies, inaccuracies are often inevitable. As a result of these inaccuracies, the actual vehicle position will not agree with the dead-reckoned vehicle position. These inaccuracies are likely to increase or continue to accumulate with additional distance traversed, thereby

increasing the dead-reckoning error. The result is more uncertainty in the true position of the vehicle.

To deal with this uncertainty, map-matching systems have been developed to match the dead-reckoned position (or trajectory) with a position associated with a location (or route) on a map. When the dead-reckoning behavior indicates that vehicle is in a certain position on the map (e.g., on a particular road segment), the vehicle position may be adjusted to some absolute position on the map. This will probably eliminate the cumulative error until the next map-matching step. By doing this process in each successive cycle of the system, a more accurate position can be determined for the system. Figure 4.1 shows a comparison for two consecutive system cycles. To convey the idea, we make the accumulated dead-reckoning errors more visible. In practice, the errors should be smaller for a well-calibrated dead-reckoning system. Although dead reckoning is often used in this chapter for the convenience of the discussion to describe the position sensing and map-matching processes, we should keep in mind that other positioning techniques, such as satellite-based positioning and terrestrial-radio-based positioning, can also be used in place of dead reckoning. Because a map and a matching process are involved, these algorithms are called *map-matching algorithms*.

A good map-matching algorithm can significantly enhance the accuracy of location on a road network. The reason is straightforward. Unlike navigation in air and sea transportation, road transportation vehicles are essentially constrained to a finite network of roads, with only temporary excursions into parking lots, driveways, or other off-road conditions. This makes it possible to use computer algorithms to correlate the trajectory of the vehicle with the digital road information discussed in Chapter 2.

A map-matching algorithm can correlate the vehicle position based on various sensors to the road network by comparing the vehicle trajectory with the routes

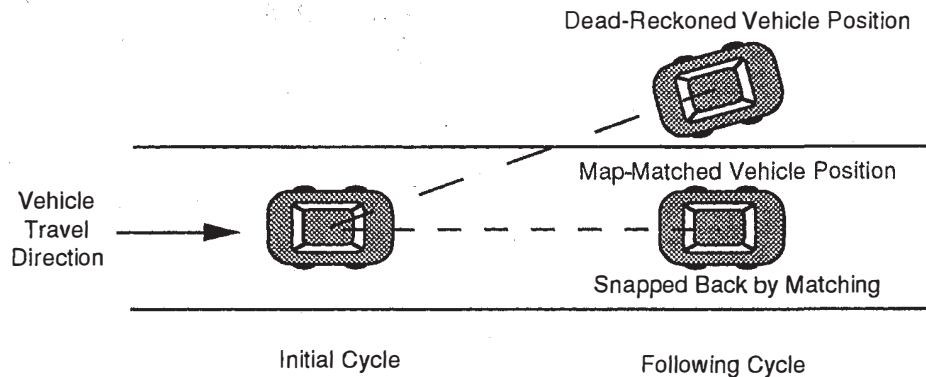


Figure 4.1 Comparison of the map-matching and dead-reckoning techniques.

present in the digital map database module. Of course, this implies that the system assumes the vehicle to be on a road. In this module, the travel path of the vehicle is constantly compared with the available routes, which consist of a series of connected roads in the road network. Through a pattern recognition and matching process, the most likely location of the vehicle with respect to the map is determined. Because the digital map database contains the coordinates of the roads, the map-matched positions can be used to reset the raw vehicle positions and limit the magnitude of the position errors.

As mentioned, map matching is the process by which the vehicle trajectory (path) is correlated with a digital road map, thereby locating the vehicle relative to this map. This is a software-based technology and makes extensive use of the digital map stored in the system. Clearly, the digital map used for this module must be relatively accurate; otherwise, the system will generate an incorrect position output, which in turn will severely degrade system performance. It is generally considered to be acceptable if the digital map is accurate to within 15m of "ground truth."

4.2 CONVENTIONAL MAP MATCHING

The basic idea of conventional map matching is to compare the vehicle's trajectory against known roads close to the previous matched position. The road whose shape most closely resembles the current trajectory and previously matched road is selected as the one on which the vehicle is apparently traveling.

The early conventional map-matching algorithm was invented in the 1970s independently by two U.S. groups and one U.K. group [1]. As time passed, this semi-deterministic method has evolved into a probabilistic method.

4.2.1 Semi-Deterministic Algorithms

The semi-deterministic method requires that an initial vehicle location (initial coordinates) and a direction of travel be provided. Conditional tests are intermittently applied to determine whether the vehicle is traveling on the known road network by comparing the dead-reckoned turns to the turns implied by the road segment vectors computed from the map coordinates. Exact correspondence is not possible due to errors in dead-reckoning sensors and in the map coordinates. The module must have a threshold to help limit potential matches. Otherwise, it would be difficult to locate any matching road segment. For this technique to work, the vehicle is generally assumed to be on a road. The accumulated errors in the dead-reckoning distance sensors are corrected whenever the system detects a turn. If the trajectory of the vehicle drifts from the road between turns, the accumulated error of the heading is also corrected. This method is based on the presumption that the vehicle is following a predetermined road. There is considerable uncertainty when the vehicle

travels off-road because there is no longer any way for the sensor errors to be corrected. For this reason, contemporary map-matching-based positioning systems tend not to use this method.

4.2.2 Probabilistic Algorithms

The probabilistic method requires that the dead-reckoning sensor errors be statistically propagated into the position determination. Elliptical or rectangular confidence regions defined based on these errors and error models are used to outline the boundary within which the true vehicle position may lie. This confidence region is superimposed on the road network. If the region contains only one intersection or road, a match is made and the coordinates are sent to be integrated or further processed as a vehicle position. If more than one intersection or road is in the region, connectivity checks are made to determine the most probable location of the vehicle. The main enhancement of this method over the semi-deterministic method is that it does not require the assumption that the vehicle is always on a road. If the vehicle appears to be off the known road network, this method repeatedly compares the sensor-detected coordinates with those road segments surrounding the off-road area and tries to identify the road segment to which the vehicle appears to have returned. Figure 4.2 contains a flowchart for a typical conventional map-matching algorithm.

Even though the flowchart for this diagram looks simple, many approaches may be used for the actual computation. Within each step, different algorithms can be used. As discussed in Section 3.5, different sensor fusion methods can be used for the first step (or logic box) in Figure 4.2.

As for the second step, many methods are available for calculating the error region associated with a particular positioning system. One way is to derive an error ellipse. From estimation theory, we know that many input and output signals can be modeled as stochastic processes. Variables associated with the true and measured values can be modeled as random variables. Variance-covariance information is propagated through appropriate algorithms to derive the variances and covariances as functions of the original random variables or as functions of parameters estimated from the original observations. These variances and covariances are often used to define error ellipses. To learn more about stochastic processes and estimation theory, readers are encouraged to consult [2-4]. The following simplified discussion should give you a basic idea of how this statistical theory is used in practice.

Suppose that the variance-covariance matrix of the system being modeled is

$$P = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

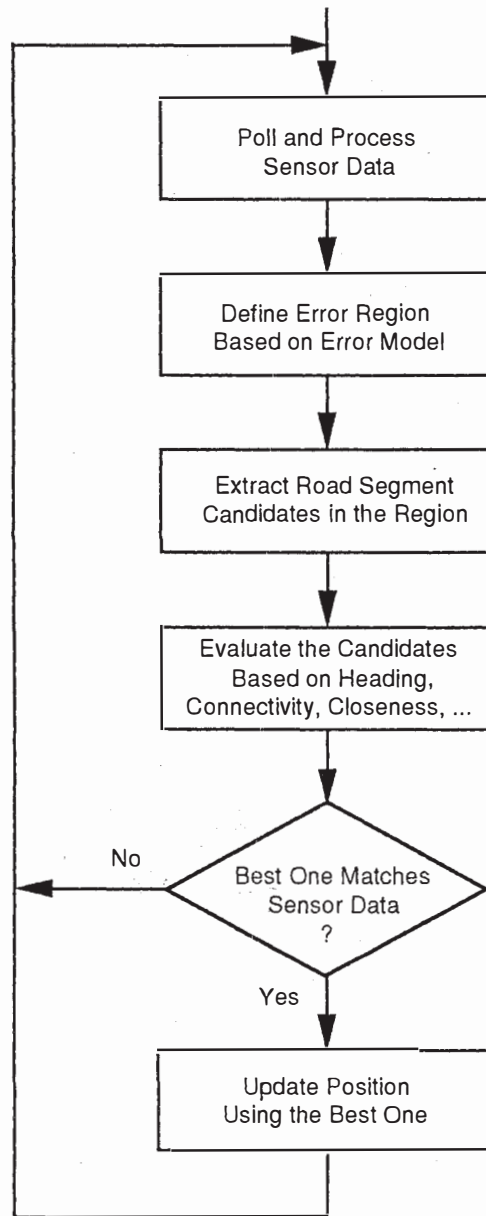


Figure 4.2 A simplified conventional map-matching algorithm.

where σ_x and σ_y are the standard deviations of the sensor measurement errors, σ_x^2 and σ_y^2 are the variances, and σ_{xy} and σ_{yx} are the covariances. The error ellipse can be derived as

$$a = \hat{\sigma}_0 \sqrt{\frac{1}{2} \left(\sigma_x^2 + \sigma_y^2 + \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4\sigma_{xy}^2} \right)}$$

$$b = \hat{\sigma}_0 \sqrt{\frac{1}{2} \left(\sigma_x^2 + \sigma_y^2 - \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4\sigma_{xy}^2} \right)}$$

$$\phi = \frac{\pi}{2} - \frac{1}{2} \arctan \left(\frac{2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2} \right)$$

where a is the length of the semi-major axis of the ellipse, b is the length of the semi-minor axis of the ellipse, ϕ is the orientation of the semi-major axis of the ellipse relative to North (Figure 4.3), and $\hat{\sigma}_0$ is the a posteriori variance of unit weight, also known as the expansion factor. Assuming that the distribution of measurement errors is a standard normal distribution, the standard ellipse ($\hat{\sigma}_0 = 1$) corresponds to a 39% confidence region. The dimensions of the ellipse can be scaled to represent various confidence levels. In the 2D case, an expansion factor $\hat{\sigma}_0$ of 2.15 can be used to obtain the 95% confidence level, and 3.03 can be used to obtain a 99% confidence level.

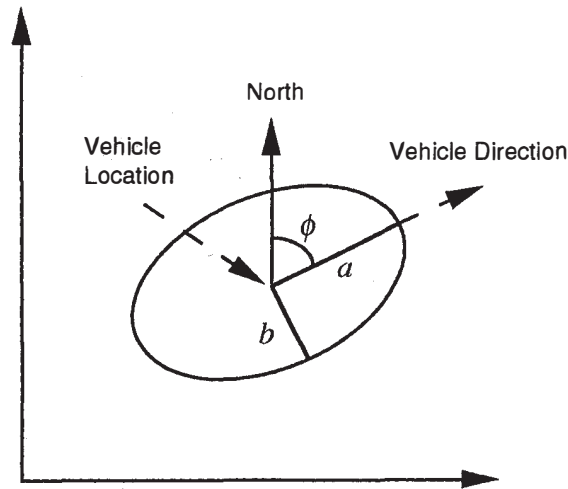


Figure 4.3 Error ellipse.

If the data received from a GPS receiver alone are used to define the map-matching error ellipse, we can obtain the variances and covariances directly from the outputs of the GPS receivers. For instance, if an Oncore GPS receiver is used, we can substitute A_{11}^{-1} for σ_x^2 , A_{22}^{-1} for σ_y^2 , and A_{12}^{-1} for σ_{xy} and σ_{yx} , where A^{-1} is the cofactor matrix P discussed in Section 3.4. These parameters are available by invoking the Position/Status/Data Extension Message command [5]. Consulting other GPS receiver manuals may allow the user to identify similar output variables, possibly under different names.

For multisensor-based map-matching systems, different statistical methods have also been developed to define the error region, such as contours of equal probability (CEP) [6], and vehicle location probability area (VLPA) [7]. We discuss the CEP algorithm in further detail later in this chapter. If a Kalman filter (Appendix E) is used for sensor integration and fusion, the task of defining the error region becomes easier. Because the variance-covariance information is available as a by-product of the filter computations, this information can readily be used to derive the error region (ellipse) as noted before.

Note that it is very common to multiply the derived error region by an expansion (confidence) factor. There may be other uncertainties in addition to the sensor errors. Map databases could have errors due to errors in the original sources, measurement errors, processing mistakes, and so on. The initial reference vehicle position might have been slightly different from the actual position. The vehicle could be anywhere within the width of a particular segment or lane. All of these factors contribute to the uncertainty, so expansion of the error region is often appropriate.

To implement the third step in Figure 4.2, it is common to first modify the error region. Since ellipses are inconvenient for locating candidate segments, a rectangle is generally used. To ensure that the error region is large enough to include all potential candidate segments, an expansion factor can once again be used to further enlarge the ellipse to a higher confidence level before the ellipse is clipped to a rectangular region. The Cyrus-Beck clipping algorithm can be used to locate the segments in the region [8]. This algorithm uses the dot products of the inner normals of the rectangular boundaries and vectors from a point on the candidate segment to a point on each boundary to determine whether the segment lies inside or outside of the clipping region.

As for the fourth step in Figure 4.2, we can sort the segments found in the previous step. The segment that most closely matches the given sensor data should be at the top of this list. Appropriateness of the segment heading is a very important criterion. Because of the many uncertainties involved with the sensor and segment heading, a threshold is used to allow the algorithm to consider the candidate whose headings are close to the sensor-detected heading. We also need to check if the segment is connected with the old segment that the vehicle was most likely traveling on in the previous computation cycle. Segments far away from the old segment should have lower priority on the list. If the vehicle was following a predetermined

route during route guidance (Chapter 6), the segments along this route should be strong candidates for matching considerations. All this checking will likely identify the best matching segment. If an off-road situation (such as a parking lot) is detected, more computations are required in order to include candidate segments within a gradually growing region of the vehicle vicinity. It uses similar matching criteria to determine when and where the vehicle returns to the road network.

Finally, this best matching segment is presented to the system along with a most probable matching point on the segment. If there are no good candidates available, if the sensor-determined position coincides with the current vehicle position, or if there is an ambiguity, nothing will be reported, provided the algorithm is designed to do so. The vehicle must then rely on the sensor data alone to determine its position.

To save on computation time, a rectangular trial-and-error region rather than an error ellipse may be defined. The drawback of this method is that the error region will never be associated with the positioning errors. A statistically defined error region can be expanded and shrunk at each cycle of the algorithm while the trial-and-error region would stay at the same size all the time. As an example, we will see below how a probabilistic algorithm, which follows the basic procedures in [6] closely, works.

This probabilistic algorithm interrupts the main system program approximately once per second to provide the current vehicle position for display. During each cycle, it first updates the old dead-reckoned or map-matched position to the current dead-reckoned position, on the basis of sensor-measured data, as discussed in Section 3.2. As in the second step of Figure 4.2, the update accuracy is estimated in terms of the CEP, which defines an error region. This region encloses an area with a certain probability of containing the actual position of the vehicle.

The basic idea is to use a 3D probability density function (PDF) to define a set of equal-probability contours (Figure 4.4). Each contour is generated by an XY horizontal plane slicing through the PDF at some level and encloses an area with a percentage of the probability density, such as 50%, 60%, and so on. The peak of the PDF is directly above the current dead-reckoned position. By projecting these contours onto the XY plane, we obtain a set of ellipses, each with an equal probability of including the actual position of the vehicle. For instance, the 90% CEP encloses an area that has a 0.9 probability of including the actual position of the vehicle. In actual implementation, one of the ellipses is selected as the error region and is approximated by a rectangle. Reference [6] contains a set of equations used to define the four corners of the rectangle based on the error models of the fluxgate magnetic compass and the differential odometer. The area of the CEP is normally proportionately larger than the area determined by the matching point (discussed shortly) from the last cycle to reflect the accumulated error and the resulting reduction in the accuracy of the current dead-reckoned position.

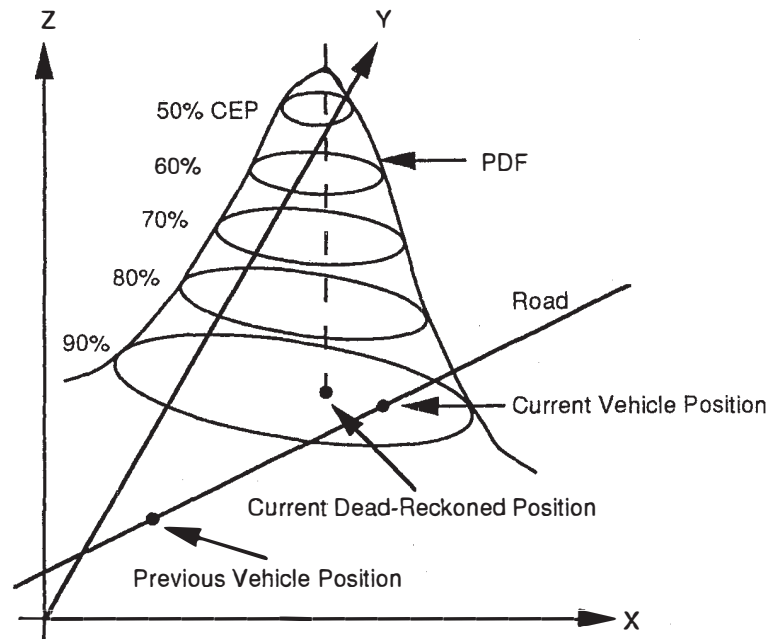


Figure 4.4 Position error regions defined by a probability density function (PDF).

Once the error region has been defined, the algorithm performs the remainder of the steps shown in Figure 4.2, except for the last update step, to find the best matching road segment. This segment is the most likely candidate in the vicinity of the vehicle to contain a point more likely than the most recent dead-reckoned position of the vehicle to be the current vehicle position. A more detailed description for each of these steps (between the second and the last steps in Figure 4.2) has been provided in Figure 4.5.

The algorithm extracts candidate road segments (segments in the error region) by examining all segments in the neighborhood of the vehicle in sequence. The segments are fetched from the digital map database. Each segment is first tested to determine if its direction is parallel to the vehicle heading within a specified threshold. If it passes the direction test, this segment is then examined to determine if it intercepts the CEP (error region), which probably contains the vehicle. After passing these initial tests, the segment is added to the segment list. Otherwise, the algorithm returns to the top and fetches another segment for the same test provided the current segment is not the last one in the neighborhood.

The algorithm tests each segment added to the list for its connectivity and closeness. The connectivity test checks if the segment connects to the segment

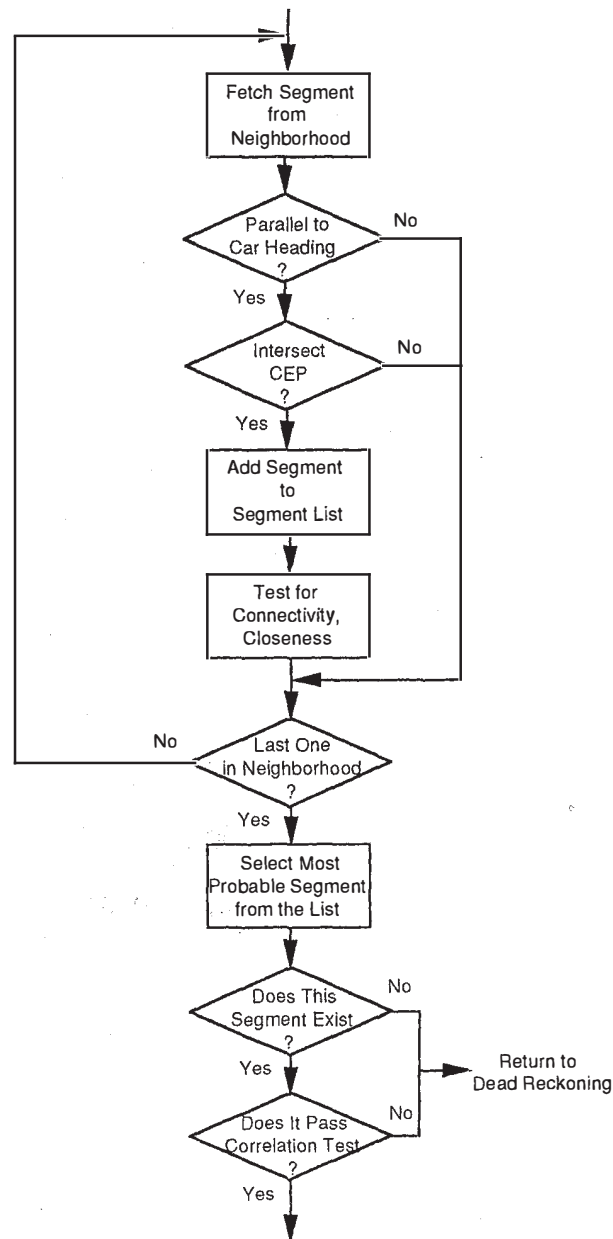


Figure 4.5 Finding a best matching segment within a confined region in a probabilistic algorithm.

previously (last cycle) determined to contain the vehicle position. In other words, the algorithm checks if each segment on the list is connected with a previously matched segment. The closeness test checks each segment on the list to see if it is very close to any previously matched segment on which the vehicle was actually traveling. In other words, the algorithm checks to see if there is an ambiguity between any pair of segments on the list. Any segment that fails these two tests is removed from the segment list. No more than two segments on the list are permitted to survive the connectivity and closeness tests.

From the surviving segment list, the algorithm selects the most probable segment. If there is only one survivor, the task is simple. This segment is the most probable segment. If there are two segments on the list, more tests must be performed. These include a test to see which segment connects to the previously matched segment properly and which one is closer to the current dead-reckoned position. The most probable segment is selected as the winner. If these two segments have almost identical characteristics or are too close together, neither segment is selected as a matching segment, and the current dead-reckoned position is not updated.

The single most probable candidate must pass the final correlation tests to be eligible as a matching segment. If the vehicle is not turning, the algorithm performs a simple trajectory match. Otherwise, a correlation function is calculated. For trajectory matching, a short history of piecewise traveled distances and headings calculated from sensor outputs are kept in memory temporarily. These vehicle trajectory data are used to compare against the vehicle route data of piecewise segment lengths and headings from the same period. If a match exists, the sole survivor is the segment to which the current dead-reckoned position can be updated and is saved as the matching segment. For correlation function computation, the algorithm calculates the function between the vehicle trajectory and certain segments (including the most probable segment and the segments connected to it). If the results pass certain thresholds, the most probable (correlation) point on the segment derived during the calculation is saved for later use in updating the current dead-reckoned position. Obviously, the segment that contains this matching point becomes the matching segment. If this most probable candidate fails the correlation tests, the current dead-reckoned position is not updated, indicated as a rightward pointing arrow in Figure 4.5.

Last, the algorithm updates the vehicle position, which is similar to the last step in Figure 4.2. If the vehicle is turning, the vehicle position is updated to the matching point determined during the correlation function calculation. Otherwise, it is updated to a most probable constant-course position along the matching segment. Once the current position has been updated to the map-matched position, the area of the CEP will be proportionally reduced to reflect the resulting increase in the accuracy of the vehicle position. Again, see [6] for the equations used to derive the new corners of this rectangular error region. Finally, the sensors are recalibrated. If the vehicle has just finished a turn, the distance sensor is recalibrated. If the vehicle

is not turning, the heading sensor is recalibrated. The turning condition of the vehicle can be easily determined either using the heading sensors or during the map-matching process. The update steps in this algorithm are shown in Figure 4.6.

We know that the probabilistic method makes no assumption that the vehicle is always on the road network. It handles off-road conditions much better than the semi-deterministic method. However, unlike traveling on a known road network, map-matching adjustments cannot prevent the accumulation of dead-reckoning errors while the vehicle is off-road. There may be considerable uncertainty in vehicle position, which could lead to incorrect conclusions when matching against the surrounding road segments. The further the vehicle travels off-road, the greater the uncertainty in the estimated position. Many possible approaches have been proposed to eliminate this problem. For instance, an absolute sensor can be used as a sanity-checking sensor to bound the absolute accuracy of the system as briefly mentioned in Section 4.4.

4.3 FUZZY-LOGIC-BASED MAP MATCHING

As discussed in preceding sections, the vehicle trajectory generated by a positioning module alone is often distorted or different from the actual route due to sensor and processing errors as well as imperfections of the database resulting from digitization.

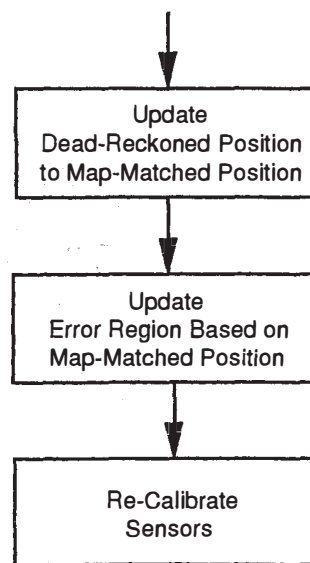


Figure 4.6 Final detailed update steps of the probabilistic algorithm.

This often causes some difficulty in the map-matching process. If a map-matching-based vehicle is traveling in a city downtown area, there may be many road patterns matching the trajectory reported by the positioning sensors at any given moment. It may be difficult to distinguish precisely on which particular street the vehicle is traveling. Rather, the system may conclude that the vehicle is "more likely" to be on certain streets, and "less likely" to be on some certain other streets. The location of the vehicle can be determined only when sufficient additional trajectory information is provided to compare against the digital map database. This ambiguity needs to be resolved if an accurate location and navigation system is desired. Therefore, techniques for dealing with qualitative terms such as likeliness are required in the map-matching process.

Fuzzy logic has been demonstrated as an effective way to deal with tasks that involve qualitative terms, linguistic vagueness, and human intervention. By making use of fuzzy sets (Appendix F), linguistic terms with vague concepts can be defined mathematically. Expert knowledge and experience is represented by a set of rules. Furthermore, inferences can be drawn through an approximate reasoning process, which simulates human reasoning capabilities. Because map matching is a qualitative decision-making process involving a degree of ambiguity, this suggests that the development of fuzzy-logic-based map-matching algorithms may be useful. Fuzzy logic can also be used in other modules or subsystems in addition to the map-matching module. Readers can refer to [9] for these potential application areas.

4.3.1 Fuzzy-Logic-Based Algorithms

In this section, we discuss how to apply fuzzy logic in a navigation system by constructing a simple rule set for a map-matching module. Background for readers who are not familiar with the concept of fuzzy logic can be found in Appendix F. As discussed in that appendix, a fuzzy logic inference process consists of three main parts: the fuzzifier, the inference engine, and the defuzzifier, as shown in Figure 4.7.

Our discussion of this fuzzy-logic-based algorithm follows [10,11] closely. This method is based on the approximate reasoning in fuzzy logic. It uses fuzzy logic to assign truth values from zero to one for each of the sensor signals and road segments and then make decisions based on the assigned truth values.

Figure 4.8 shows a flowchart for this fuzzy-logic-based map-matching algorithm. To save space, "road" has been used for "road segment" in this flow chart. The rules are numbered to correspond with the logic boxes in the flow chart. Some of them consist of more than one sub-rules.

As Figure 4.8 indicates, this algorithm uses eight rules. Some of the fuzzy membership functions associated with each rule are given next. In the following discussion of the rules, we assume that the root mean square error of the past five headings is used as the heading error to determine if the heading has become steady.

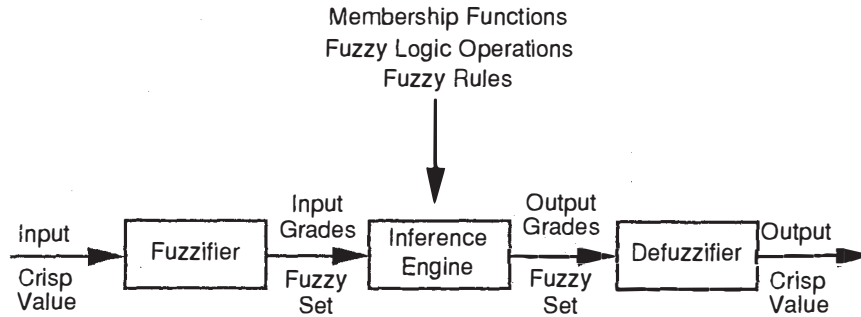


Figure 4.7 Fuzzy logic system structure.

Rule 1: IF Δ heading is small
THEN resemblance of the route is high

where " Δ heading" is defined as the difference between the direction of the road segment and the heading of the vehicle. The membership functions "small," "large," and "high" for " Δ heading" and "resemblance" are defined in Figure 4.9.

Rule 2: IF Δ heading is almost 180° AND heading error is zero
THEN possibility of U-turn is high

where, as described in Appendix F, AND is a min (minimal) operator. Membership functions of "almost 180° " for " Δ heading" and "zero" for "heading error" are defined in Figure 4.10. The membership function for "possibility of U-turn" being "high" is defined the same as "resemblance" in Figure 4.9.

Rule 3.1: IF Δ distance is large
THEN necessity to retrieve successive segments is high

Rule 3.2: IF Δ heading is large and heading error is zero
THEN necessity to retrieve successive segments is high

where " Δ distance" is defined as the difference between the segment length and the distance the vehicle has traveled on that particular segment. The membership function of "large" for " Δ distance" is defined in Figure 4.11. The membership functions of "large" for " Δ heading" and "high" for "necessity" (same as "resemblance") are defined in Figure 4.9.

Rule 4: IF the heading errors and the root mean square errors for the vehicle speed are small
THEN the motion is steady

Rule 5: IF the truth value of the previous candidate road pattern is high
AND

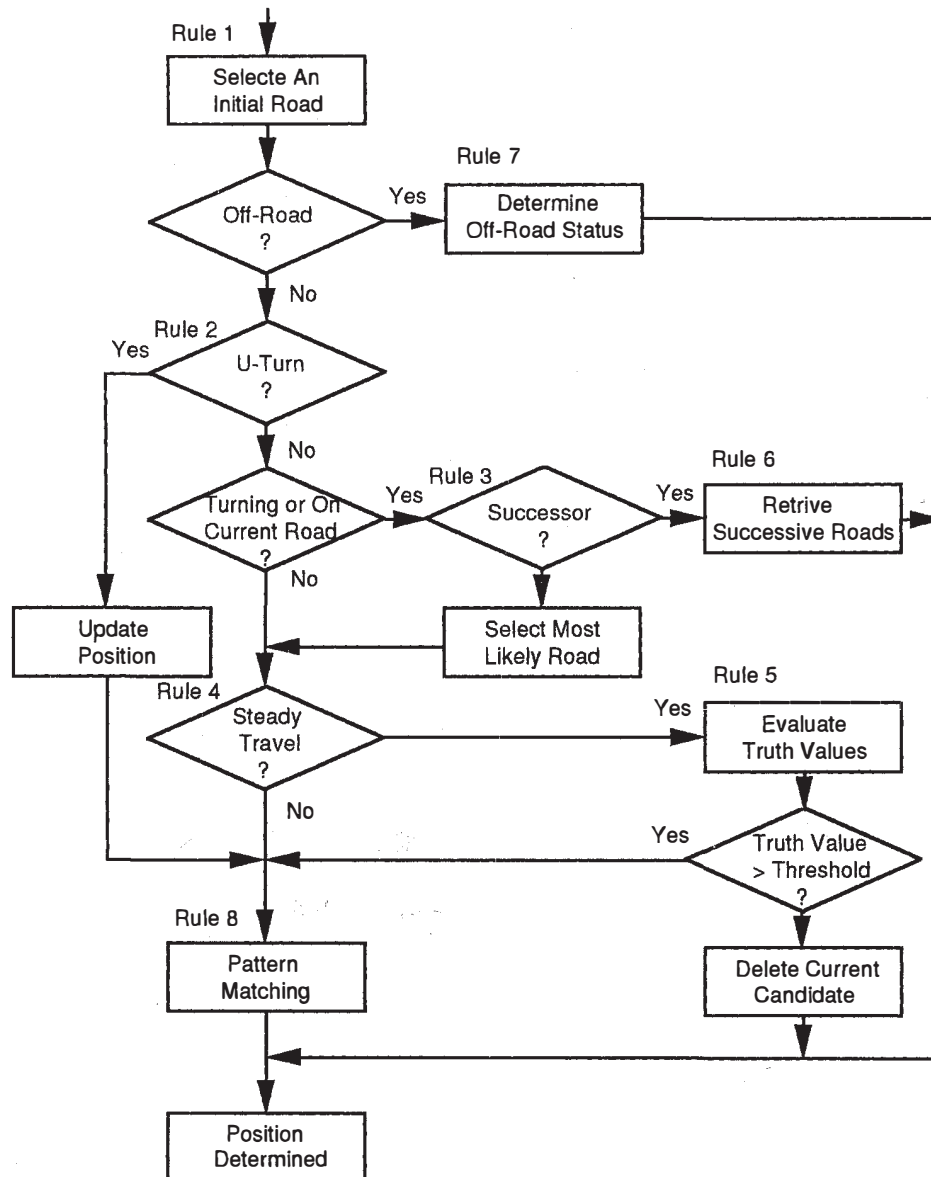


Figure 4.8 A fuzzy-logic-based map-matching algorithm.

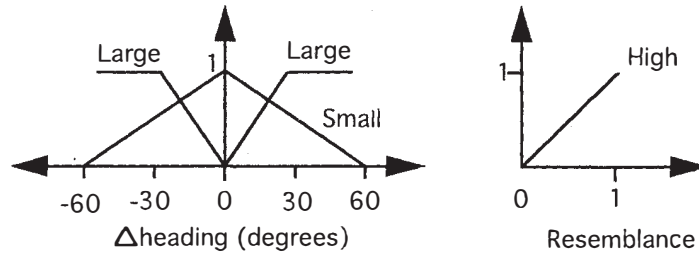


Figure 4.9 Membership functions for Rule 1.

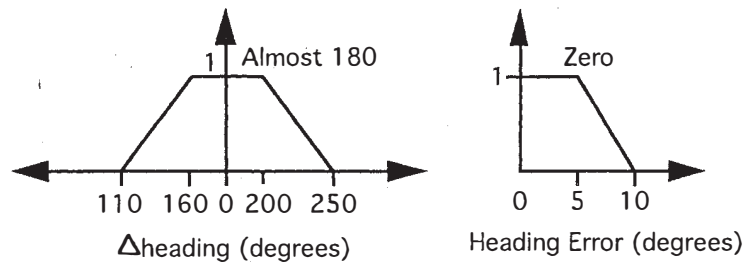


Figure 4.10 Membership functions for Rule 2.

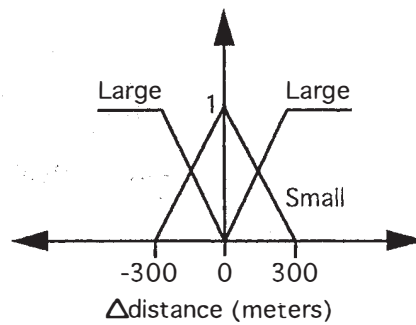


Figure 4.11 Membership function for Rules 3.

IF the truth value of the current candidate road pattern is high
THEN the truth value of the correspondence is high

Rule 6.1: IF the difference between the distance traveled along the current candidate road and the length of the candidate road is small
AND

IF the difference between the vehicle heading and direction of the successive road is small

THEN the truth value for the successive road is high

Rule 6.2: IF the truth value of the candidate road is high

AND

IF the truth value for the successive road is high

THEN the combined truth value of the moving vehicle on the successive road is high

Rule 6.3: IF no road pattern similar to the path of travel can be found within a given distance

THEN the vehicle is off-road

Rule 7: IF there is more than one road pattern within a given distance similar to the current vehicle motion

THEN the vehicle is on-road

Rule 8.1: IF Δ heading is small AND Δ distance is small

THEN resemblance of this segment is high

Rule 8.2: IF resemblance of this segment is high AND resemblance of history path is high

THEN resemblance of the whole path is high

where the AND operator in Rule 8 reflects the different degree of resemblance between successive segments and the vehicle trajectory. The membership functions of "small" for " Δ heading" and " Δ distance" are defined in Figures 4.9 and 4.11, respectively.

Through this matching process, a candidate road segment may be removed due to its resemblance being too low, for example, zero. On the other hand, a new candidate segment may be generated if the vehicle is either on or close to another segment. Because the resemblance is constantly being evaluated, the matching possibility for each candidate segment will decrease since continuous evaluation causes the truth value of the old candidate segment to decrease as the vehicle keeps moving. One way to continue the process is to reset the degree of resemblance to one whenever there is only one candidate.

The preceding set of rules is primitive. Implementation of a robust fuzzy-logic-based map-matching module requires the use of additional conditions, such as drivability of successive road segments, or the addition of special rules for GPS, if a GPS receiver is used. However, this example demonstrates very well how fuzzy logic can be successfully used in a map-matching algorithm. With these primitive rules as a foundation, one can easily develop a good map-matching module. Many fuzzy-logic development tools are available. New rule sets can be developed and tested using these tools. Users can either use the tool to generate the embedded computer language code directly for integration with the system, or design their own fuzzy inference engine to be embedded in the system [12].

The basic principles of the conventional and fuzzy-logic-based map-matching algorithms are the same. Both algorithms try to identify the road segment on which the vehicle is traveling and precisely determine the position of the vehicle on this segment to update the dead-reckoned position. The main difference is that one approach determines this matching segment by probabilistic reasoning and the other does it via a fuzzy-logic inference process. More specifically, the conventional algorithm defines the sensor error models relative to the candidate segments in probabilistic terms. The fuzzy-logic-based algorithm, on the other hand, uses membership functions to describe the sensor error models in relation to the candidate segments. The matching processes for both algorithms are essentially the same. A possibility is associated with each candidate segment. These possibilities increase or decrease according to similarity. Candidates with low possibilities are discarded and the one with the highest possibility is presented to the system. However, different methods can be used to evaluate the possibility. This leads to the interesting conclusion that other inference methods besides the two methods discussed here can also be used in map-matching algorithms.

4.4 OTHER MAP-MATCHING ALGORITHMS

Map matching is basically a pattern recognition process. Therefore, many pattern recognition methods could be used. One such candidate method is based on a neural network. As mentioned in Section 3.5, neural networks consist of many layered nodes (neurons) that are interconnected. A neural network is a dynamic system and must be trained to rearrange the layers and interconnections to model the real-world application. Besides different pattern recognition methods, centralized, decentralized, or hybrid approaches can be used to work with positioning sensors. In the centralized approach, the integrated and fused multisensor data are matched with the map. In the decentralized approach, each individual sensor or group of sensors is matched with the map separately. In the hybrid approach, the centralized and distributed methods are applied sequentially to obtain a better matching result. Although there are many variations on this technique, the underlying principle is the same: A digital map is used to filter out vehicle sensor errors and to determine the best position. As examples of these variations, we show some typical map-matching methods used with GPS receivers in Figure 4.12.

Because of the popularity of GPS receivers, many applications have been developed on the basis of this satellite-based radio navigation technology. There is no question that a map-matching process can also help improve the accuracy of the receiver output as shown in Figure 4.12(c), especially, when selective availability (SA) is on (see Section 3.4.2).

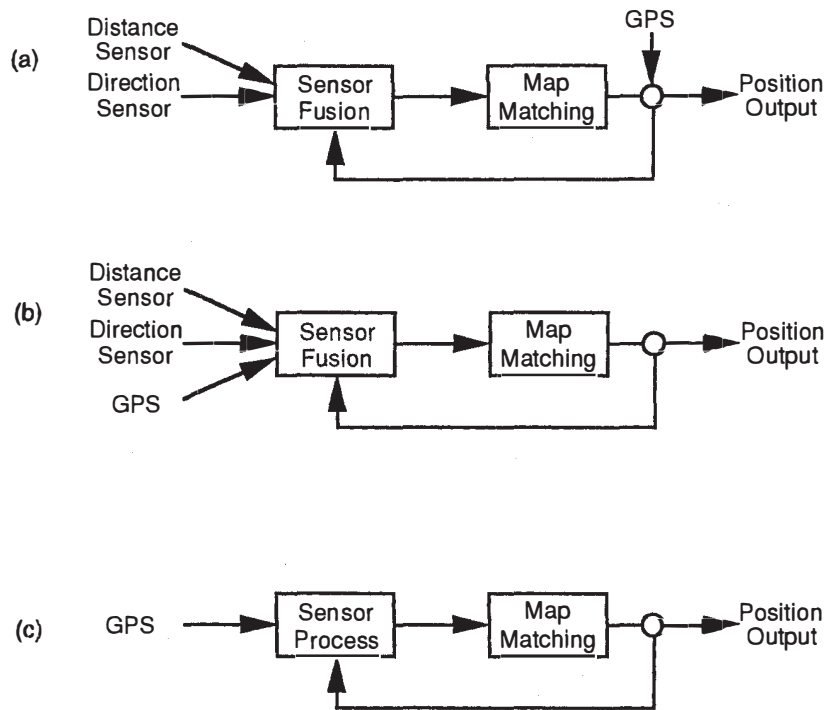


Figure 4.12 Variations of map-matching algorithms used with GPS: (a) as a sanity-checking sensor, (b) as one of integrated sensors, and (c) as the only sensor.

4.5 MAP-AIDED SENSOR CALIBRATION

Map data elements (attributes) can be used to calibrate both distance sensors and angular sensors. These may be done during a trip as briefly discussed for certain sensors in Chapter 3. For instance, sensor blunders (short-term anomalies) can be detected and corrected using the map-matching results [13]. The length of a road segment can be compared against the distance measured by the distance sensor. The direction of a road segment can be compared against the direction measured by the direction sensor. When the vehicle travels on this segment, any errors above a certain threshold will be detected and the corresponding corrections and calibration can be performed.

Assume that the map database is accurate and the fact that the vehicle is traveling on a specific road segment can be confirmed by a map-matching-based positioning module. The system can record the starting and ending points of the segment. If the segment is long enough, a comparison can be done to monitor the

performance of the distance sensor and the direction sensor. For instance, the length of the segment should be nearly equal to the distance traveled by the vehicle on the segment and the direction of the segment should be closely equal to the heading of the vehicle as it travels along that segment. Otherwise, either a sensor anomaly (blunder) has occurred or the scale factor of the sensor needs to be recalibrated. Recall that many sensors could be affected by the environment so that recalibration becomes necessary. Using a map, this can be done dynamically during the journey.

A digital map database can help a dead-reckoning-based positioning module present very good position information to the location and navigation system. When developing a map-matching-based positioning system, we must keep in mind that this system relies heavily on the map. Therefore, the accuracy of the vehicle position determined by the system should be similar to that of the map database itself. If the map is of low quality, the system may not meet expected design specifications or user requirements.

As mentioned, a dead-reckoning system has a tendency to accumulate sensor errors, so that the system may eventually fail to locate its position. A map database may be unable to provide accurate information due to unmapped roads, off-road travel, human errors during the mapping and conversion process, etc. Integration of dead-reckoning and map-matching systems may improve the performance of the positioning module. However, it still may fail due to the fact that the errors are accumulative and/or the presence of error in the maps. User intervention may be required to recover. Since the GPS satellite deployment is now complete, it is very common to find location and navigation systems that integrate GPS with dead reckoning and map matching to boost the self-recovery ability, and to provide system reliability. Besides, as a source of independent position, direction, and velocity information for the vehicle, GPS can help to calibrate the dead-reckoning sensors and detect sensor blunders dynamically. On the other hand, GPS receivers can occasionally generate suspicious position and velocity information or even no solution at all because of satellite signal blockage and multipath reflection. Dead-reckoning sensors and the map can detect GPS blunders and fill voids by erratic satellite signals. As we shall see in Section 10.2, similar map-matching techniques can also be used to assist terrestrial-radio-based positioning or other systems. By complementing each other, these systems provide much better performance than systems that do not use these matching techniques (Table 10.1).

In conclusion, let us consider an interesting question. During the segment matching process, how can you determine which is the best matching candidate when a road forks into almost parallel roads with very small differences in road direction?

References

- [1] R. L. French, "Map Matching Origins, Approaches and Applications," *Proc. Second International Symposium on Land Vehicle Navigation*, July 1989, pp. 91-116.

- [2] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed., New York: McGraw-Hill, 1991.
- [3] B. Porat, *Digital Processing of Random Signals: Theory and Methods*, Englewood Cliffs, NJ: Prentice Hall, 1994.
- [4] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, New York: Wiley, 1968.
- [5] Motorola, *Oncore User's Guide*, Northbrook, IL: Position and Navigation Systems Business, May 1996.
- [6] S. K. Honey, W. B. Zavoli, K. A. Milnes, A. C. Phillips, M. S. White, and G. E. Loughmiller, Jr., "Vehicle Navigational System and Method," *United States Patent No. 4796191*, Jan. 1989.
- [7] M. L. G. Thoone, "CARIN, A Car Information and Navigation System," *Philips Tech. Rev.*, Vol. 43, No. 11/12, Dec. 1987, pp. 317-329.
- [8] M. Cyrus and J. Beck, "Generalized Two- and Three-Dimensional Clipping," *Computers & Graphics*, Vol. 3, 1978, pp. 23-28.
- [9] Y. Zhao, A. M. Kirson, and L. G. Seymour, "Fuzzy Logic Applications in Vehicle Control and Navigation," *Abs. Third World Congress on Intelligent Transport Systems*, Paper #1225 (full paper in CD-ROM), Oct. 1996, p. 204.
- [10] L.-J. Huang, W.-W. Kao, H. Oshizawa, and M. Tomizuka, "A Fuzzy Logic Based Map-Matching Algorithm for Automotive Navigation Systems," *IEEE Roundtable Discussion on Fuzzy and Neural Systems, and Vehicle Applications*, Paper No. 16, Nov. 1991.
- [11] W.-W. Kao and L.-J. Huang, "System and Method for Locating a Traveling Vehicle," *United States Patent No. 5283575*, Feb. 1994.
- [12] G. Viot, "Structuring Fuzzy in C," *Proc. 7th Annual Embedded Systems Conference*, Sep. 1995, pp. 153-170.
- [13] Y. Zhao, L. G. Seymour, and E. M. Kozikaro, "Absolute Heading Sensor Blunder Detection Using Relative Heading Sensor and Road Segment," *Technical Developments*, Motorola, Vol. 29, Nov. 1996, pp. 104-107.

CHAPTER 5



ROUTE-PLANNING MODULE

5.1 INTRODUCTION

Route planning is a process that helps vehicle drivers plan a route prior to or during a journey. It is widely recognized as a fundamental issue in the field of vehicle navigation. Route planning can be further classified into either multivehicle (system-wide) route planning, which plans multidestination routes for all vehicles on a particular road network, or single-vehicle route planning, which plans a single route for a single vehicle according to the current location and a given destination.

In the computer literature, people often refer to finding a route from point A to point B as a *shortest path* problem. Many algorithms have been developed to solve single-origin shortest path and all-pairs shortest path problems [1]; these algorithms treat situations that are quite analogous to single-vehicle route planning and multivehicle route planning, respectively. In this chapter we discuss mainly algorithms for solving the single-vehicle route planning problem. Multivehicle route planning is more complex than single-vehicle route planning. However, the background on solving the single-vehicle route-planning problem will facilitate the study of the multivehicle route-planning case. Readers interested in multivehicle route planning will find a brief discussion in Section 5.6 as well as in relevant readings on dynamic programming, operations research, data structures, and algorithms [1–3].

In this chapter, we follow the convention used in Chapter 2. We define a *node* to be a road intersection or a dead-end point on a road, and a *segment* to be a piece

of roadway between two nodes. Note that segments are often called *edges*, *arcs*, or *links*, and nodes are often called *vertices* or *points* in the computer literature on directed graphs.

A variety of route optimization criteria (planning criteria) may be used in route planning. The quality of a route depends on many factors such as distance, travel time, travel speed, number of turns and traffic lights, and dynamic traffic information. We refer to all these factors as the *travel cost*. Some drivers may prefer the shortest distance. Others may prefer the shortest travel time, etc. Therefore, the evaluation function chosen to minimize this cost depends on the system design and user preference. These route selection criteria can be either fixed by a design or implemented via a selectable user interface. To minimize the travel distance (road segment length), distance can be stored in a digital map database, so that the route-planning algorithm can use the database when performing minimization. If travel time is being minimized, the road segment length and speed limit can be stored for use in calculating the travel time for each road segment. As noted in Section 2.6, segment length and speed limit are often defined as attributes of the road segment in the digital map database. In summary, determination of the best route involves using a digital map to select a route that minimizes variables such as time and distance.

Before introducing route-planning algorithms, we shall familiarize ourselves with some commonly used technical terms from computer science. To analyze the performance of various algorithms, we need to know the key factors that affect performance, and the common terminology used to discuss them. The *running time* (execution time) of a program depends on the input to the program, the quality of code generated by the compiler used, the nature and speed of the machine instructions, and the time complexity of the algorithm. The running time of a program is defined as a function of the input. For many programs, the running time is a function of some particular input. Often, this computation time depends not on the exact input but only on the "size" of the input. We usually define this time complexity to be the worst-case running time. Because the running time of a program depends on the compiler used to compile the program and the machine used to execute it, we cannot express the time in standard units such as seconds. Instead, all we can say is that the running time of a certain algorithm is proportional to the size of the input, etc. The constant of proportionality is left unspecified since it depends very heavily on the compiler, the machine, and other factors.

Big O notation is commonly used to describe algorithm running times. For instance, let the size of the input to the algorithm be n . We thus use $O(1)$ to mean computing time is approximately independent of the size of the input, or bounded by a constant. The term $O(n)$ means that the running time is directly proportional to n , and is called *linear time*. The term $O(n^2)$ refers to *quadratic time*; $O(n^3)$ is called *cubic time*; $O(c^n)$ is called exponential time, where c is a constant; and $O(\log n)$ and $O(n \log n)$ are called logarithmic time. Thus, programs can be evaluated

by comparing their running time functions (neglecting constants of proportionality). In the preceding notation, for instance, a program with running time $O(n)$ is better than one with running time $O(n^2)$ even though they are both polynomial-time algorithms. If you are not interested in the timing analysis, you should be able to skip the sentences and paragraphs discussing this subject and still understand the algorithms.

The timing analysis is often referred to as the *time complexity*. Similarly, the analysis of memory used to perform the search is called the *space complexity*. The *completeness* of an algorithm refers to the ability of an algorithm to guarantee obtaining a solution if one exists. If the algorithm is capable of identifying the best of several different solutions, it is referred to as being *optimal*. In this chapter, we focus on the time complexity to give readers a basic idea of how to compare various planning (or search) algorithms. We now discuss several popular algorithms. The interested reader may refer to the subject under search in [4] to see how some of the algorithms discussed here can be implemented in artificial intelligence and other computer languages.

5.2 SHORTEST PATH

We first study a popular shortest path algorithm, and then learn how to modify it to solve the route-planning problem.

5.2.1 Dijkstra's Shortest Path Algorithm

One of the main representatives of the shortest path algorithm is Dijkstra's algorithm [5,6]. This algorithm uses a greedy technique often employed in optimization problems. Greedy algorithms make locally optimal choices at each step in the hope that these choices will produce a globally optimal solution. Given a weighted and directed graph, similar to Figure 5.1, Dijkstra's algorithm maintains a set of nodes (e.g., small circles with node numbers in them) whose minimum costs (e.g., labeled weights between nodes) relative to the origin node are already known. Initially, this set contains only the origin node, with all other nodes being "remaining nodes." At each step, we add to the set one remaining node whose cost relative to the origin node is as small as possible. Assuming all segments have non-negative costs, we can always find a shortest path route from the origin node to the remaining node that passes only through nodes in the set. Call such a route "special." At each step in the algorithm, we use an array to record the length of the shortest route to each node. Once the set includes all nodes, all routes are "special," so this array will hold the shortest distance from the origin to each node. This algorithm bears some similarity to the breadth-first search algorithm, which is roughly analogous to

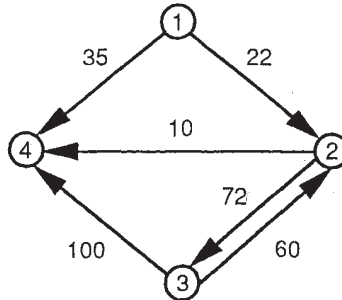


Figure 5.1 Directed graph with costs.

level-by-level traversal of an ordered tree [1,7]. It is easy to convert a road network to a directed graph similar to Figure 5.1.

One necessary condition for using this algorithm is that the cost must be non-negative for each directed segment, which is always true in vehicle navigation. A directed segment means that each road segment has a travel direction associated with it. Provided this non-negative cost condition is satisfied, Dijkstra's algorithm will find optimal solutions from the origin node to every other node. In our example, this means that this algorithm will find minimum-cost routes from node 1 to nodes 2, 3, and 4, respectively.

Let n stand for the total number of nodes and s for the total number of segments in the road network. The running time for the entire Dijkstra algorithm is $O(n^2 + s) = O(n^2)$. The equality results from the fact that we can always find constants c_1 , c_2 , and c_3 , which satisfy $c_1n^2 + c_2s \leq c_3n^2$, and constants are always ignored in Big O notation. Therefore, the running time of Dijkstra's algorithm is quadratic.

Let b stand for the *branching factor*, which is the average number of segments starting at each node. If the road network is sparse, that is, b is much less than n ($b \ll n$), it is practical to organize the nodes into a priority queue with a binary heap [1]. The resulting algorithm is called a modified Dijkstra algorithm. For vehicle navigation problems, it is easy to satisfy $b \ll n$ because the number of intersections in the problem domain is usually much larger than the average number of streets entering each intersection. The total running time of this algorithm is $O((n + s)\log n)$ or $O(s \log n)$ if all nodes are reachable from the origin node. Another modification is to implement the priority queue with a Fibonacci heap. As a result, the total running time spent in this version of Dijkstra's algorithm is $O(n \log n + s) = O(n \log n)$. In other words, using better data structures reduces the quadratic running time of the Dijkstra algorithm to logarithmic time. Further reduction of the running time has also been reported by using other data structures such as a radix heap [8].

5.2.2 Modified Shortest Path Algorithm

For single-vehicle navigation, it is not necessary to find the optimal route from the origin node to every other node in the road network. Dijkstra's algorithm can be modified to terminate when it has found the optimal route to the destination node. Listed later is a double linked-list implementation of the modified shortest path algorithm (or uniform-cost search [9]). Each node in the list has a back pointer that points to its predecessor (parent) node and one or more next pointers that point to its successor (children) nodes or a null. Although this algorithm is not very popular for route planning, it is still beneficial to discuss it, so that the reader can compare it with other algorithms and become familiar with the associated terminology. We will see that the other algorithms discussed in this chapter are basically further improvements of this algorithm. Note that in general there are a variety of implementations for each algorithm. When studying these algorithms, try to understand the algorithm itself and how it can be used to solve the route-planning problem rather than any particular implementation technique.

As mentioned in Chapter 2, the road network is represented by a digital map database. A route-planning algorithm needs to operate by searching through this directed road network, in which each node represents an intersection or a dead-end point of a road. Assume that in the algorithm we convert the node in the map database into a node data structure containing at least x and y coordinates, current planning status, and some road attributes, an indication of its promise as an intermediate routing point, a parent link that points back to the "best node" from which it came, and a next link that points to the next node in one of two linked lists, OPEN and CLOSED:

OPEN: a list of the nodes that have been generated but not yet expanded.

CLOSED: a list of the nodes that have been expanded.

The term *generate* means to create a data structure corresponding to a particular node, and the term *expand* means to generate all successors (children) of a node. This data structure should include all information required for planning as discussed briefly in the last paragraph. Figure 5.2 shows an example of generated and expanded nodes on the respective lists, where b and c are successors of a . A black dot indicates the head of each list. An electrical ground signal indicates the end of the list. In the actual implementation, a pointer to null is used. To simplify the figure, the nodes have been represented using small circles, and the data structure and pointer arrows associated with each node have been ignored. The letters in each circle indicate the name of the list, OPEN or CLOSED.

An evaluation function (cost function) is used to estimate the merit of each node generated. This function enables the algorithm to search the more promising routes first. The evaluation function is defined for node n as $g(n)$, where $g(n)$ is a measure of the actual cost of getting from the origin node to the current node n . It

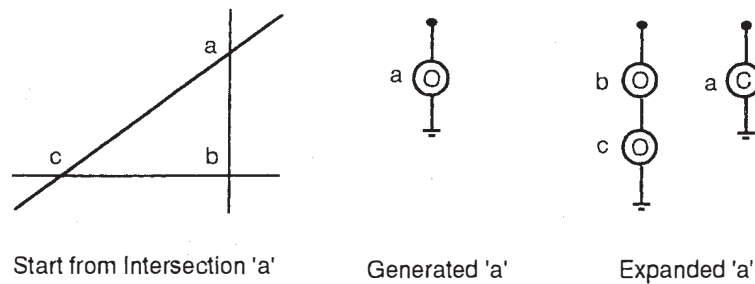


Figure 5.2 Linked lists with generated and expanded nodes for road intersections.

is up to the system designer or user to determine the travel cost used for the evaluation function as discussed in Section 5.1.

Modified Shortest Path Algorithm

1. Let the initial OPEN list contain only the origin node with zero cost (g value) and the CLOSED list be empty. Let the costs of all other nodes be infinity.
2. If there are no nodes on the OPEN list, report failure. Otherwise, select the node on the OPEN list with the lowest cost (g value) and call it BEST. Remove it from the OPEN list and place it on the CLOSED list. See if BEST is a destination node. If so, go to step 3. Otherwise, generate the successors of BEST based on its connecting segment attributes contained in the map database. For each such successor node n , perform the following steps:
 - 2a. Compute the cost of n : $g(n) = \text{cost of BEST} + \text{cost from BEST to } n$.
 - 2b. If n matches a node already in OPEN, check to see if node n has a lower cost (g value). If so, replace the matching node's cost with the cost of node n , and set the matching node's back pointer to BEST.
 - 2c. If n matches a node already in CLOSED, check to see if node n has a lower cost (g value). If so, replace the matching node's cost with the cost of node n , set the matching node's back pointer to BEST, and move the matching node to OPEN.
 - 2d. If n is not already in either OPEN or CLOSED, set node n 's back pointer to BEST and place node n in OPEN. Repeat step 2.
3. From BEST, traverse the back pointer up to the origin node and report the solution route.

This algorithm will find a "best" route from a given origin to a given destination. The same techniques used in Dijkstra's algorithm can be used to reduce the running time of the modified shortest path algorithm. Another interesting fact is

that if all segments have equal cost, the algorithm becomes a breadth-first search. In our discussion, we have ignored the case where an origin or a destination may not coincide with any node on the map. Every node is associated with a latitude and a longitude as discussed in Chapter 2. The user interface and positioning system can help determine the latitude and longitude of the origin and destination. We can then find the closest node in the map database to either the origin or destination without much difficulty.

In this algorithm, we check each successor node to see whether it is on either the OPEN or CLOSED list already in order to avoid having a node appear more than once. It is possible that more than one route may reach a destination passing through a same particular node or intersection. The fact of a node encountered more than once during the search, i.e., a recent successor node is already on the OPEN or the CLOSED lists, reflects in reality that multiple routes exist via this node. One of these routes should have a minimal cost. This check guarantees that the optimal route is always kept in memory.

Another technique often used in route planning is to include forbidden states (or drivability flags) in the data structure for each node [10]. In general, any road network will contain a fair number of dead-end roads, blocked roads, or one-way roads. We cannot use them as traversable roads for route planning. The same is true for traversing a one-way road in the wrong direction. A simple way of dealing with this problem is to dynamically incorporate this information into forbidden states for the corresponding node data structures. Forbidden nodes cannot be used in construction of the search tree, so the system automatically avoids forbidden roads. One alternative is to avoid generating a successor to a valid node for the OPEN list if the node is connected to a forbidden segment. We can further expand this idea to enhance the route-planning function. For instance, a user may want to avoid expressways for a particular trip. If the system is designed in such a way that the user can make this choice before executing the planning algorithm (one of the planning criteria), the system can then mark all expressways as forbidden to satisfy the user's request.

There is one difference between the usage of the shortest path algorithm and usage of the modified version described above in vehicle navigation applications. The shortest path algorithm is often used to find the solution for a given graph, or to precompute routes off-line for storage in memory (see Section 5.6 for examples), while the modified version and other algorithms are used when real-time navigation is required. When a navigation system is up and running, the user expects the best route to be presented almost instantaneously. Although we can view a large, detailed road network as a graph from the vehicle-navigation point of view, it is impractical (given current technology) to load the entire road network into main memory for route planning. The common practice is to load, on demand, the partial network likely to be used during the planning process into memory. We do not have the entire graph to work with, especially during real-time computation. We can imagine,

then, that we are searching a tree with the origin node as the root and the rest of the generated nodes as leaves or descendants (see the example tree shown later in Figure 5.5). Therefore, a different measurement function is needed to evaluate the running time of the algorithm.

Let d stand for the *search depth* of the shortest solution route from the origin node to a destination node. The search depth is the number of levels in a tree a search must traverse in order to find a solution. For instance, if we treat the root as level 0, all successors (leaves, children) of the root are at level 1 of this tree. In this notation, the running time of the modified shortest path algorithm becomes $O(b^d)$. On completion of the planning process, the total number of nodes on the tree in main memory should be approximately equal to b^d .

The modified shortest path algorithm takes more time and space to execute than a heuristic search algorithm. In the next section, we introduce a heuristic search algorithm. A simple example is discussed, and the final search tree, final OPEN list, and final CLOSED list are shown so that potential users can become more familiar with this popular algorithm.

5.3 HEURISTIC SEARCH

A heuristic search is an informed search strategy, which means that the search has information concerning the number of steps or the cost from the initial state and current state to the destination state. Examples of informed search methods include the best-first search, the memory bounded search, and iterative improvement algorithms such as the hill-climbing search and simulated annealing [11].

5.3.1 A* Algorithm

The most popular heuristic search used in route planning is the A* algorithm [12], which is a best-first search method. The A* algorithm has been used in fields other than vehicle route planning, such as robotics [13] and other areas where minimum-cost solutions are required.

The reason that we must do a search is that we do not know exactly which way to go. The situation would be much better if we knew in advance. Therefore, heuristic information is introduced to improve the efficiency of the search process. A heuristic is a rule of thumb, a technique for improving the efficiency of a search process by possibly sacrificing claims of completeness [14]. Heuristic information can help to determine which is the "most promising" node, which successors to generate, and which irrelevant search branches to prune. The purpose of using a heuristic is to provide an estimate of how far a node is from the destination node so that the system can determine how likely that particular node is to be on the best solution route. Heuristics are good to the extent that they point in the right direction,

but they may temporarily or occasionally lead us to search an inappropriate node, such as a dead-end street, for a particular trip. Despite this temporary problem, the process will still lead to the destination node if such a route exists. Using good heuristics may enable us to obtain good solution routes in less time using less memory space.

The A* algorithm is an extremely useful technique for implementing a best-first search of a road network. If the evaluation function selected satisfies a certain property, explained below, the A* algorithm is guaranteed to find an optimal route, if one exists, while searching less space than Dijkstra's algorithm or the modified shortest path algorithm [15,16]. To understand this savings, we use a circle to represent roughly the abstract space expanded in the modified shortest path algorithm (or Dijkstra's algorithm), and an egg-shaped area to represent roughly the abstract space expanded in the A* algorithm. Thus, the space examined in the two algorithms can be represented as shown in Figure 5.3.

A good way of using heuristic information is to calculate a heuristic function that evaluates every node generated to determine its goodness or badness. In this way, having the heuristic function suggest which route to traverse first when more than one route is available will make the search as efficient as possible. In the A* algorithm, this heuristic evaluation function enables the algorithm to search the most promising nodes first. The evaluation function $f'(n)$ for node n is defined as:

$$f'(n) = g(n) + h'(n)$$

where $g(n)$ is a measure of the actual cost of getting from the origin node to the current node n , and $h'(n)$ is an estimator for the minimum cost to get from the current node n to the destination node. Note that the primes in this equation represent

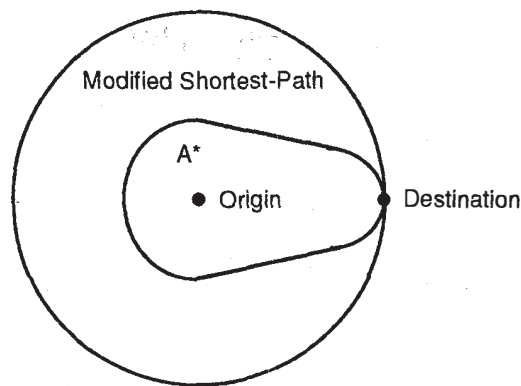


Figure 5.3 Spaces examined in the modified shortest path and A* algorithms.

This is called the *admissibility property*. As we see from our sample evaluation function, overestimation of $h(n)$ can easily be avoided because the function $b'(n)$ underestimates the cost for any node n . Therefore, provided the heuristic function is carefully selected, the admissibility property is readily achieved. The A* algorithm can generally be shown to be optimal, because it generates the fewest nodes in the process of finding the solution to a problem. Since all route costs and cost estimates are positive because of the shortest-route cost criterion and the admissibility property, the algorithm will find an optimal route for the vehicle, provided that one exists.

A* Algorithm

1. Let the initial OPEN list contain only the origin node with zero cost (g value) and the CLOSED list be empty. Let the costs of all other nodes be infinity.
2. If there are no nodes on the OPEN list, report failure. Otherwise, select the node on the OPEN list with the lowest cost (f' value) and call it BEST. Remove it from the OPEN list and place it on the CLOSED list. See if BEST is a destination node. If so, go to step 3. Otherwise, generate the successors of BEST based on its connecting segment attributes contained in the map database. For each such successor node n , perform the following steps:
 - 2a. Compute the cost of n : $g(n) = \text{cost of BEST} + \text{cost from BEST to } n$.
 - 2b. If n matches a node already in OPEN, check to see if node n has a lower cost (g value). If so, replace the matching node's cost with the cost of node n , and set the matching node's back pointer to BEST.
 - 2c. If n matches a node already in CLOSED, check to see if node n has a lower cost (g value). If so, replace the matching node's cost with the cost of node n , set the matching node's back pointer to BEST, and move the matching node to OPEN.
 - 2d. If n is not already in either OPEN or CLOSED, set node n 's back pointer to BEST, place node n in OPEN, and compute the heuristic evaluation function (cost) for node n : $f'(n) = g(n) + b'(n)$. Repeat step 2.
3. From BEST, traverse the back pointer up to the origin node and report the solution route.

For completeness, we have described the entire algorithm. On carefully examining the A* algorithm, we see that the only difference between this algorithm and the modified shortest path algorithm is that the evaluation function has been a heuristic one. To highlight the difference, we have marked it with a vertical bar in step 2d. The modified shortest path algorithm is actually a special case of the A* algorithm with $b'(n) = 0$.

To understand the algorithm better, let us see how the A* algorithm builds a search tree for the simple road network shown in Figure 5.4, where each intersection is represented by an English letter.

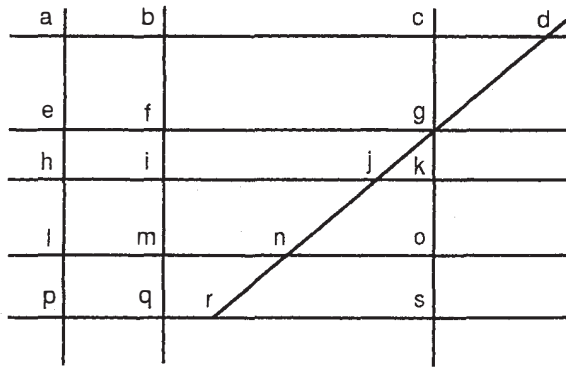


Figure 5.4 A simple road network.

For this trip, assume that the origin is at *m* and the destination is at *g*. On completion of the route-planning process, the final search tree will appear as shown in Figure 5.5. As before, we use a small circle to represent each node, while ignoring the data structure and pointer arrows associated with the nodes. We use a straight line to represent both the upward pointers (back pointer) pointing from a node to its predecessor when applicable and the downward pointers (next pointer) pointing down to the successors for a node. The letters in the circles imply that a node on the search tree is either in an OPEN list or in a CLOSED list. Although each node on the tree could appear more than once as a successor, it can occur only once on any one of the lists, since the checking steps in 2b and 2c eliminate the redundant nodes. This multiple appearance of nodes in the tree is not difficult to understand,

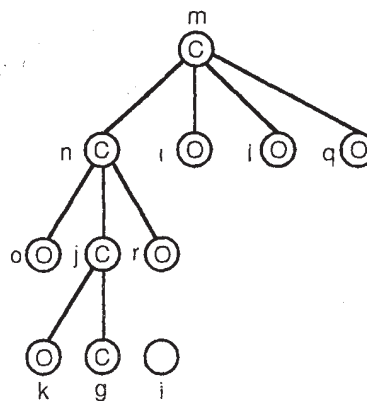


Figure 5.5 Final search tree for the A* algorithm.

since there might be different routes to the destination, which pass through the same intersection. Note that this did not occur in our simple example. What does occur is that the algorithm never checks the final successor i and places it on the tree because it has already found the destination node g . On traversing the series of back pointers, the best route turns out to be $m-n-j-g$.

The final OPEN and CLOSED lists for this example are shown in Figure 5.6. As before, a black dot indicates the head of the list and an electrical ground signal indicates the end of the list. In the actual implementation the end of the list is indicated by a pointer to null. The OPEN list is arranged in such a way that the minimum-cost node is always at the top of the list.

As before, let b stand for the branching factor, which is the average number of segments starting at each node. Let d stand for the search depth of the shortest solution route from the origin node to the destination node; A* algorithm has running time $O(b^d)$. An initial glance suggests that the running time in Big O notation is the same as that for the modified shortest path algorithm. In reality, the heuristic information used in the A* algorithm will in general reduce the time complexity, because the value of b for the A* algorithm should be smaller than that for the modified shortest path algorithm. After all, this is the main reason that we are using the heuristic approach. As in the modified Dijkstra algorithm, the running time of the A* algorithm can be further reduced by using better data structures.

Because of the increasing popularity of research and development in the field of vehicle navigation, a large number of experimental studies are available for various route-planning algorithms [17–19]. There is no doubt that these activities will continue because route planning is one of the key components of a vehicle navigation system.

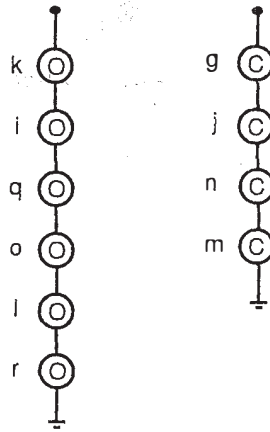


Figure 5.6 Final OPEN and CLOSED lists for the A* algorithm.

5.4 BIDIRECTIONAL SEARCH

Our understanding of the shortest path and heuristic search algorithms will now enable us to generalize these two algorithms to obtain two bidirectional search algorithms. In our discussion of the previous algorithms, we always assumed that the algorithm searched for the minimum-cost route from a given origin node to a destination node. This is called a *forward search*. A *backward search* from the destination node to the origin node should actually yield the same results, as long as the cost for each segment is the same. This leads to the idea of a bidirectional search.

The idea of a bidirectional search algorithm was first proposed in [20], formalized in [21], and further analyzed and expanded in [15,22]. The basic procedure for a bidirectional search is to compute the minimum-cost route from the origin node to the destination node (forward search) at the same time as the minimum-cost route from the destination node to the origin node (backward search). If the search algorithm is running on a single processor, the algorithm must switch quickly between the forward and backward searches. A bidirectional search algorithm may reduce the search space by at least 50%, so that the running time of this algorithm may be much shorter than that of a unidirectional search. The general relationship between the search space examined by the modified shortest path algorithm and that examined by the bidirectional search algorithm is shown in Figure 5.7.

Successful implementation of the bidirectional search algorithm requires two additional conditions (assuming a single processor is used): first, a criterion for stopping the search, and second, a criterion for alternating between the forward and backward searches. For instance, we can define a stopping criterion stating that the first node must be reached from both search directions and define a static alternation

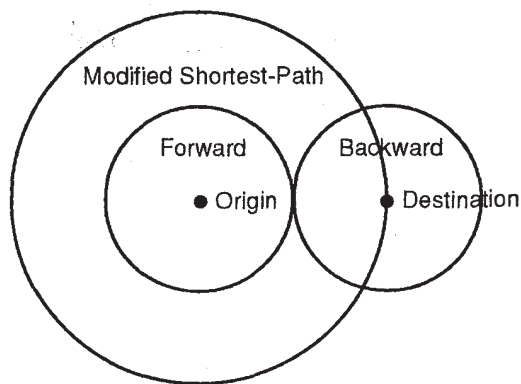


Figure 5.7 Spaces examined by the modified shortest path and bidirectional search algorithms.

criterion requiring 10 iterations in each direction. If these criteria are constructed properly, the bidirectional search will usually examine fewer nodes and compute a solution in less time than the unidirectional search.

Bidirectional Search Algorithm

1. Let the initial forward OPEN_f list contain only the origin node with zero cost (g value) and the forward CLOSED_f list be empty. Let the costs of all other nodes be infinity.
2. Let the initial backward OPEN_b list contain only the destination node with zero cost (g value) and the backward CLOSED_b list be empty. Let the costs of all the other nodes be infinity.
3. Alternate the forward and backward search directions based on a static or dynamic criterion. If there are no nodes on the OPEN list for the selected direction, try the opposite-direction list. If both OPEN lists are empty, report failure. Otherwise, select the node on the OPEN list with the lowest cost (f' value) and call it BEST. Remove this node from the OPEN list and place it on the CLOSED list. See if BEST is a node that satisfies a stopping criterion. If so, go to step 4. Otherwise, generate the successors of BEST based on its connecting segment attributes contained in the map database. For each such successor node n , perform the following steps:
 - 3a. Compute the cost of n : $g(n) = \text{cost of BEST} + \text{cost from BEST to } n$.
 - 3b. If n matches a node already in OPEN, check to see if node n has a lower cost (g value). If so, replace the matching node's cost with the cost of node n , and set the matching node's back pointer to BEST.
 - 3c. If n matches a node already in CLOSED, check to see if node n has a lower cost (g value). If so, replace the matching node's cost with the cost of node n , set the matching node's back pointer to BEST, and move the matching node to OPEN.
 - 3d. If n is not already in either OPEN or CLOSED, set node n 's back pointer to BEST and place node n in OPEN. Repeat step 3.
4. From BEST, traverse the forward back pointer up to the origin node, the backward pointer up to the destination node, link these lists together and report the solution route.

Recall that the running time for the unidirectional search (modified shortest path algorithm) is $O(b^d)$. The bidirectional search has running time $O(2b^{d/2}) = O(b^{d/2})$. From this calculation, we see that a bidirectional search will explore fewer nodes than a unidirectional search, as shown in Figure 5.7. This leads to a fast computation time for the bidirectional search, verified experimentally in [15].

Similarly, a bidirectional heuristic search can also be used. In Figure 5.8 the abstract spaces explored by the unidirectional and bidirectional heuristic search algorithms are shown.

If we search in both directions, we have the bidirectional heuristic search algorithm listed below. As in Section 5.3, the difference is highlighted by a vertical bar in step 3d.

Bidirectional Heuristic Search Algorithm

1. Let the initial forward $OPEN_f$ list contain only the origin node with zero cost (g value) and the forward $CLOSED_f$ list be empty. Let the costs of all other nodes be infinity.
2. Let the initial backward $OPEN_b$ list contain only the destination node with zero cost (g value) and the backward $CLOSED_b$ list be empty. Let the costs of all the other nodes be infinity.
3. Alternate the forward and backward search directions based on a static or dynamic criterion. If there are no nodes on the OPEN list for the selected direction, try the opposite-direction list. If both OPEN lists are empty, report failure. Otherwise, select the node on the OPEN list with the lowest cost (f' value) and call it BEST. Remove this node from the OPEN list and place it on the CLOSED list. See if BEST is a node that satisfies a stopping criterion. If so, go to step 4. Otherwise, generate the successors of BEST based on its connecting segment attributes contained in the map database. For each such successor node n , perform the following steps:
 - 3a. Compute the cost of n : $g(n) = \text{cost of BEST} + \text{cost from BEST to } n$.
 - 3b. If n matches a node already in OPEN, check to see if node n has a lower cost (g value). If so, replace the matching node's cost with the cost of node n , and set the matching node's back pointer to BEST.
 - 3c. If n matches a node already in CLOSED, check to see if node n has a lower cost (g value). If so, replace the matching node's cost with the cost

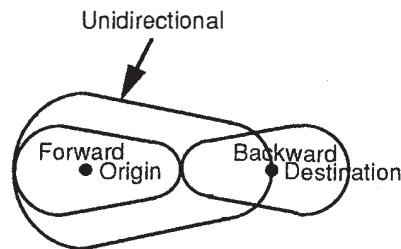


Figure 5.8 Spaces examined by unidirectional and bidirectional heuristic search algorithms.

- of node n , set the matching node's back pointer to BEST, and move the matching node to OPEN.
- 3d. If n is not already in either OPEN or CLOSED, set node n 's back pointer to BEST, place node n in OPEN, and compute the heuristic evaluation function (cost) for node n : $f'(n) = g(n) + h'(n)$. Repeat step 3.
 4. From BEST, traverse the forward back pointer up to the origin node, the backward pointer up to the destination node, link these lists together and report the solution route.

Ideally, the bidirectional search should meet at the middle (Figures 5.7 and 5.8). As discussed in [15], if there are several different paths from the origin to the destination, the search may not meet in the middle, as shown in Figure 5.9. The two searches may each find nearly complete paths before intersection occurs, or one may find a complete path and the other almost a complete path before they intersect. In other words, there may be duplication of effort. In the worst case, such as an improper termination criterion for a heuristic evaluation function that considers only the estimated travel cost, a bidirectional heuristic search could double the work of a unidirectional heuristic search. Therefore, the choices of termination criterion and heuristic evaluation function are very important for a proper implementation of a bidirectional heuristic search.

As discussed in the section on the heuristic search, the A* algorithm is capable of finding an optimal route if the admissibility property is satisfied. A bidirectional heuristic search must explore almost twice as many nodes as a unidirectional heuristic search to determine the optimal route. The reason is that a bidirectional search must compare all the connected routes during the search to ensure the route is optimal. In real life, human beings seldom need to search for an optimal solution during their daily activities. A near-optimal solution is often a satisfactory solution for many problems. If an optimal solution is not required, the bidirectional search is a very good candidate for route planning. In other words, the optimality constraint should be relaxed when using a bidirectional search algorithm. By relaxing the optimality

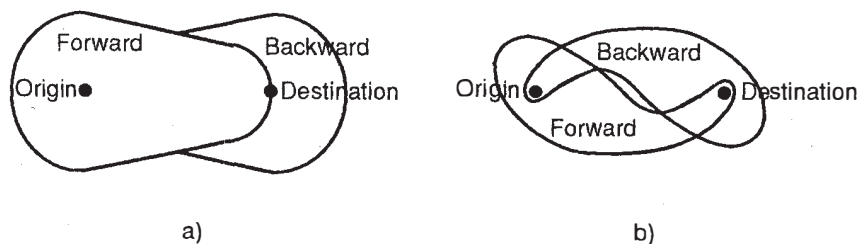


Figure 5.9 Spaces examined by bidirectional heuristic search algorithms with (a) a poor termination criterion and (b) a poor heuristic evaluation function.

constraint, the bidirectional search will find a near-optimal route long before the optimal solution is reported.

There are several different variations of the bidirectional search algorithm that can be used to speed up the search process. In the algorithms discussed earlier, the travel cost from the origin or destination is used to determine which routes (nodes) need to be further expanded. One method involves instead the use of the number of nodes expanded to make such decisions. This method results in selecting routes generated from the origin with about the same number of nodes as the routes generated from the destination. Another method involves using a weighting factor for each road segment during the search, such that a higher weighting factor is used for segments close to the straight line between origin and destination than for segments further away from this line. The algorithm thereby forces the search process to select a route that is close to the straight line connecting the two starting nodes. Yet another variation is to use road type or class to reduce the amount of unnecessary searching. This operates on the assumption that every road needs to be examined in the areas near the origin and destination, but beyond these areas only major roads need be considered in the search. Clearly, its operation reduces the total number of nodes examined. This leads naturally to our next topic of discussion.

5.5 HIERARCHICAL SEARCH

There is a natural hierarchy of roads, with expressways at the top and side streets at the bottom. This leads to the idea of a more efficient, hierarchical search algorithm for a given road network. The basic idea is to first perform a search in an *abstraction space* rather than in the entire original problem space. An abstraction space is a simplified representation of the problem space in which unimportant details are ignored. The search then fills in the details from the original space based on the results obtained in the search on the abstraction space. Choosing a simplified representation of this type leads to a significant increase in problem-solving ability and efficiency.

Human beings are well aware of the value of abstraction. For instance, when solving a very complex problem, we often first ignore the low-level details and concentrate on the essential features of the problem. We then fill in the details. This idea is easily generalized to multiple hierarchical layers of abstraction such that each layer contains a different level of detail. Because the computer was developed to help human beings, it is very natural for the problem of abstraction to have been addressed in the field of computer science, especially artificial intelligence.

The main reason that abstraction reduces the complexity of problems is that the total complexity becomes the sum of the complexities of the individual searches, rather than the product of the complexities [23]. This method has been verified experimentally to be very effective in reducing the search space for complex

problems [24]. The problem of abstraction was analyzed in detail and formalized in [25]. To make analysis of the hierarchical search easier, first assume that the running time for a given search algorithm is $O(b^d)$. For a single layer of abstraction, the hierarchical search can reduce a search algorithm with running time $O(b^d)$ to $O(b^{d/2})$. For multiple layers of abstraction, the optimum abstraction hierarchy for a problem space with b^d states (in our case nodes) will have $\ln b^d$ layers of abstraction. The size ratio for successive layers of abstraction is e . This type of optimum abstraction hierarchy can reduce the search time from $O(b^d)$ to $O(d \log b)$. In other words, it has been found [25] that the optimum hierarchy has a logarithmic number of layers with a constant size ratio, and that this hierarchy reduces the search from being exponential in the number of states in the problem space to being linear. Certain search algorithms have a performance problem in that the number of nodes to be expanded in the road network often grows exponentially. From this analysis, we see that multiple layers of abstraction can reduce exponential complexity to linear complexity. It is therefore clear that a hierarchical search can potentially reduce the time complexity of a search algorithm.

To implement a hierarchical search and to simplify the route-planning task, the map database needs to be constructed hierarchically as discussed in Chapter 2. Others have also proposed and implemented hierarchical map databases to aid route planning [16,26,27]. The higher a hierarchical database layer, the less detailed the abstracted representation. Because a higher layer abstraction space contains fewer nodes and segments than its underlying search spaces, a search in an abstract space often yields an incomplete solution (but with reduced search time). The nodes in the abstract space can be either regions in the original space or a subset of the original nodes. Clearly, during the search we need to switch between different layers to take advantage of the hierarchy. Avoiding overhead during layer switching and determining when to switch to a different layer play important roles in the algorithm efficiency.

In practice, it is very difficult to construct a database or to design an algorithm that will satisfy the conditions of optimum abstraction hierarchy as discussed. For vehicle navigation, it is natural to use road ranks to define a hierarchy for route planning. Despite the difficulty of achieving the theoretically optimum hierarchy, the bidirectional heuristic-based hierarchical search is still the most popular algorithm for route planning in a very large map database.

5.6 OTHER ALGORITHMS

Route planning can also be performed using a divide and conquer method [28,29]. The basic idea is to divide the road network into regions (blocks) and precompute optimal regional routes for each region and then find and assemble these regional routes into a complete route from origin to destination. In addition to precomputation

and prestorage of regional routes in the first stage, the algorithm must be coupled with the ability to determine entry and exit nodes for each route in each region and sequentially connect them into a final route solution in the second stage. Because all optimal routes within a region must be prestored in memory, Dijkstra's algorithm (unmodified) is generally used for precalculation. A different algorithm may be used for the second stage. Depending on how the road network is divided into regions and how the information available in the precomputed regional routes is used, various techniques can be used to provide the final solution to the vehicle user. This method can also be viewed as an approach to reconstruct the road network hierarchically (two layers in the preceding discussion) and then precompute all origin and destination pairs in the network. A similar idea can be extended to multivehicle route planning in hierarchically structured multilayered databases [30].

The real-time heuristic search algorithm proposed in [31] can be modified for route planning in certain situations [16], which can be classified as an anytime algorithm. Unlike the A* algorithm, which finds the optimal route, this algorithm finds a satisfactory route. For a real-time embedded in-vehicle navigation system and a very large map database, time constraints usually place a limit on the number of nodes that can be searched. For example, a sequential computer may need to switch to a more critical task before the entire search is completed. This fact can be represented by a *search horizon*. In a real-time heuristic search algorithm the search horizon is a search depth determined by the time available for searching the road network (or a directed graph tree during the search). This limited search horizon causes the real-time heuristic search to make a decision based on local optimality. The depth of the search forward from the current node (state) is determined by the computer resources available, that is, information provided via a recent update from the dynamic map database (containing updated dynamic traffic information), CPU time, or the time available for planning. If a more powerful CPU becomes available, the search depth can easily be increased, that is, the search horizon can be adjusted according to the available hardware and software. By contrast, neither the heuristic search algorithm nor the search algorithms discussed in this chapter place any restrictions on the search horizon to limit the use of either CPU or memory. This could lead to problems involving excessive searches. For instance, a search process may fail to report a solution route due to resource limitations, even though at least one solution route does exist.

Algorithms developed for the traveling-salesman problem can be used to plan a trip with multiple stops [32]. This study has been generalized as the vehicle routing problem [33]. Although there are a number of extensions, the basic problem involves a vehicle to be stopped at various intermediate destinations before reaching a final destination during a journey or a list of customers to be served by a vehicle or a fleet of vehicles from a single warehouse. For some service problems, each customer may place a demand for a certain quantity of goods. There are limits on vehicle capacity, on the maximum distance that any particular vehicle can travel, etc. In

short, the objective of the algorithm is to find a set of routes that minimizes the total travel cost. The vehicle routing problem is known as *NP-complete*, that is, it is unlikely that any polynomial-time algorithm exists for solving the problem. Historically, the focus has been on methods of generating a feasible solution which can then be improved using a local search. Genetic algorithms, tabu searches, and robust algorithms have recently become popular. Some of these algorithms have been expanded to cover multivehicle routing problems. For information on current trends, benchmark problems, and more references on the vehicle routing problem, see [34,35].

In recent years, interest in dynamic on-line algorithms for shortest path problems has grown. The main application domains for these algorithms involve dynamically maintaining maximum flow in a network, bipartite matching in graphs, incremental computations for data flow analysis, and interactive systems design. More information about these algorithms can be found in [36] and the references contained therein. These algorithms can be modified for use in vehicle route planning. The main goal of these algorithms is to design efficient data structures that offer fast solutions and can operate in dynamic environments where, for example, the input data change. All of the other search algorithms discussed in detail here are only capable of dealing with static input data. Given the capability of these dynamic algorithms, it should not be very difficult to adapt them for dynamic vehicle navigation under conditions where the road conditions (travel costs of road segments) are dynamically updated on the vehicle by either a traffic information center or a traffic management center.

We have now reached the end of this chapter. Before reading the next chapter, consider the following interesting questions. Assume that a bidirectional search algorithm is used in dynamic route planning. Suddenly, the travel cost of a two-way road is updated in one direction only while the route is being planned. Can this type of search still find a nearly optimum solution in reasonable time and space for any road network? On the other hand, can a dynamic algorithm do a better job planning the route in this situation?

References

- [1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, Cambridge, MA: MIT Press; New York: McGraw-Hill, 1990.
- [2] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, 6th ed., New York: McGraw-Hill, 1995.
- [3] H. A. Taha, *Operations Research: An Introduction*, 5th ed., New York: Macmillan, 1992.
- [4] M. Kantrowitz, *CMU Artificial Intelligence Repository*, <http://www.cs.cmu.edu/Web/Groups/AI/html/repository.html>, 1996.
- [5] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, Vol. 1, 1959, pp. 269-271.
- [6] E. F. Moore, "The Shortest Path Through a Maze," *Proc. International Symposium on the Theory of Switching*, Cambridge, MA: Harvard University Press, Apr. 1959, pp. 285-292.

- [7] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*, Reading, MA: Addison-Wesley, 1983.
- [8] R. K. Ahuja, K. Mehlhorn, J. B. Orlin, and R. E. Tarjan, "Fast Algorithms for the Shortest Path Problem," *J. Assoc. Computing Machinery*, Vol. 37, No. 2, Apr. 1990, pp. 213-223.
- [9] A. Barr and E. A. Feigenbaum (Eds.), *The Handbook of Artificial Intelligence*, Los Altos, CA: William Kaufmann, 1981.
- [10] Y. Zhao, "Adaptive Real-Time Route Planning for IVHS," *Fall 1989 Summary Report*, IVHS Technical Report No. 89-A, The University of Michigan, Dec. 1989.
- [11] R. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Englewood Cliffs, NJ: Prentice Hall, 1995.
- [12] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Syst. Sci. Cybernetics*, Vol. 4, No. 2, July 1968, pp. 100-107.
- [13] Y. Zhao, "Theoretical and Experimental Evaluation of a Local-Minimum-Recovery Navigation Algorithm," in *Recent Trends in Mobile Robots*, Yuan F. Zheng (Ed.), Singapore/New Jersey: World Scientific, 1993, pp. 75-117.
- [14] E. Rich and K. Knight, *Artificial Intelligence*, 2nd ed., New York: McGraw-Hill, 1991.
- [15] I. S. Pohl, "Bidirectional Search," in *Machine Intelligence*, B. Meltzer and D. Michie (Eds.), New York: American Elsevier Publishing Co., 1971, pp. 127-140.
- [16] Y. Zhao and T. E. Weymouth, "An Adaptive Route-Guidance Algorithm for Intelligent Vehicle-Highway Systems," *Proc. American Control Conference*, June 1991, pp. 2568-2573.
- [17] P. C. Nelson, C. L  n, and J. Dillenburg, "Vehicle-Based Route Planning in Advanced Traveler Information Systems," *Proc. Intelligent Vehicles Symposium*, July 1993, pp. 152-156.
- [18] L. R. Rilett, C. Blumentritt, and L. Fu, "Minimum Path Algorithms for In-Vehicle Route Guidance Systems," *Proc. IVHS America 1994 Annual Meeting*, Apr. 1994, pp. 27-36.
- [19] S. Shekhar and A. Fetterer, "Path Computation in Advanced Traveler Information Systems," *Proc. 1996 Annual Meeting of ITS America*, Apr. 1996, pp. 304-312.
- [20] G. B. Dantzig, *Linear Programming and Extensions*, Princeton, NJ: Princeton University Press, 1963.
- [21] T. A. J. Nicholson, "Finding the Shortest Route Between Two Points in a Network," *The Computer Journal*, Vol. 9, No. 3, Nov. 1966, pp. 275-280.
- [22] I. S. Pohl, "Bidirectional and Heuristic Search in Path Problems," Ph.D. dissertation, Ann Arbor, MI: University Microfilms International, 1969.
- [23] M. Minsky, "Steps Toward Artificial Intelligence," in *Computers and Thought*, E. A. Feigenbaum and J. Feldman (Eds.), New York: McGraw-Hill, 1963, pp. 406-450.
- [24] E. D. Sacerdoti, "Planning in a Hierarchy of Abstraction Space," *Artificial Intelligence*, Vol. 5, 1974, pp. 115-135.
- [25] R. E. Korf, "Planning as Search: A Quantitative Approach," *Artificial Intelligence*, Vol. 33, 1987, pp. 65-88.
- [26] T. Hashimoto, K. Hirano, and Y. Kobayashi, "Development of Route Calculation System," presented and distributed at *Intelligent Vehicles Symposium*, June/July 1992.
- [27] T. Yagyu, M. Fushimi, Y. Ueyama, and S. Azuma, "Quick Route-Finding Algorithm," *SAE Paper No. 930555*, Society of Automotive Engineers, 1993, pp. 121-126.
- [28] C. B. Harris, R. Goss, E. J. Krakiwsky, and H. A. Karimi, "Optimal Route Information System—Automatic Vehicle Location (AVL) Route Determination," *Proc. Urban Regional Information Systems Association (URISA)*, Nov./Dec. 1987, pp. 29-41.
- [29] M. Sugie, O. Menzilcioglu, and H. T. Kung, "CARGuide—On-Board Computer for Automobile Route Guidance," *Proc. 1984 National Computer Conference*, July 1984, pp. 695-706.
- [30] Y.-H. Huang, E. A. Rundensteiner, and N. Jing, "Evaluation of Hierarchical Path Finding Technologies for ITS Route Guidance," *Proc. 1996 Annual Meeting of ITS America*, Apr. 1996, pp. 340-350.
- [31] R. E. Korf, "Real-Time Heuristic Search," *Artificial Intelligence*, Vol. 42, Mar. 1990, pp. 189-211.

- [32] G. Reinelt, *The Traveling Salesman: Computational Solutions for TSP Applications*, Berlin: Springer-Verlag, 1994.
- [33] N. Christofides, "Vehicle Routing," in *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. Shmoys (Eds.), New York: Wiley, 1985.
- [34] D. J. Bertsimas and D. Simchi-Levi, "A New Generation of Vehicle Routing Research: Robust Algorithms, Addressing Uncertainty," *Operations Res.*, Vol. 44, No. 2, Mar./Apr. 1996, pp. 286-304.
- [35] T. Duncan, *The Vehicle Routing Problem*, <http://www.aiai.ed.ac.uk/~timd/vehicles/vrp.html>, 1996.
- [36] H. N. Djidjev, G. E. Pantziou, and C. D. Zaroliagis, "On-Line and Dynamic Algorithms for Shortest Path Problems," *STACS 95: 12th Annual Symposium on Theoretical Aspects of Computer Science (Proceedings)*, Berlin: Springer-Verlag, March 1995, pp. 193-204.

CHAPTER 6



ROUTE GUIDANCE MODULE

6.1 INTRODUCTION

Route guidance is the process of guiding the driver along the route generated by the route planning module. Guidance can be given either before the trip or in real time while en route. The pretrip guidance could be presented to a driver as a printout. Such a printout would be similar to travel tips provided by a travel agency, but with detailed turn-by-turn driving instructions, or just like the output generated at a kiosk near the car rental counter. These instructions might include turns, street names, travel distances, and landmarks. On the other hand, en-route guidance would require providing turn-by-turn driving instructions to a driver in real time. It is much more useful, but requires a navigable map database, an accurate positioning module, and demanding real-time software and computation power. In this chapter, we concentrate on real-time route guidance.

The route guidance module uses the outputs of both the route-planning module and the positioning subsystem to guide the vehicle on the road. In addition to a map database module, the positioning subsystem could consist of the positioning module alone or a positioning module and a map-matching module. Once a particular route has been generated by the route-planning module and the position of the vehicle has been determined by the positioning subsystem, the route guidance module needs to coordinate with these subsystems to present proper guidance to the driver.

The presentation is aided by the human-machine interface module to be discussed in the next chapter. A simplified functional diagram of the interaction between the route guidance module and the other modules is shown in Figure 6.1. All modules except for the human-machine interface module have already been discussed in previous chapters. Note that in some systems the maneuver generation task is an integral part of the route-planning module. We discuss it here primarily for purposes of exposition.

As the vehicle moves, real-time route guidance requires that the location of the vehicle as a function of time continually be compared to the best route generated by the route-planning module. The real-time route guidance system continuously updates its knowledge of the vehicle's current position and direction of travel as well as the road on which the vehicle is traveling. As a turn or a maneuver approaches, the route guidance system alerts the driver with visual signals, audible signals, or driving instructions.

6.2 GUIDANCE WHILE EN ROUTE

As indicated in Figure 6.1, route guidance consists of two tasks, a maneuver generation task and a route-following task. Note that there are many different designs and implementations for the route guidance module other than the one sketched in Figure 6.1. For instance, the maneuver generation task could be an integral part of the route-planning module. Both the maneuver generation task and route-following task might deal with the data generated by the route-planning module directly. To

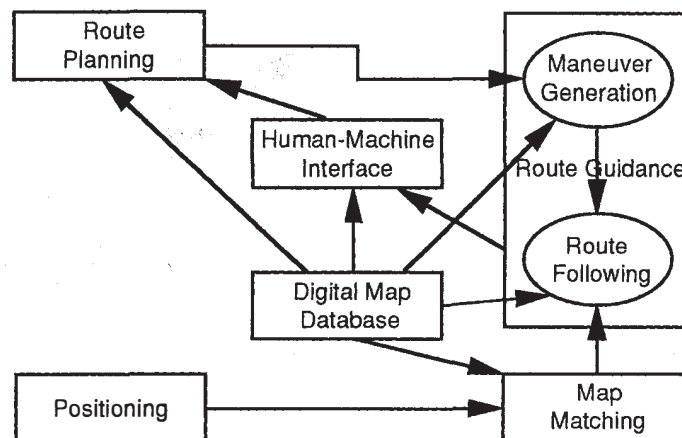


Figure 6.1 Simplified diagram showing the interaction between the route guidance module and the other modules.

convey the basic ideas, however, we use this simplified functional diagram to describe the principles and practical issues involved in this module. Many of the approaches described in this section are applicable to the other situations discussed later in this chapter.

6.2.1 Maneuver Generation

As mentioned, the maneuver generation task can be either a by-product of the route-planning module or an individual task within the route guidance module. The main function of this task is to generate a maneuver list that can be followed by the route-following task. The typical output route generated by a nonhierarchical route-planning module is a simple list of road segments that does not provide enough guidance information. The maneuver generation task can then augment each road segment in the list with appropriate maneuver and turn type information that can later be interpreted by the human-machine interface in the form of specific visible and audible instructions supplied with assistance from the route-following task. Figure 6.2 lists some samples of these types. Different combinations of maneuver and turn types may correspond to different visual and audible presentations to the driver.

If a hierarchical route plan module is being used, it may be necessary to break the output segments into the lowest layer segments before augmenting them as discussed earlier, depending on how high-layer segments are constructed. Recall that in Section 2.6 we discussed the fact that in some hierarchical map databases the

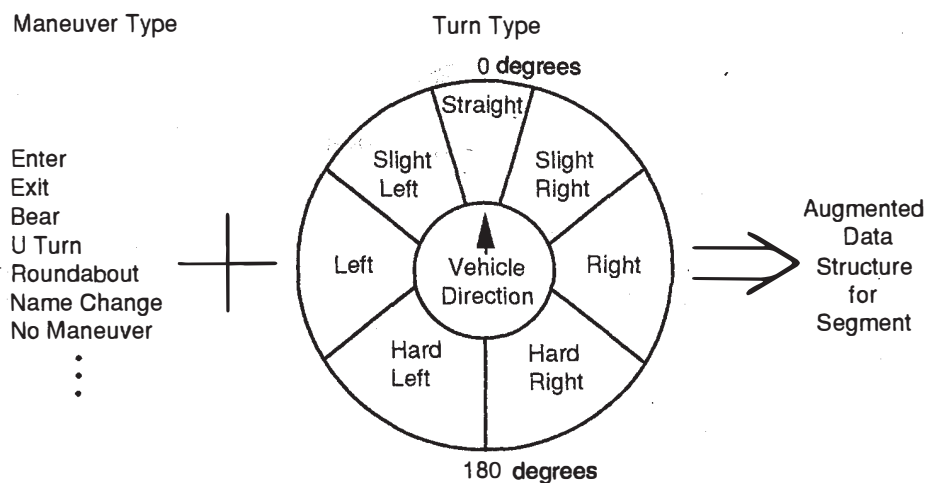


Figure 6.2 Maneuver and turn types used for augmentation of the segment data structure.

segments at higher layers are equivalent to a combination of several segments stored in the layer immediately below. To save memory, the segments in the higher layers only contain information relevant to route planning. For positioning and guidance, it is better to use the segments in the lowest layer, which contains detailed guidance information on each segment. Also, almost all vehicles use segments at both high and low levels during any given trip, which means that it is much more efficient to guide the vehicle using the lowest layer where all relevant data are stored. In fact, the data access and storage methods used during planning and guidance are critical to achieving maximum performance, particularly in a system based on a hierarchical map database.

Careful consideration should be given to memory space and data access mechanisms in order to guarantee efficient implementation of the route guidance module. The higher layer data may be in a form suitable only for planning to minimize the memory usage of the route-planning module. As we saw in Section 5.5, route planning will benefit considerably from the use of a hierarchical data representation for the map. One side effect is that extra time may be required to process the resulting route in the route guidance module. Of course, there will always be a trade-off. For a system designed to travel on a large road network, it is usually (based on current technologies) worth the trouble to do a little more work in guidance, compared with the time saved in planning. Furthermore, currently available real-time computing technology allows multiple tasks to be executed simultaneously. Once just enough maneuvers to enable initial route following have been computed, the remainder of the work can be performed in background. After all, the on-board navigation system is a real-time embedded computer application. It is much easier in terms of keeping the map-matching-based positioning simple to report the current vehicle position based on the basic layer (lowest one). In other words, a route planned on the basis of certain hierarchical databases may not be sufficiently detailed for guidance. A maneuver list based on the basic segments with sufficient detail for guidance should be generated. In a hierarchical implementation, the maneuver generation task might include breaking all higher layer segments on a route into basic road segments while adding maneuver and turn type information to the maneuver list.

If a slow secondary or remote storage device is used for the hierarchical map database, efficient use of memory space requires that we keep at least the information required for route planning in main memory so that only the information on the relevant road segments needs to be retrieved. In other words, the information required for positioning, maneuver generation, and route following should be kept separate from that needed for route planning in order to maximize planning efficiency when the database is stored on a slow device. After route planning has been completed, we need to maintain detailed information in main memory only for those route segments on the maneuver list and in the immediate vicinity of the current vehicle location. The memory required for route storage should appear similar to the body of a snake that has recently swallowed an egg, with the body around the egg

representing route storage space for the map area in the immediate vicinity of the vehicle, and the rest of the body representing the route storage space for areas that are not near the current location of the vehicle. Similar techniques have been used in mobile robot guidance [1]. To save memory space, we can also discard any data used by the route-planning module that are not necessary for route guidance.

Good data structures should be used for efficiency of data access [2,3]. These data structures should be able to help the system avoid exhaustive searches, on-demand calculations, and repeated accesses of the same data. For instance, a hash table can be used for quick and frequent look-up of the current location of the vehicle. Ideally, once one segment on the route is found, identifying the upcoming segment should be as simple as accessing a structure field or following a pointer. One possible data structure of this type is a double linked list of maneuvers with a back pointer to the last maneuver just accessed and a next pointer to the next maneuver. Enough calculated travel cost results can be stored to limit on-demand computation during route guidance. If there is insufficient main memory available to share with other navigation tasks, a *cache* may be needed to manage data access and avoid repeated requests for the same data from secondary memory. A cache is a collection of data blocks that are logically part of secondary memory, but which are kept in the main memory for performance reasons. In other words, a cache provides a high-speed buffer in main memory to act as a temporary storage area for the secondary memory. In summary, we should store as little guidance data as possible in main memory while still providing fast access and a good balance between time and space tradeoffs.

6.2.2 Route Following

On-road guidance is not difficult, provided the current position of the vehicle can be determined precisely. Once the position is known, proper signals and instructions can be prepared for or presented to the driver after comparison of the position against the planned route. This planned route consists of a sequence of road segments that leads to a destination generated under planning conditions determined either by the system design or by driver-selected planning criteria. At any point during the trip, the vehicle should be traveling on a segment contained in the planned route. The route-following task closely monitors the position of the vehicle on this segment to determine when to take proper guidance action. These actions are presented to the driver through the human-machine interface. In short, the main function of the route guidance module should be to decide when and how to guide the driver.

Once the route-following task has identified the driver's next maneuver, it must have a way to convey this information to the driver. The most popular method in current navigation systems is to provide one or a series of voice announcements to warn the driver of the approaching maneuver so that he or she can take appropriate

action (see Figure 6.3). Other methods could involve displaying the maneuver instructions on a screen either manually or automatically, providing simple audio tones, etc. The announcement can be made through a speaker using either a prerecorded human voice or a synthesized voice as discussed in Section 7.3.

One study recommends using certain message types, contents, and timings when a series of three messages is issued for each maneuver [4]. These three messages are called "early," "prepare," and "approaching." The "early" message warns the driver of an upcoming maneuver (this message should always be given). The "prepare" message informs the driver to move to the appropriate lane and begin searching for the street, landmark, or exit (this message may be omitted if there is no time to announce it). The "approaching" message instructs the driver to perform the maneuver (this message is a final reminder and should always be given). The content and timing of each message are listed in Table 6.1 [4]. To facilitate study and compare with other approaches, we have rounded off the distances in meters and provided approximate equivalents in miles.

Another study proposes the following timing for maneuver messages [5-7]:

- For nonexpressways (or urban areas), 300m before next maneuver and 700m if lane changes are necessary;
- For expressways (or rural areas), 500m before entrances and 2,000m before junctions and exits.

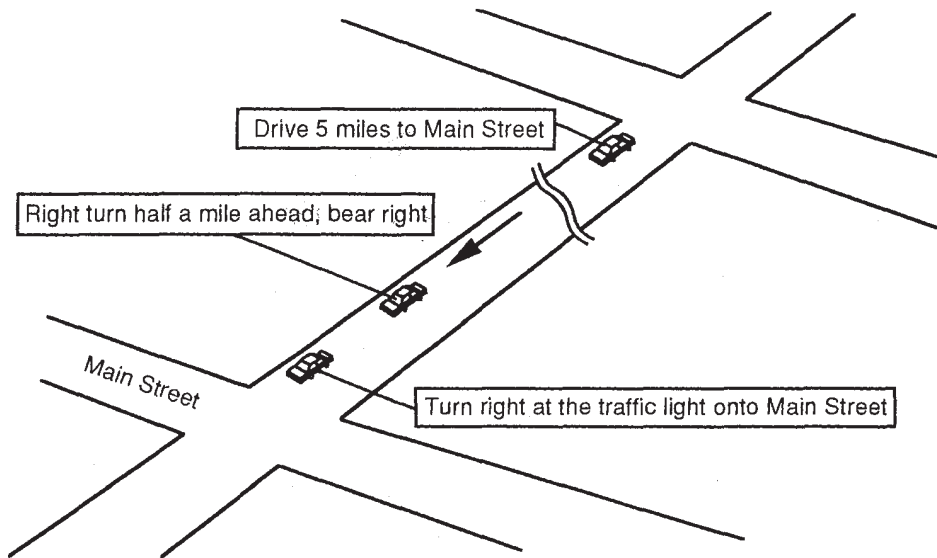


Figure 6.3 Maneuver announcement during the journey.

Table 6.1
Content and Timing for Maneuver Messages

<i>Message</i>	<i>Content</i>	<i>City Timing</i>	<i>Highway Timing</i>
Early	"In" {distance} "at" {location} {action}	5 sec after turn	15 sec after turn
Prepare	{distance} {location} {action}	1500m	3000m
	or	(0.932 miles)	(1.865 miles)
	{distance} {landmark} {location} {action}	Before turn	Before turn
Approaching	"Approaching" {location} {action}	150m	500m
	or	(0.093 miles)	(0.311 miles)
	"Approaching" {landmark} {location} {action}	Before turn	Before turn

If two maneuvers are too close to one another, messages for both intersections will be given before the first intersection [8]. If the next maneuver follows the turn by a large distance, an "early" message will be given after the turn to warn the driver. If the driver presses the Verbal Guidance switch and a message containing turning information for the next maneuver is announced. Note that the two proposals discussed here consider only two speeds, that is, city roads and highway roads. This may not be sufficient for a broader range of road types.

Many of these message announcement (maneuver) timing proposals are based on experience gained from actual route guidance systems. From these proposals, we can see that these maneuver timings depend on the road type and distance to or after the maneuver. This implies that the road type, intersection type, travel speed and distance, and number of lanes are important factors to be considered when designing the maneuver timing of a route guidance module. Other reasoning techniques (such as a fuzzy-logic-based approach [9]) may also be used to combine all relevant factors and determine the best timing. Furthermore, the exact timing of advance messages such as the "early" and "prepare" messages is not critical, but the timing of turn messages such as the "approaching" message is. If the maneuver instructions are stored in a slow secondary memory device, a pre-fetching technique or cache may be required to make them available as an anticipated maneuver approaches. Whichever approach is used, the main purpose of this subsystem is to avoid announcing upcoming maneuvers either too early or too late. In this way, the driver can benefit from the instructions to increase safety and achieve maximum performance.

6.3 GUIDANCE WHILE OFF-ROUTE

Drivers may occasionally drive off the planned route for various reasons. They may make a wrong turn at an instructed intersection, may be unable to change to the

correct lane for an intersection, may temporarily get off the route, etc. A mechanism to get them back on route must exist once the system has determined that the vehicle is no longer on the route (or not traveling on any road segment within a predetermined vicinity).

Different methods can be used to solve this problem. A simple solution is to alert the driver that the vehicle is off its route once this scenario is confirmed. A directional arrow then displays the direction toward the original destination on the screen. The driver should use these rough directions to get back to the original road or find a way to reach the destination. Another solution is to notify the driver that the vehicle has gone off-route and that it needs to replan a new route to the original destination. Replanning a short route from the current off-route location to the original route is another option. The next on-route segment following the segment where the vehicle went off-route could be used as an intermediate destination in this circumstance. One could also replan a route using the original destination. The tricky part of this solution is how to determine a new origin for the route-planning system. If the vehicle is stopped after the off-route occurred, the current location of the vehicle can be used as the origin. Otherwise, we must select an origin that is ahead of the current travel direction of the vehicle. One good candidate is the node that will be reached in a travel time equal to the maximum amount of time assumed necessary to replan the new route.

All of the methods discussed assume that the vehicle is off the planned route but still on road. How do we guide an off-road vehicle? Clearly, an off-road vehicle is off the planned route too. One strategy is to notify the driver that the vehicle is off the road and that it needs to be driven back to the nearest road. The system keeps monitoring to see whether the vehicle has returned to the road network. Once it determines that the vehicle is on the road again, the preceding methods can be used to guide the vehicle again. We can also try to direct the driver back to the nearest road by pointing in that direction on the screen or determining and displaying the nearest road segment to which the driver should return. There are many other approaches in addition to those mentioned here that can also be used as off-road recovery mechanisms.

6.4 GUIDANCE WITH DYNAMIC INFORMATION

In the previous discussion, we implicitly assumed that the route guidance module was based on a static route-planning module. A system based on static route planning uses the static travel cost of each road segment to calculate the minimum-cost route and to guide the vehicle along this route. As mentioned in Chapter 5, this cost may involve distance, travel time, minimum number of turns, etc. This static cost may be a value derived from a prestored map database that is independent of the time of day or a statistical value that has been predetermined for this particular segment.

during a certain time period. The static cost is an oversimplified assumption. In the real-world environment, the actual cost is a dynamic value that depends on traffic conditions at a specific time. Therefore, a system based on dynamic route planning is preferable from the point of improved route guidance. Ideally, the route used for guidance would be calculated on the basis of the real-time cost of each particular road segment. To be more specific, this dynamic travel cost should be based on the link cost, which is the segment cost plus the turning cost received during the trip from the traffic control center or traffic management center. Remember that the cost of left turns, right turns, turns at stop signs, and turns at traffic lights all depend on the specific road.

Similar to the route-planning module, dynamic route guidance (or dynamic navigation) can be classified into either multivehicle (system-wide) dynamic route guidance, which guides all vehicles on the road network while minimizing the total travel cost of all the vehicles, or single-vehicle dynamic route guidance, which guides one vehicle while minimizing the travel cost of this particular vehicle. We discuss only the single-vehicle dynamic route guidance case in this section.

Assume that the system in the vehicle is capable of receiving, filtering, and storing information received from a control center or information center (such as a traffic information center). The route guidance module must take appropriate actions in response to incoming congestion reports for the planned route.

In dynamic route guidance systems, we must face three important issues with regard to the system configuration: First, we need to determine whether the dynamic (and some of the static) information should be processed by the central system (host end) or the on-vehicle system (mobile end); second, we need to determine the dynamic information update frequency; and, third, we need to determine the communication medium to be used in ensuring smooth information flow between the infrastructure and vehicle. The communication medium is discussed in Chapter 8. In this section, we limit ourselves to a configuration in which the static travel information is stored on board and the dynamic information is provided periodically by a traffic control center or traffic management center. We assume that the information update rate occurs frequently enough to reflect the traffic conditions on the road and route planning is done in the vehicle. In this way, the guidance activities are the sole responsibility of the mobile vehicle but not the centralized host. In Chapter 10, this is classified as a distributed navigation approach. We now discuss one dynamic route guidance algorithm that could be used for this configuration (Figure 6.4). Note that some algorithms discussed in Section 5.6 can also be used. Similar approaches can be used for other configurations.

The main idea of this algorithm is to rely on the dynamic travel cost as much as possible for route guidance. To achieve this goal, the on-board navigation computer must be able to accept the dynamic cost data and consequently change the static database into a dynamic one. A simple method involves maintaining a

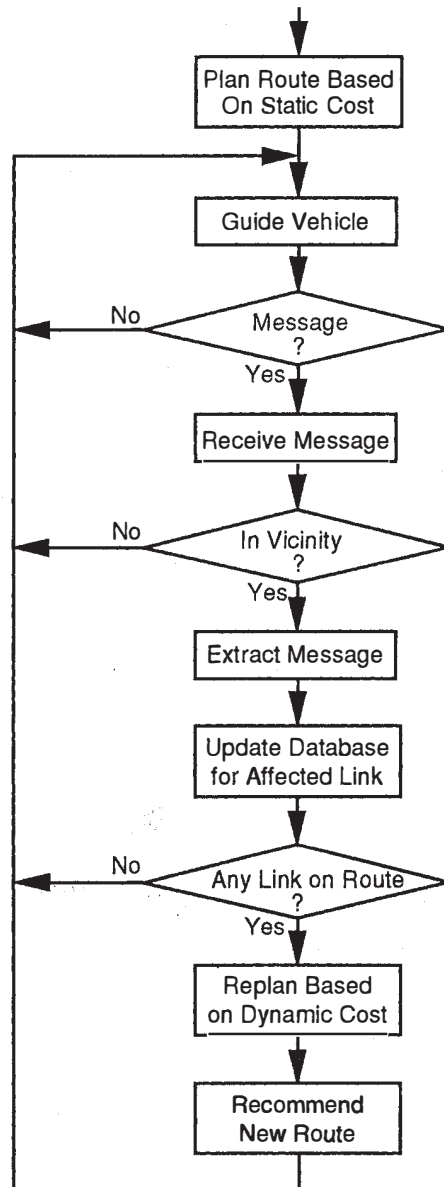


Figure 6.4 A dynamic route guidance algorithm.

fixed static database on board. During the route guidance process, this database can be dynamically updated to reflect the new cost information.

Assume that the travel time is used as the route-planning cost. As we learned in Chapter 2, the length and speed limit of a road segment are attributes used in the static map database. These attributes can be easily converted into travel time for route-planning purposes as discussed in Chapter 5. With the travel time as the cost, the static route-planning system can then generate a minimum-travel-time route for the driver. This route is derived under the assumption that an average driver usually drives near the speed limit on every road segment. We know now that a better way of providing the travel cost information is to use a static database that more accurately reflects actual travel conditions because vehicles do not always travel at the allowed speed limit.

One example of this type involves integrating a historical profile with the database. This profile is originally obtained in an off-line process. Recall that each link represents a road segment plus information on turning delays. This process uses a detailed representation of a traffic flow model (supported by the road network and other data) to estimate link travel times and flows for origin-destination trip tables at each time of day [10]. The 24-hour day is divided into different periods so that the travel time for each link can be estimated appropriately. The model also incorporates traffic delay functions (which include turning movements and signal control parameters). When incorporated with the real implementation, these link travel times are validated and updated using actual data received from vehicles acting as traffic probes or from other traffic sensors and anecdotal information. Probe vehicles are vehicles that report back real-time traffic data for the roads on which they are traveling. This historical profile can be used as one of the inputs to the database compiler described in Section 2.6. After compilation, the link travel time stored in the profile becomes an attribute for the navigable database. When used in conjunction with a route guidance algorithm, it is hoped the system will present a realistic route. The historical profiles can be further improved at periodic intervals after system deployment on the basis of new traffic data, provided this feature has been included in the system. Despite this improvement over using the speed limit as the travel cost, the route recommended by the system will be still based on static costs prestored in a database. Only when a dynamic cost is used does the route guidance system become truly dynamic.

The algorithm described in Figure 6.2 first uses the static costs to plan a route. While the driver is following the planned route, the on-board system continually monitors the incoming message broadcast by a traffic control center or traffic management center. For simplicity, assume that the message contains travel link cost data for the area in which the vehicle is traveling. When the system receives a message, a filter is used to determine whether the message contains information for links in the vicinity of the vehicle. If the message contains such data, the affected links in the static database are updated using the new link cost information. The

updated data can be placed in main memory (as shown in Figure 2.10 in Chapter 2) for fast storage and retrieval. This is not only convenient for maintenance of the dynamic database, but also beneficial for efficient route planning and guidance. If one or several of the affected links happen to fall on the portion of the preplanned route that the vehicle has not yet traversed, the route-planning module is automatically triggered to calculate (in the background) an alternative route based on these dynamic costs. If this alternative route is better than the current route, the driver is notified that a shorter route has been identified. A criterion for a route being better than the current route (such as saving of at least 20% in travel time relative to the current route) can be predefined. The driver can then accept or reject this recommendation. If the driver accepts this alternative route, the system will discard the current route and guide the driver along the new one.

Some of the implementation details for this dynamic route guidance algorithm have been omitted here. These include, but are not limited to, defining the incoming message frequency, defining what is meant by "in the vicinity of the vehicle," and defining the threshold for determining whether the travel time saved is worth alerting the driver about. The representation used for the road network and the dynamic model to be used are other areas that need to be addressed during implementation. Field validation and updating will be much easier if the representation of the road network and dynamic model are relatively accurate. Many representations and models have been proposed for estimating or predicting link travel times. These have been designed both for the normal situation and abnormal situations involving incidents or congestion. Various representations and models for the road network are described in [10-13]. See Chapter 11 for a working example of a dynamic route guidance system.

From common sense, dynamic route guidance should be able to direct drivers to less congested roads. Can you imagine what would happen if vehicles all equipped with the same dynamic guidance unit travel on a highly congested road network? If all these vehicles receive the same traffic information, they could be dispatched to the same, previously uncongested road. This road may very quickly become congested and make the traffic even worse. This phenomenon is known as the Braess paradox [14]. Clearly, this is the one situation that a traffic management system or a dynamic route guidance system must avoid. When this happens, it is not a good idea to direct all the vehicles to a single less-congested road.

The Braess paradox is caused by providing the same traffic information to all vehicles within a particular road network. Since dynamic route guidance has not been implemented on a widespread basis, most studies on this phenomenon are based on computer simulations. Some strategies have been proposed for dealing with this problem [14,15].

One strategy involves adding distortion factors to the link cost data. By doing this, we hope that different link cost data will be used by different vehicles in the area. From the last chapter, we know that when the costs are different, the route-

planning algorithm will generally generate different routes. Clearly, guided by different routes the chance of congesting a single road is greatly reduced. However, if the link costs are distorted, we also know that the planning algorithm will not be able to find the optimal route in the absence of congestion. One issue still remains: selection of distortion factors for maximum efficiency of the road network. Another strategy involves broadcasting forecast link-cost data to the vehicles in the road network. Each vehicle reports its planned route to the traffic center. Based on the reported information, the center updates the forecast link-cost data and broadcasts it periodically. However, this requires each vehicle to have its own route planner. It also requires that the vehicle or infrastructure be able to provide enough communication bandwidth to send the information back to the center. Other strategies can also be used. More research as well as practical experience is needed in order to obtain enough data for further evaluation of these different strategies.

Before leaving this chapter, let us consider two questions. What is the best method for determining the new origin when replanning the route of a moving vehicle that has left the previously planned route? Besides the message announcement timing method introduced in this chapter, are there any other good methods for determining the best time to announce a maneuver instruction? As design engineers, we certainly do not want to surprise our drivers with any unexpected announcements, do we?

References

- [1] Y. Zhao, C. V. Ravishankar, and S. L. BeMent, "Coping with Limited On-Board Memory and Communication Bandwidth in Mobile-Robot Systems," *IEEE Trans. Syst. Man Cybernetics*, Vol. 24, No. 1, pp. 58-72, Jan. 1994.
- [2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*, Reading, MA: Addison-Wesley, 1983.
- [3] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, Cambridge, MA: MIT Press; New York: McGraw-Hill, 1990.
- [4] P. Green, W. Levison, G. Paelke, and C. Serafin, *Preliminary Human Factors Design Guidelines for Driver Information Systems*, Ann Arbor, MI: The University of Michigan, Transportation Research Institute, Technical Report No. UMTRI-93-21. Also published in McLean, VA: U.S. Dept. of Transportation, Federal Highway Administration, Report No. FHWA-RD-94-087, Dec. 1995.
- [5] K. Kimura, K. Marunaka, and S. Sugiura, "Human Factors Considerations for Automotive Navigation Systems," *Proc. Triennial Congress of the International Ergonomics Association*, Vol. 4, Aug. 1994, pp. 162-165.
- [6] H. Kishi and S. Sugiura, "Human Factors Considerations for Voice Route Guidance," *SAE Paper No. 930553*, Society of Automotive Engineers, 1993.
- [7] E. J. Krakiwsky and R. L. French, "Japan in the Driver's Seat," *GPS World*, Oct. 1995, pp. 53-60.
- [8] T. Ito, S. Azuma, and K. Sumiya, "Development of the New Navigation System—Voice Route Guidance," *SAE Paper No. 930554*, Society of Automotive Engineers 1993.
- [9] Y. Zhao, A. M. Kirson, and L. G. Seymour, "Fuzzy Logic Applications in Vehicle Control and Navigation," *Abs. Third World Congress on Intelligent Transport Systems*, Paper #1225 (full paper in CD-ROM), Oct. 1996, p. 204.

- [10] D. E. Boyce, A. Tarko, S. Berka, and Y. Zhang, "Estimation of Link Travel Times with a Large-Scale Network Flow Model for a Dynamic Route Guidance System," *Proc. IVHS America 1994 Annual Meeting*, Apr. 1994, pp. 37-43.
- [11] J. F. Gilmore and N. Abe, "Neural Network Models for Traffic Control and Congestion Prediction," *IVHS Journal*, Vol. 2, No. 3, 1995, pp. 231-252.
- [12] B. Ran, H. K. Lo, S. Weissenberger, and B. Hongola, "Predicting Dynamic Travel Times under Incidents in Transportation Networks," *Proc. IVHS America 1994 Annual Meeting*, Apr. 1994, pp. 37-43.
- [13] V. Sethi, N. Bhandari, F. S. Koppelman, and J. L. Schofer, "Arterial Incident Detection Using Fixed Detector and Probe Vehicle Data," *Transportation Res. C*, Vol. 3, No. 3, Apr. 1995, pp. 99-112.
- [14] L. R. Rilett and M. W. Van Aerde, "Modeling Distributed Real-Time Route Guidance Strategies in a Traffic Network that Exhibits the Braess Paradox," *Proc. IEEE-IEE Vehicle Navigation and Information Systems Conference (VNIS '91)*, Society of Automotive Engineers, Oct. 1991, pp. 577-587.
- [15] M. Tokoro and S. Takaba, "Route Guidance Strategy Under High Penetration Condition," *Proc. IEEE Vehicle Navigation and Information Systems Conference (VNIS '94)*, Aug./Sep. 1994, pp. 303-308.

CHAPTER 7



HUMAN-MACHINE INTERFACE MODULE

7.1 INTRODUCTION

The human-machine interface is a module that provides the user with the means to interact with the location and navigation computer and devices. This subject is often treated under ergonomics or human factors engineering in the literature. Researchers in the field of ergonomics conduct research and apply information about human behavior, abilities, limitations, and other characteristics to the design of tools, machines, systems, tasks, jobs, and environments [1]. The objectives are to increase productivity and safety, to maximize performance, and to ensure comfortable and effective human use. The reader may become familiar with various human factors problems and how to deal with them in general reference [2]. Information on automotive human factors engineering can be found in [3] and references contained therein. In addition to human factors for vehicle location and navigation systems, this chapter also addresses the basic principles of popular interface technologies.

To develop a successful human-machine interface, a certain procedure must be followed that may include identification of requirements, determination of functions to be supported, specification of interface type(s), selection of controls and displays, and, finally, designing and implementing these interfaces. We will concentrate on the various control devices and display technologies often used in vehicle

location and navigation systems. These control devices and display technologies form an integral part of the human-machine interface module.

The primary design principles for a good human-machine interface module are as follows [4]:

1. The interface design should be consistent.
2. Controls and displays should function the way people expect them to function.
3. Controls and displays should be arranged in text-based order: from left to right, and from top to bottom.
4. The interface should minimize the need for the user to remember.
5. Operations that occur most often or have the greatest impact on driving safety should be the easiest to perform.
6. Controls, displays, and information elements that are used together should be adjacent to one another.

In short, a good design is often based on good engineering judgment and accepted human factors practice. It should stand the test of time and experience.

Every interface needs one or more control devices. Table 7.1 lists various control devices along with evaluation scores and recommended usage ([5], mostly from [6-8]). The evaluation scores are based on the following criteria:

- Usability under vibration conditions;
- Usability from multiple locations;
- Space required;
- Operability over desired temperature range;
- Any built-in display properties;
- Usability in noise;
- Susceptibility to dirt, grease, etc.;
- Hands-free use.

Whenever the control has one of the properties listed, it receives a score of 1. Otherwise, it receives a score of 0. The highest score should be 8. The vibration conditions described above mean those normally encountered inside an automobile traveling at highway speeds on a reasonably maintained concrete or asphalt roadway. The anticipated temperature range is approximately -34.4° to $+54.4^{\circ}\text{C}$ (-30° to $+130^{\circ}\text{F}$).

From Table 7.1, we see that there is great variety of different computer and display control devices from which to choose. Although several devices have very similar scores, these scores result from different combinations of the criteria [5]. The key is to understand the advantages and disadvantages of each device and know when it should be used to achieve maximum performance while meeting proper safety guidelines. (For discussions of safety and driver performance under various

Table 7.1
Various Control Methods

<i>Control</i>	<i>Score</i>	<i>Recommended Usage</i>
Foot push button	6	For use only when both hands occupied
Isometric joystick	6	For applications requiring return to center after each entry
Isotonic joystick	6	For controlling various display functions
Key-operated switch	6	For prevention of unauthorized machine operation and support of on/off functions
Keyboard	4	For use when vehicle is completely stopped (score for sealed keyboards)
Knob (continuous)	5	For use when low force or precise adjustment of a continuous variable is recorded
Legend switch	6	For display of qualitative information or other types of information when space is limited
Light pen	2	Use when vehicle is completely stopped and only for noncritical inputs
Push button	5 or 6	Support simple switching between two conditions
Rocker switch	6	As an alternative to toggle switches when two discrete positions are required
Rotary selector switch	6	For discrete functions when three or more detented positions are required
Slide switch	6	For two or more discrete positions
Toggle switch	6	For two discrete positions where space limitations are severe
Touch screen	5	For low training when targets are large, discrete, with less or no text input
Voice recognition	6	When the user's hands or eyes are busy

human-machine interfaces, see Section 7.2.3, [9–11], and the other references listed at the end of this chapter.)

In the following sections, we first consider the commonly used interfaces based on visual displays. We then discuss several potentially useful voice-based interfaces.

7.2 VISUAL-DISPLAY-BASED INTERFACES

A visual display is an electronic device that converts electrical signals into a visual image in real time for direct interpretation by a human observer. It serves as the visual interface between human and machine. The display itself is typically an electro-optical device of some kind. It presents information within a fraction of a second from the time it was received, and continuously holds the information on the output device until new information is received. The image is created by making visible contrast patterns.

7.2.1 Display Technologies

For engineers, it is a challenging task to design or select a visual display for vehicle use because of the severe environment associated with such effects as varying light conditions, varying temperature, frequent vibration, and constant dirt and grease. In addition to these environmental constraints, we expect the display to be lightweight and have low power consumption. Because of the unique safety requirements, attention requirements, body multiplexing requirements, and visual environment in a vehicle, many human factors must also be considered.

Research suggests that the following functions can be supported using visual displays (modified from [12]):

- A wide range of quantitative and qualitative information can be displayed.
- Complex information can easily be extracted from the display.
- The display can emulate conventional displays such as paper maps.
- In-vehicle or heads-up displays become possible.

However, it is distracting if too much information is displayed or the information displayed is too complex. While driving, minimal information should be presented. This leads to the important subject of how to design the graphical user interface (GUI). Before discussing it, we first take a look at the design guidelines for the display interface used for GUI.

Many important factors need to be considered when selecting and using display technologies, such as display color, contrast, luminance, format, content, size, labeling, orientation, and placement. As discussed in [5] and the references contained therein, there should be no more than six symbol colors. Commonly accepted color conventions should be observed. Pure blue or pure red and pure blue displayed simultaneously on a dark background should be avoided. The user should be able to adjust the luminance level. The minimum monochrome contrast ratio (higher/lower) should be 1.5 for acceptable legibility and 2.5 for comfortable reading. The display should be positioned properly and the area surrounding the display screen should be matte black to avoid glare. The display luminance should range anywhere from 70 cd/m^2 to 150 cd/m^2 , measured after any optical effects. It should be adjustable over a range of at least 50:1 (highest/lowest), preferably 100:1. The total number of operating hours until the display luminance is reduced to 50% of its original maximum luminance should be at least 5000 hr. The visibility envelope for the display should be no more than ± 30 degrees to 40 degrees laterally from the design eye reference point and no more than 30 degrees above or 5 degrees below the design eye reference point. Interested readers may refer to [4,5] and the references contained therein for further discussion of other guidelines to help in the selection and evaluation of various technologies.

Based on the visual display characteristics and design guidelines just discussed, it becomes obvious that designing a safe and efficient GUI is another challenge for engineers. A variety of designs are possible, depending on the user specifications and the display devices used. These designs all have a very similar goal, that is, to simplify a variety of complicated functions so that the user will not be intimidated and can actually easily obtain useful information while driving. The interested reader is urged to refer to Chapter 11 and [4,13,14] for examples of actual and proposed user interfaces, as well as ITS-related magazines, conferences, and exhibitions to learn and see more about various different interfaces. A performance evaluation of different designs for touch screen keyboards can be found in [15], and the basic touch screen technologies are addressed in the next section. A more general discussion on GUI design can be found in [16].

Several different technologies are available for display devices. Table 7.2 lists various display devices, along with evaluation scores and recommended usage (mostly from [5]). The evaluation score is based on the following criteria:

- Salience in low ambient illumination;
- Salience in high ambient illumination;
- Salience in high ambient noise;
- Attention getting properties;
- Visibility from multiple locations;
- Usability if eyes are busy;
- Ability to display graphics;
- Ability to display same object in multiple colors;
- Space required;
- Operability over desired temperature range;
- Flexibility of formatting.

Whenever the display has one of the properties listed, it receives a score of 1. Otherwise, it receives a score of 0. The highest score should be 11. The anticipated temperature range is the same as in Table 7.1, that is, from approximately -34.4° to $+54.4^{\circ}\text{C}$ (-30° to $+130^{\circ}\text{F}$).

Early in-vehicle display screens generally used cathode-ray tube (CRT) displays. The inner screen of a CRT is coated with phosphors that emit light when struck by one or more beams of high-energy electrons. The electron beams are emitted from a cathode at the rear of the tube. On a raster-scanned CRT display, the electron beam sweeps across the entire display on each refresh cycle and illuminates the proper points as it comes to them. On a vector-scanned CRT display, the electron beam strikes only the points needed to be illuminated on the display. The color of the image is controlled by the selection of phosphors and electron guns.

An electroluminescent display (ELD) produces light from a polycrystalline phosphor on application of an electric field. The phosphors are sandwiched inside

Table 7.2
Various Display Methods

<i>Display</i>	<i>Score</i>	<i>Recommended Usage</i>
Analog/mechanical	6	For qualitative and quantitative value or qualitative value only
CRT	7	For text and graphics applications
Counter/mechanical	6	For presenting large ranges of quantitative data
ELD, matrix addressed	6	For text and graphics applications
HUD	7	To present an image overlaid on a view of an outside scene and to decrease visual transitions from "head-down" displays to the outside, forward scene
Indicator lights	5	To indicate system, equipment, and/or control condition
LED, indicator	5	For simple indications
LED, character based	5	For text applications
LED, matrix addressed	5	For text and graphics applications
LCDs	4-6	For text only or text and graphics applications. Use transfective LCDs when visibility in sunlight, low power consumption, and visibility of display from multiple viewer positions are more important than having a multicolor, high resolution display. Use transmissive LCDs when display brightness, display resolution, and multicolor display are more important than having visibility in sunlight, low power consumption, and visibility of display from multiple viewer positions.
Reflective, character based	5	
Reflective, matrix addressed	5	
Transfective, character based	6	
Transfective, matrix addressed	6	
Transmissive, character based	4	
Transmissive, matrix addressed	6	
PDP, character based	4	For text applications
PDP, matrix addressed	6	For text and graphics applications
VFD, character based	6	For text applications
VFD, matrix addressed	8	For text and graphics applications

a grid of wires. Most ELDs are monochromatic, with a typical color being yellow on a black background. ELDs are often used as matrix-addressed text and graphics displays and as backlights for liquid crystal displays (LCDs). The main advantages of ELDs are ruggedness and ability to be viewed from any angle. The main disadvantages are higher production cost and limited multiple color capability.

A heads-up display (HUD) consists of an image source, projection optics, and a combiner. The image is generated in the image source and projected through collimation optics onto the combiner. It is viewed as an image overlaid on the forward visual scene. The forward windshield of the vehicle is often used for the HUD combiner. A vacuum fluorescent display (VFD) or a vector-scanned CRT is generally used as the image source.

A light-emitting diode (LED) produces light via injection of electrons into a solid-state semiconductor. An LED consists of a single p - n junction, which is a

semiconductor device. An LED emits light when the junction is forward biased so that minority carrier injection and electron-hole recombination occur. Different semiconductor materials emit different colors after electron injection.

A plasma display panel (PDP) uses the physical properties of a gas discharge. A cathode glow is produced when electrical current passes through small pockets of gas. Plasma displays can generally be divided into two major families, direct-current and alternating-current plasma displays. The color is typically orange on a black background.

A vacuum fluorescent display (VFD) produces light using rapidly moving electrons and phosphors in a vacuum under the control of a mesh grid. The VFDs are typically blue-green on a black background. Different colors are produced by combining different phosphors and using a wide range of filters.

An LCD is essentially a "light valve." The principle of the LCD is different from all of the other display devices discussed thus far. The LCD does not produce light but modifies it. When an electric field is applied, the alignment of the liquid crystal molecules in the display is altered. This alignment controls the polarization of light passing through the liquid crystal material by either blocking the light or allowing the light to pass through. LCDs can be classified into one of three types: reflective (ambient light), transmissive (backlit), or transreflective (integration of transmissive and reflective). Both monochrome and color LCDs are available. Different addressing techniques have been developed, such as fast scanning, two-frequency addressing, hysteresis multiplex addressing, thermal addressing, and matrix addressing. The most popular addressing methods are passive matrix addressing and active matrix addressing. The first method uses a grid of wires (switches), whereas the second uses one to three transistors per pixel.

The low power consumption and compact nature of LCDs are major advantages. LCDs typically have power consumption of 2 to 3W (much less than a CRT). Lower driving voltage contributes to low power consumption in LCD products and to cost reduction of driving integrated circuits (ICs).

There are two types of color LCDs: Thin-film transistor (TFT) active matrix (AM) LCDs, and super-twisted nematic (STN) LCDs. At the same time, new breakthroughs are under way in development of glassless LCDs and low-temperature polysilicon TFT devices. A new method of liquid crystal patterning (which uses polarized ultraviolet light for patterning [17]) has been touted as very successful in increasing the viewing angles. Future LCD panels will see much improvements over the ones used in vehicles today. These improvements include but are not limited to the following [18]:

- Increasing the viewing angle by orientation control over the orientations of the liquid crystals: One such method involves using a multiple-domain technology to divide each pixel into two parts to double the viewing angle to ± 40 degrees;

- Reducing the amount of reflection from the screen surface by using a black matrix and antiglare polarized panels;
- Reducing power consumption by developing better light-transmissive color filters, a wider aperture ratio, and highly efficient backlighting systems while improving brightness;
- Maintaining a lightweight, low profile by introducing highly reliable, high-density connections, a thinner backlight system, and lightweight materials;
- Development of a shock-proof structure by using new materials and production processes.

We now turn to flat panel displays. Flat panel displays are flat and light, and do not require a great deal of power. The ideal flat panel display is expected to be thin, have low volume, have an even surface, have high resolution, have high contrast, be readable in sunlight, display color, have low power consumption, and be solid state and lightweight. One example of this type of flat panel display is the field-emitter display (FED). Research and development on this technology have been very strong recently because LCDs, the technology with the largest market share, still produce an inferior image and have restricted viewing angles compared to CRTs.

The basic principle of the FED is similar to that of the CRT. In a CRT, there is one hot cathode that sprays electrons against the inside of the screen surface. In contrast, FEDs have millions of cold microtips that can be fired to turn pixels (picture elements) on and off (see Figure 7.1, where ITO stands for indium tin oxide). The FED has row and column conductors separated by an insulator layer. One conductive layer is the gate (extraction grid) and the other is the cathode (emitter). Each pixel is defined by an area where two conductive layers cross. Each pixel may have several (up to several hundred) microtips. Unlike active-matrix LCDs, which take one to three transistors to activate each pixel, the FED requires no transistors. A simple

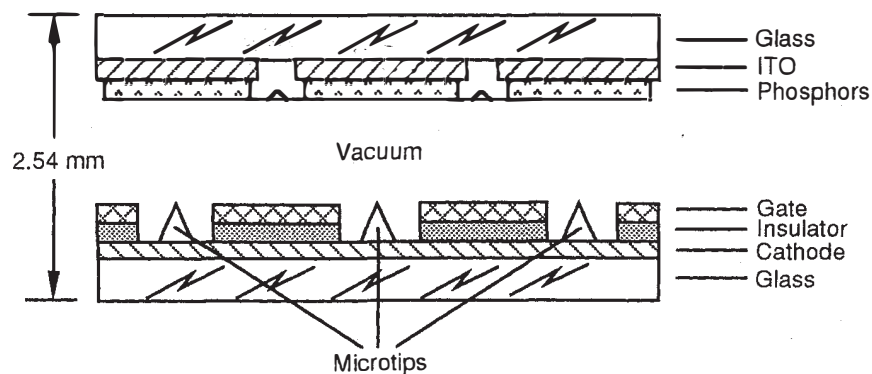


Figure 7.1 Simplified cross-sectional view of FED layers.

wire grid powered by an electric voltage drives the firing of the microtips. Therefore, FEDs require less power than LCDs for an equivalent brightness. They look sharp from any angle, and are about 2.54 mm (one-tenth of an inch) thick, or about half the thickness of an LCD. FEDs have better or equivalent brightness, contrast, and resolution compared to both STN and AM LCDs. Furthermore, FEDs are 56% thinner, 65% lighter, have a 35% larger viewing area, and consume three to five times less power [19].

Interested readers will find a detailed discussion of the major visual display technologies (and flat panel displays) discussed in this section and the latest developments in [20–22] as well as the relevant display technology publications.

7.2.2 Touch Screens

Recent years have seen a steady increase in the amount of electronic equipment on vehicles, and the number of switches on the vehicle dashboard has made vehicles more complicated to operate. Reducing the number of input and output devices has therefore become an important design issue. One solution is to use a touch screen (panel), as a multifunction transparent switch to help simplify the input and output devices used in the vehicle.

As summarized in [12], using touch screens as control devices enables implementation of the following functional features:

- The ability to control a large number of functions from a small area;
- A potential reduction in the proliferation of conventional controls;
- Uniquely suited to provision pointing input in vehicles;
- Natural for menu selection;
- Best suited to occasional, low-resolution pointing.

To facilitate the usage of touch screens, general guidelines for the use of touch screens are provided in [5] and the relevant references therein. The touch-sensitive areas of the display should be indicated to the user. These areas should be at least 1.9 cm square and at least 0.32 cm apart. A touch screen should have sufficient luminance transmission so that the display is clearly readable in its intended environment and meets the 70 to 140 cd/m^2 luminance requirement, measured after any optical effects. Feedback should be provided to indicate touch screen actuation. A force-actuated touch screen should restrict the operating force to a value between 0.25 and 1.5 N. Only one touch command at a time should be accepted within approximately 100 ms, even though some technologies are capable of accepting more than one touch at the same time.

Touch screens are control devices. These screens are unique in that they can be overlaid on any flat CRT, LCD, ELD, or PDP panel. They are operated by the

touch of a finger or stylus. The currently available technologies include resistive, capacitive, infrared, and acoustic wave. Since the resistive technology has the biggest market share, we discuss it in further detail in order to describe how it works and we also explain other implementation issues involved. Resistive touch screens are different from traditional devices because they are transparent, resistive, membrane switches. These screens are manufactured from thin, transparent, electrically conductive polyester films. Resistive touch screens are available in flat or curved configurations with either analog (continuously resistive) or matrix formats.

A matrix-format screen is divided into a grid of individual switches (or cells, see Figure 7.2). Touching an individual cell activates the corresponding switch. Matrix touch panels are relatively inexpensive, have relatively longer lives, and have lower cost electronic circuitry. However, the resolution is discrete rather than continuous.

The analog-format screen is not divided into a grid, but has a continuous resolution (Figure 7.3). The location on the screen that was touched is measured in both the X and Y directions. The resolution is determined by the decoding hardware and software. Analog touch panels have high resolution, provide quick response, and can be triggered by either nonconductive or conductive probes.

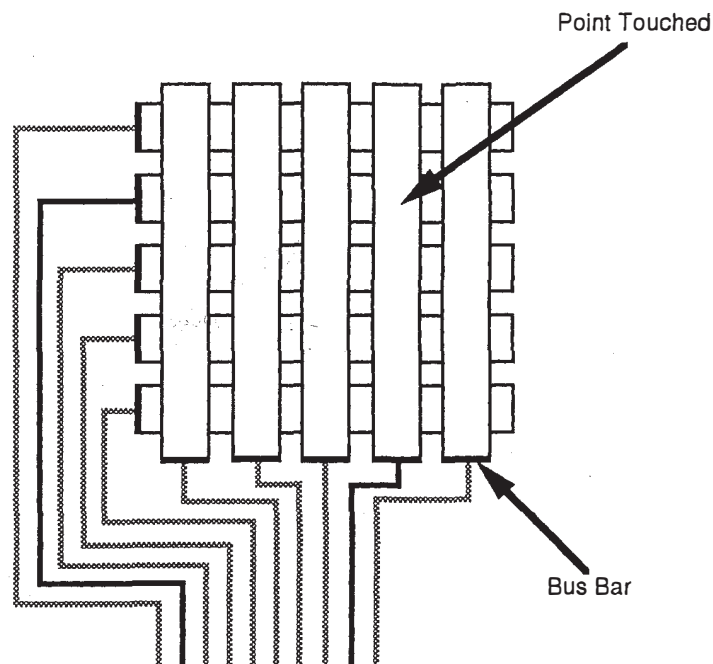


Figure 7.2 Matrix-format touch panel.

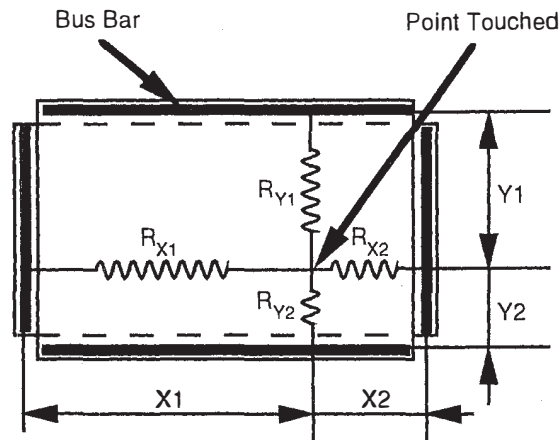


Figure 7.3 Analog-format touch panel.

Even though touch panels look very thin, they consist of many different layers. Figure 7.4 shows a simplified cross section through a touch panel on top of an LCD (in this diagram, ITO stands for indium tin oxide and PET stands for polyester). In our example, the two opposing transparent resistive (or conductive) layers consist of PET with sputtered ITO and glass with sputtered ITO separated by an insulating

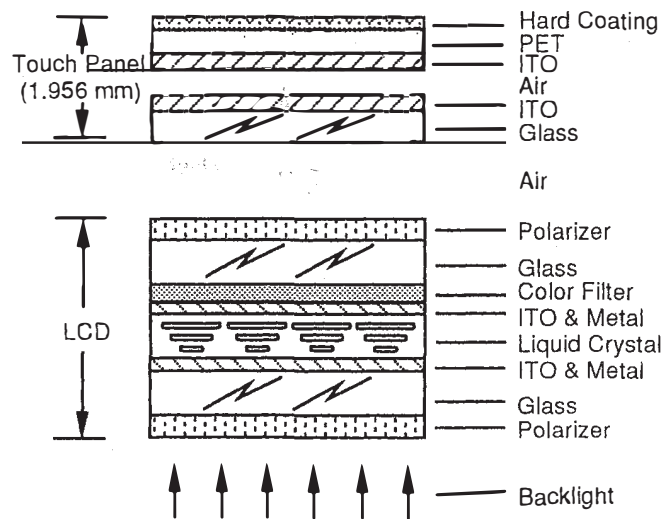


Figure 7.4 Simplified cross-sectional view of the layers in a touch-screen panel and LCD.

layer (in Figure 7.4 this layer consists of air). The top substrate, in our case PET, should be flexible enough to allow actuation of the switch. The bottom substrate, in our case glass, can be either rigid or flexible. Note that for simplicity Figure 7.4 does not show the conductive bus bars.

Analog resistive touch panels operate by applying a voltage gradient across one conductive layer and measuring the voltage at the point of contact with the opposing conductive layer. Actuation with a finger or stylus on the touch panel brings the two opposing layers into electrical contact. Position information on the point of contact can be obtained by measuring the voltage levels. The conductive effects of these two resistive layers when activated are indicated by resistors in Figure 7.3. Even though the resistors on each axis appear on the same figure, they do not function simultaneously. For each decoding cycle, there are two scans, one along each axis. During each scan, only one layer is conductive and has a voltage gradient across it. The other layer acts as a high-impedance probe (sensor). Together these two layers act like a potentiometer with a sliding wiper. One layer is denoted as X and the other is denoted as Y. The resistance primarily originates in the ITO sputtered surface and the silver-ink bus bars connected to the decoding circuitry. The ratio of the coordinates is proportional to the ratio of resistances in each direction:

$$\frac{R_{X1}}{R_{X2}} = \frac{X1}{X2}$$

$$\frac{R_{Y1}}{R_{Y2}} = \frac{Y1}{Y2}$$

The output voltage for the X position may be calculated using

$$V_X \propto \frac{R_{X2}}{R_{X1} + R_{X2}}$$

where V is the voltage and R is the resistance. Similarly, the output voltage for the Y position may be calculated from

$$V_Y \propto \frac{R_{Y2}}{R_{Y1} + R_{Y2}}$$

Since each touch panel may have slightly different characteristics, such as a different ratio of resistance, calibration is often necessary for accurate determination of the point touched. The touch panel can be calibrated first measuring the output voltages for known locations. Comparing these measurements to the expected values enables conversion factors to be determined. Because the actual output response and the

expected response are proportional to one another, the conversion factors simply amount to a scale factor and an offset for each axis. Measurements of two points on each axis are required to determine the two conversion factors for each axis. Note that the calibration eliminates the major nonlinear effect due to the bus-bar resistance and can handle minor nonlinear perturbations such as that due to unevenly sputtered ITO. One calibration equation of this type is shown here:

$$P_{\text{out}} = \frac{P_A - P_B}{V_A - V_B} V_{\text{out}} - \frac{P_A - P_B}{V_A - V_B} V_{\text{offset}}$$

where P is the position in either X or Y , V_{offset} is the voltage due to the driver and the bus-bar resistance. The terms A and B represent the two measured points, respectively. The calibration is also a function of the alignment between the touch panel and the display. If this alignment is not a constant, a dynamic calibration may be required. Moreover, when touch screens are being mounted on top of display screens, the distance between the touch panel and the surface of the display should be minimized in order to reduce parallax error.

At any boundary where there is a medium layer change, a certain amount of light is reflected. This causes glare on the screen. If the light is normal to the surface of the touch screen, the reflection can be calculated by

$$R = \left(\frac{n_1 - n_2}{n_1 + n_2} \right)^2$$

where n_1 is the index of refraction of one layer and n_2 is the index of refraction of the adjacent layer. From the preceding equation, we see that the most serious reflection occurs between layers having a large difference in the index of refraction and that more layers give rise to additional reflections. For the example shown in Figure 7.4, the indexes of refraction for PET, ITO, air, ITO, and glass are 1.67, 2.0, 1.0, 2.0, and 1.51, respectively. Knowing the thickness and index of refraction for each layer, the amount of light reflection and the transmittance of different touch-panel configurations can be calculated.

As the previous equation indicates, the extra resistive film added on top of the flat display panel for the resistive touch panel reduces the brightness or contrast and increases the amount of glare. One alternative to avoid this problem is to use an infrared touch technique, which has negligible effects on contrast. This technology is based on an infrared sensor array with scanned horizontal and vertical beams. These beams cross the surface of the display panel when they are turned on. Whenever both horizontal and vertical beams are blocked by a finger or stylus, a touch point is identified. The drawbacks of this technique are that it requires more computational power, costs more, has lower resolution (like the matrix format in Figure 7.2) and is more sensitive to the environment.

7.2.3 Additional Design Considerations

In addition to selecting a proper display device, one important factor that should be considered is that drivers often have difficulty reading detailed maps while driving. When a vehicle is in motion, it is preferable to display a limited amount of information such as a turn arrow, the shape of the next intersection, and the distance to the next intersection. When the vehicle is not in motion, a detailed map overlaid with the suggested route should provide a better overall picture of the surrounding area. For safety considerations, use of this map should be discouraged while the vehicle is in motion. In addition to variations on each display format [4,23], it is interesting to examine the two display formats, turn arrow versus route map, from the human factors perspective. One study has found that the route map makes drivers glance more often at the display screen than the turn arrow [9]. When assisted with voice guidance, the total number of glances is much smaller in both formats. In short, a turn arrow display with voice guidance should be used whenever possible.

As a design note, an interlocking mechanism, a dimmer mechanism, and a rotational mechanism (for displays that do not have a full viewing angle) should be included as control features for any visual-display-based interface. When the vehicle is moving, the interlocking mechanism should be able to automatically freeze most of the control functions for safety considerations. This is particularly important for touch-screen controlled displays. Interaction with the map display while driving should be discouraged. A dimmer mechanism can allow the driver to control the screen brightness. Otherwise, it might be very difficult for a driver to see outside clearly when switching from a very bright screen to a dark road at night. As an additional incentive to incorporate a dimmer, a U.S. National Highway Traffic Safety Administration regulation (NHTSA S5.3.5) requires this mechanism for any illumination source forward of a specified vertical transverse plane in the driver's compartment unless the light is more than barely discernible. The rotational mechanism for a limited-viewing-angle display is beneficial not only for the driver but also for any passengers who may wish to see the display screen. Some unpredictable light sources may make the screen very difficult to see from a particular angle. The ability to rotate the display can reduce these potential problems. Otherwise, a larger viewing angle display must be used.

The most important design considerations for an in-vehicle display are safety and ease of use. Eye glance frequency and duration of eye fixations on the display device are directly related to these two important design considerations. One study has found that the first-time user spent 20.4% of driving time looking at a poorly designed and complex route guidance device [24]. This reduced the glance time spent monitoring the road from 91% to 72.4%. Thus, this factor (which does not only apply to visual devices) must be kept in mind when designing the in-vehicle user interface. Because of the lack of adequate baseline data, it is still premature to establish guidelines or to provide a widely accepted set of numbers for glance

frequency and duration. One study based on limited research has suggested the glance limits given in Table 7.3 [25] as a means of stimulating further discussions. Refer to this study and its references for further information on other human factor considerations, the methods used, and the data collected during field and laboratory research. More research is needed in this area.

Other design considerations for driver interfaces include when, where, and how the user should be informed of an upcoming turn (Section 6.2.2), what minimum text size should be used, how the graphical representation of the intersection should be designed, and so on. For instance, the most commonly used method of determining the character height is to specify that the visual angle subtended by a character (its height divided by the viewing distance) be greater than or equal to 0.007 radians. The turn-by-turn display should be simple and consistent while providing landmarks such as traffic lights and stop signs.

Before leaving the section devoted to visual-display-based interfaces, we want to briefly discuss a new software technology that has been developed recently [26] for displaying maps. It provides a simulated 3D view on the in-vehicle display monitor equivalent to that seen 1,000m from the rear of the car and at a height of 360m. The driver seems to "look down" at an angle of approximately 16 degrees, in the direction of travel (Figure 7.5).

In 2D display systems, it is often necessary for drivers to change the scale of their map displays manually, that is, to "zoom in" and "zoom out." For example, zooming out is commonly required in order to view the entire route in a large road network. And zooming in is often necessary when approaching an intersection. Because the 3D display can show a map with a foreground 500m wide and a background 7,000m wide [Figure 7.5(b)], it is easy to display the entire planned route on the screen without zooming features.

This 3D display was developed by converting a map database into a hypothetical image on which the vehicle position and planned route have been superimposed. It is designed to permit the section of the map closest to the current vehicle position to be shown in enlarged format. The scale of the display is continually updated as the trip progresses to show additional areas of the map. The 3D view is generated by turning the 2D map into a certain degree so the top edge of the original 2D map becomes a foreground (back width, Figure 7.5) and the bottom edge of the original

Table 7.3
Glance Limits for Visual Display Devices

<i>Acceptability</i>	<i>Number of Glances</i>	<i>Glance Duration (sec)</i>
Best expected	2.0	0.9
Desired/planned	2.5	1.1
Worst case	4.0	1.2

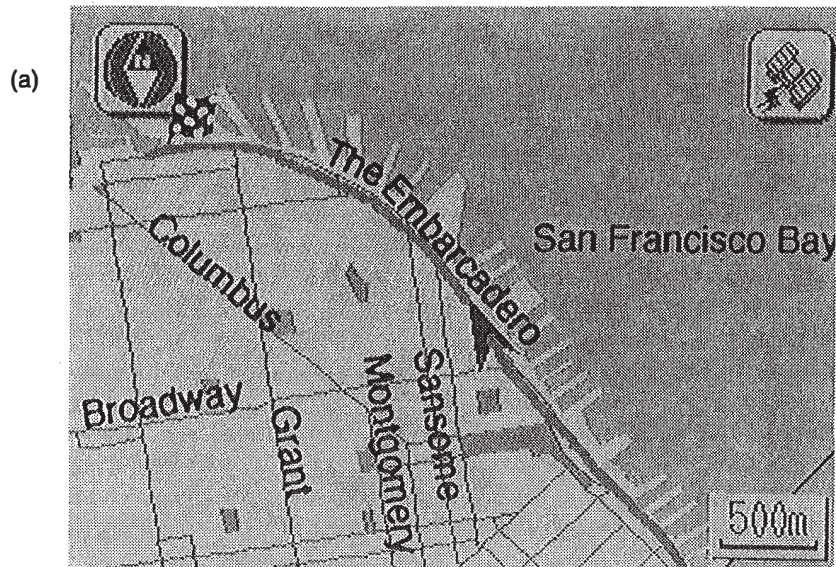


Figure 7.5 Comparison of (a) a 2D map (front and back widths: 500 m) and (b) a 3D map (front width: 500m; back width: 7,000m). (Courtesy of Nissan/Xanavi.)

2D map becomes a foreground (front width). Another technology that generates a true 3D display is discussed in [27]. Besides the 2D digital map database, it requires an additional database to derive altitude data.

7.3 VOICE-BASED INTERFACES

Speech is a natural interface that can be used to provide hands-free control of location and navigation systems. It is particularly important for the vehicle driver to operate the steering wheel rather than the navigation device during the trip. Because people typically speak three times faster than they can type, the access time for any computer device can be reduced, and access devices such as touch screens, push buttons, and keyboards can be eliminated using speech.

Table 7.4 lists two voice input and output methods, along with their evaluation scores and recommended usage [5]. The evaluation criteria used to grade these methods are identical to those used in Section 7.2. Upon comparing Tables 7.2 and 7.4, we see that voice input and output generally score higher than visual display technology for vehicle use.

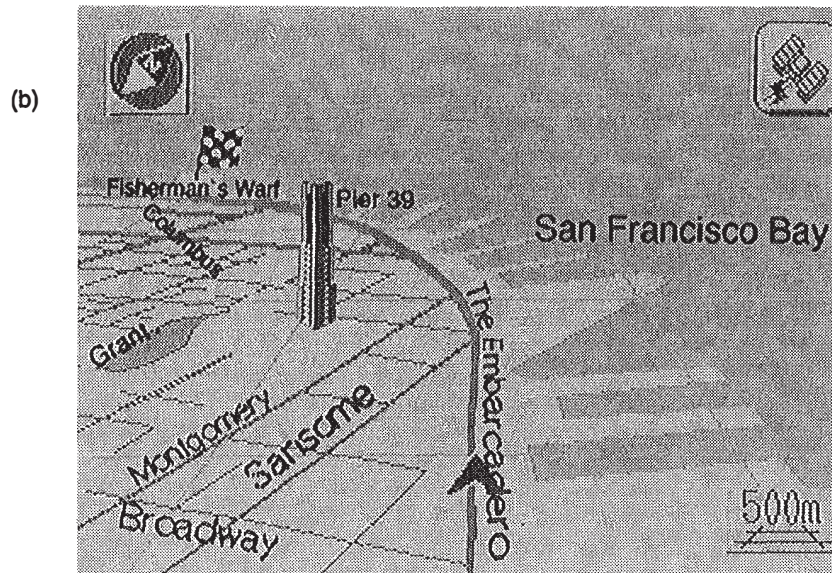


Figure 7.5 (continued)

Table 7.4
Voice Input and Output Methods

Voice	Score	Recommended Usage
Speech	8	Where the user's eyes are busy or mobility is required
Auditory	8	When information is short, simple, and transitory, requiring an immediate or time-based response, visual display is restricted, etc.

Voice processing methods can be classified as follows [28]:

- *Speech coding*: Obtaining compact digital representations of acoustic speech signals for efficient transmission or storage;
- *Speaker verification*: Verifying the claimed identity of a speaker for access to restricted areas or functions;
- *Speech recognition*: Recognizing an acoustic speech signal to identify the linguistic message intended;
- *Speech synthesis*: Generating an acoustic speech signal from ASCII text to convey a message from a machine to a human.

All four technologies can be used in vehicle location and navigation systems. A detailed review of these technologies can be found in the references listed in [28,29]. Speech coding can be used to convert human voice analog signals into coded digital signals [30]. When used for storage, speech encoding will maintain a desired level of voice quality while keeping the bit rate as low as possible. When used for transmission, speech coding will conserve bandwidth or bit rate while maintaining adequate voice quality. Human voice signals can be recorded and stored in memory and then transmitted to a loudspeaker on demand for voice output used to guide the driver. Speaker verification can be used to add a security feature to the vehicle such as verifying the identity of persons attempting to access the vehicle. It works by determining whether an unknown speech sample was spoken by the individual whose identity was claimed. The process of speaker verification is similar to that of speech recognition. Speech recognition can be used as an input control device. Speech synthesis can be used as an output device. These two latter technologies are discussed in more detail in the next section.

7.3.1 Speech Recognition

Perfectly reliable speech recognition is difficult for a number of reasons. First, speech differs from person to person. Second, background noise often affects the voice signal. Third, people commonly make errors, slightly mispronounce words, insert extraneous words, or run words together when talking fast.

As summarized in [12], the following functions can be achieved when speech recognition is used for device controls:

- Hands-free control while driving;
- The availability of speaker-dependent and speaker-independent devices.

However, until confidence in speech recognition outcomes can be reliably demonstrated, it is better not to use it for critical inputs.

Speech recognition systems are often classified into three different types [28]. Systems of the first type are known as "isolated-word systems." In these systems, the user is expected to speak a single word or phrase from a specified vocabulary. The speech patterns used by these systems involve isolated words or phrases. Systems of the second type are known as "connected-word systems." In these systems, the user is expected to speak a fluent sequence of words and phrases, and the system matches each sequence to a string of words. The speech patterns used by these systems involve distinct words or phrases. Systems of the third type are known as "continuous-speech systems." In these systems the user is expected to speak fluently, and the system recognizes the speech based on word lexicon and an associated word grammar. The speech patterns of these systems are subword units from which words and phrases are created using a lexicon of word pronunciations.

Commercial speech recognition systems can also be classified as speaker dependent or speaker independent. Speech recognition systems can further be classified on the basis of discrete words versus continuous words, read text versus spontaneous speech, vocabulary size, etc. The low-end speaker-dependent technology is usually based on a template or acoustical representation of speech. User-trained "voice prints," or templates are stored in the system and then compared against the commands spoken by the user. Template-based speaker-dependent recognition is used for relatively small- to medium-sized vocabularies, occasionally up to a few hundred words. Another technique that uses speaker-dependent recognition systems involves matching subword units, such as phonemes, syllables, etc. This technique is often used for medium or large vocabulary sizes (in the thousands of words). Speaker-independent systems are supposed to work for any speaker, without training. The ideal goal is to develop a high-end continuous speaker-independent system with a large vocabulary, no required training, and high accuracy. In these systems, the subword units used in such systems are trained using a large number of speech samples that fairly represent the gender, accent, and dialect characteristics of the speaker. This large number of speech samples should provide a fair representation of the acoustic, phonetic, and linguistic variance within a recognition task or across several tasks. Despite recent advances, the best high-end system today is still impractical for embedded vehicle applications.

It is very common for vendors' products to use a statistical modeling technique based on hidden Markov models (HMMs) to match the features (parameters) extracted from human voice. HMMs are sophisticated statistical models. In the second-generation technology, HMMs are used to model subword units (phonemes), as opposed to words. HMMs can improve speech recognition performance in several ways. An HMM-based small-vocabulary system can tolerate wide variations in the analog speech signal and so can perform speaker-independent recognition, work reliably in high-noise environments, and recognize continuous speech. Alternatively, the added power of HMM technology can be used to distinguish a larger number of subtle differences between words, so that the recognizer can support a larger vocabulary.

Everything from small-vocabulary command and control using isolated words and small-vocabulary continuous speech through large-vocabulary dictation can be based on the same core HMM technology. In older products, spectral templates are used in place of HMM. Historically, the template-based method has not fared as well [29].

Many different algorithms and implementations exist for processing voice signals. The basic principle is to match the incoming transformed speech input against a stored representation. (This is often called the decoding process [29].) Acoustic models are generally used to capture the phonetic or word-level properties of speech while statistically based language models are often used to capture the

syntactic and semantic regularities of language. The approximate diagram of Figure 7.6 provides a simple visual explanation of how a computer recognizes speech.

Current research has underlined the importance of robust and accurate front ends, detailed acoustic modeling, and statistical language modeling. Natural language processing techniques have also been studied for speech interface and interpretation of spontaneous speech. Combinations of neural networks for signal classification and HMMs for speech modeling have been demonstrated successfully.

Recent products for recognition of continuous speech typically have maximum vocabularies of around 2,000 words, with a *perplexity* of 30. Perplexity is a measure of sentence complexity corresponding to the average number of words in the vocabulary that could follow any given word under the recognizer's syntax rules. A typical *Wall Street Journal* article has perplexities of around 200 to 300. Thus, there is very clearly a large gap between user expectations and reality for current speech recognition products.

Some researchers believe that the most important challenge is the integration of speech recognition and natural language processing technology to achieve speech understanding. They have attempted to build a probabilistic parser that operates top-down, using a stack-based control strategy that favors the most likely analyses. It

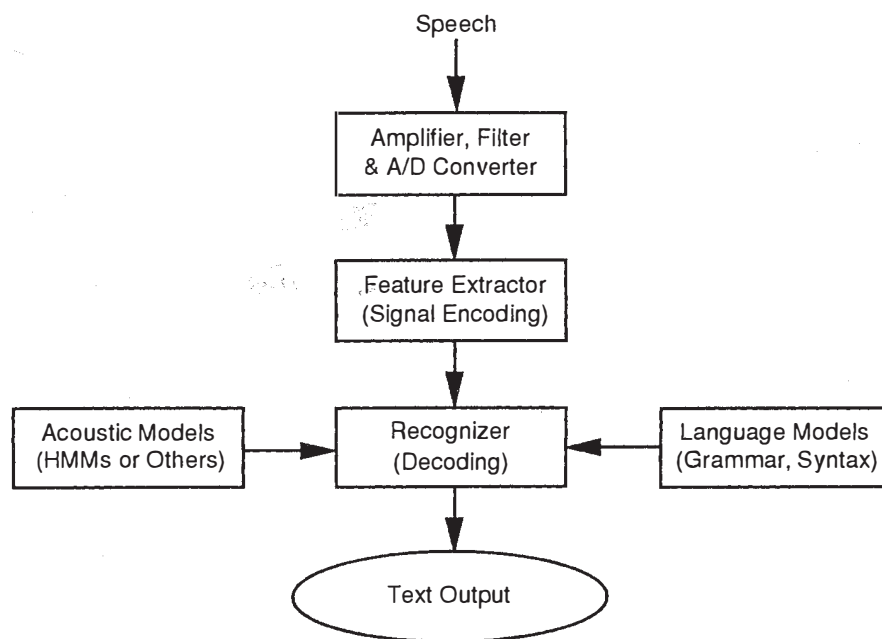


Figure 7.6 Simple block diagram for speech recognition.

is designed so that its grammar rules and associated probabilities can be automatically trained from a set of parallel sentences. The framework is organized such that the prediction probabilities for next-word candidates are explicit. This approach is different from the way probabilities are handled in other probabilistic natural language frameworks, where probabilities are typically associated with rule predictions.

For practical systems, large vocabulary systems must be made robust against interfering noise. Two types of methods are being studied. First, the corrupted speech waveform may be preprocessed so that the resulting parameters are very similar to those of clean speech. Techniques in this category include spectral subtraction, spectral mapping, and inherently robust parameterization. In the first category, some schemes have also attempted to estimate the clean speech signal using information about the speech. Their representatives are HMMs and minimum mean square error estimators. The second group of methods attempts to adjust the pattern-matching stage to compensate for the interfering noise. Methods using this approach include noise masking, state-based filtering, cepstral mean compensation, HMM decomposition, and parallel model combination.

Many other techniques (such as fuzzy logic and artificial neural networks) are being investigated by researchers around the world. Only time will tell if these techniques can be successful in the real-world environment.

Even when the host CPU has enough horsepower to run the entire speech recognition algorithm, a digital signal processor (DSP) has distinct cost advantages, especially in continuous-speech systems. Low-cost DSPs replace the expensive analog hardware formerly required to perform high-quality speech filtering and digitization. High-end DSPs can also replace the conventional CPUs.

The trend of DSP development is that the performance of DSP chips doubles about every 3 years, while the costs decrease at approximately the same rate. All commercial speech recognition algorithms can run on general-purpose platforms, but sometimes it pays to optimize the hardware for the task at hand, particularly in cost-sensitive, high-volume consumer applications. Therefore, there has been a trend to port speech recognition technology to specialized DSPs for products such as personal digital assistant (PDA) and other hand-held or mobile platforms.

Because there are so many speech recognition products available, one might consider integrating these products into various vehicle location and navigation systems. Generally, most of the hardware cost is in the memory and in the audio front end. The cost of the software varies over a very wide range. High-volume applications can bring per-unit licensing costs down to a very competitive price (less than \$100, perhaps even less than \$10). The best way to keep the cost down is to minimize the need for modifications to the vendors' original designs.

A few general guidelines for design of spoken input are available (see [4] and the references contained therein). The input should be easy to learn and remember, and should be controlled by brief commands. It should present explicit feedback describing whether the system is ready to accept speech or not. Acknowledgment

should be given for each user command in understandable phrases; the feedback should be context sensitive, and distinguish between data and commands. Driver speech inputs should be acoustically distinguishable. Command names should be chosen so as to be semantically discriminable. The speech recognition devices should be able to accept a variety of pronunciations.

Development of advanced speech recognition technology requires expertise in signal processing, system theory, statistics, pattern recognition, and computer science, built on a solid understanding of speech science and linguistics. This is especially challenging for vehicle-based applications. As a rule of thumb, a speech recognition product is ready to be used in an application once the word error rate has been reduced below about 5% in its intended environment. Unfortunately, this is still not reliable enough for any critical control input to a vehicle during an emergency situation.

7.3.2 Speech Synthesis

Speech synthesis is a computer-aided technology output understandable human language. As summarized in [12], the following functions can be achieved using speech output:

- Wide range of possible output information;
- Low visual demand;
- Low training or skill demand;
- Best for short, simple messages.

However, the speech output is poor for visual or spatial information. Complex information cannot be extracted as easily as in a visual display. If speech synthesis is overused, the driver could be distracted.

Speech synthesis systems can convert text messages to speech based on a linguistic analysis of the text and an acoustic analysis of the speech sounds within the context of the desired sentences (Figure 7.7). These sounds can be phonemes, syllables, words, etc. Intonations, word pronunciations, exceptions, and rules for generating contextual changes in the sounds are used in the linguistic and acoustic analysis. Finally, the speech output can be obtained by generating appropriate word durations as well as global and local pitch contours for the prominence of words and ideas within the text. Hopefully, the output will be intelligible and reasonably natural.

Two synthesis techniques (as shown in Figure 7.7) are commonly used [29]. Formant synthesis uses a mathematical model of the waveform generated by the human speech production apparatus. Concatenation synthesis splices prerecorded segments of speech. The first technique uses sound symbols to define a sequence of

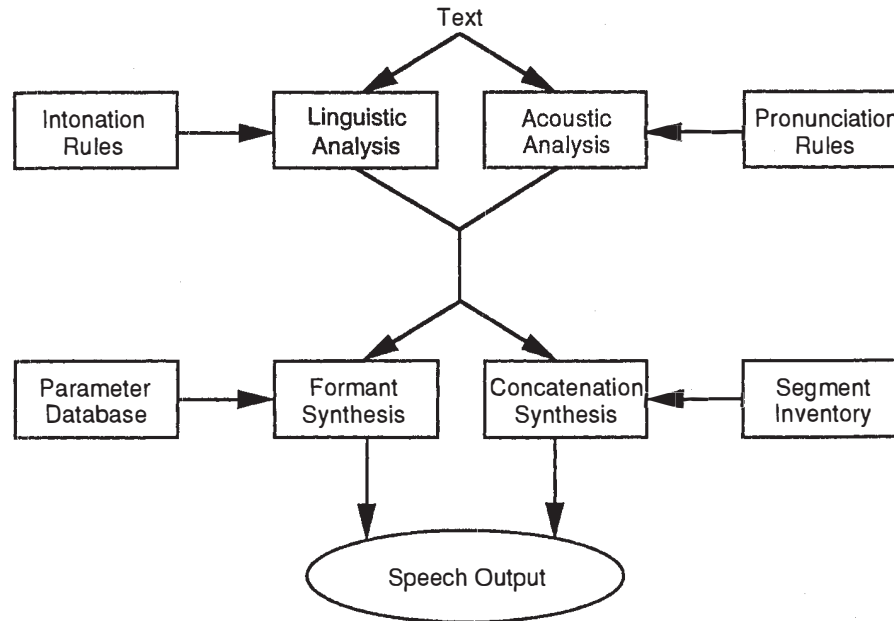


Figure 7.7 Simple block diagram for speech synthesis.

acoustic targets. It interpolates the acoustic signal between these targets to mimic the dynamics of the human voice. The second technique obtains the same effect by using a sequence of prerecorded elements corresponding to the sound symbols in context. It smooths the junctures between these elements. Because concatenation synthesis is based on real human voices, it can produce a more natural voice, although more storage space is required (unless CPU-intensive speech coding is performed).

Progress in synthesizers appears to have reached an asymptote because successive small improvements are now difficult to achieve [29]. Although speech synthesis is currently a stable technology with error rates as low as 3.25% for individual word perception on a standardized test, the available synthesizers still do not sound natural. Because formant synthesis seems to have reached a plateau, any dramatic improvements in the future are expected to depend on concatenative synthesizers.

A few general guidelines concerning auditory output are available (see [4] and references contained therein). Nonspeech auditory sounds should be used only for providing alerts or warnings. Other messages should be spoken. Synthesized speech is only recommended for situations that require substantial flexibility in generating the spoken messages. When the messages are relatively limited and known ahead of time, a recorded human voice is preferred. Instructions should always be limited to four chunks (three or less in most cases). For a maneuver instruction, it is

better to provide "early," "prepare," and "approach" messages in sequence (see Section 6.2). Each decision should be accompanied by a separate instruction when time permits. Sometimes, an auditory output may be required to remind the driver that the system is still operational. When the street name changes or a new city is entered, proper instructions should be announced. A function to activate and deactivate auditory output and to adjust the output volume should be provided.

Both synthesized and prerecorded (digitized or coded) voice output have been shown to be very useful in vehicle guidance. As mentioned earlier, studies of voice guidance systems have found reductions in the duration and frequency of glancing at the display relative to nonvoice guidance systems [9,31]. This can certainly help improve safety.

As we have learned, speech synthesis has become a stable technology. There are many examples describing the use of this technology [32,33]. Speech recognition technology has even recently become available in vehicle navigation products. One example of this is a speaker-independent voice-activated navigation system called AudioNav [34]. It recognizes isolated letters, numbers, and three dozen words and has a synthesized voice output. This system uses HMM technology and claims to be sufficiently robust for use in this important, but difficult, application environment—the automobile. Road noise and the impracticality of using a headset microphone while driving make automotive speech recognition a particular challenge. This fact is reflected in AudioNav. One main disadvantage of AudioNav is the inconvenience of voice input. It requires the user to spell out the destination and requires the user to know his or her present vehicle location.

Suppose that future speech processing technologies are natural and robust enough for vehicle applications. Will drivers prefer voice devices over visual displays? If both visual and voice devices are integrated in a vehicle location and navigation system, what is the optimum combination of visual and voice interfaces to maximize the safety and performance for both the system and the human? Will a route guidance system ever be better than a human travel guide who is familiar with the area [35]?

References

- [1] M. S. Sanders and E. J. McCormick, *Human Factors in Engineering and Design*, 7th ed., New York: McGraw-Hill, 1993.
- [2] D. A. Norman, *Psychology of Everyday Things*, New York: Doubleday, 1988.
- [3] B. Peacock and W. Karwowski, *Automotive Ergonomics*, Washington, DC: Taylor & Francis, 1993.
- [4] P. Green, W. Levison, G. Paelke, and C. Serafin, *Preliminary Human Factors Design Guidelines for Driver Information Systems*, Ann Arbor, MI: The University of Michigan, Transportation Research Institute, Technical Report No. UMTRI-93-21. Also published in McLean, VA: U. S. Dept. of Transportation, Federal Highway Administration, Report No. FHWA-RD-94-087, Dec. 1995.
- [5] L. Levitan, M. Burrus, W. L. Dewing, W. F. Reinhart, P. Vora, and R. E. Llaneras, *Preliminary Human Factors Design Guidelines for Automated Highway System Designers*, McLean, VA: U.S.

- Dept. of Transportation, Federal Highway Administration, Report No. FHWA-RD-94-116, Dec. 1995.
- [6] Department of Defense, *Handbook for Human Engineering Design Guidelines*, MIL-HDBK-759C, Philadelphia, PA: Department of Defense Single Stock Point—Customer Service, July 1995.
 - [7] K. H. E. Kroemer, H. B. Kroemer, and K. E. Kroemer-Elbert, *Ergonomics: How to Design for Ease and Efficiency*, Englewood Cliffs, NJ: Prentice Hall, 1994.
 - [8] D. J. Mayhew, *Principles and Guidelines in Software User Interface Design*, Englewood Cliffs, NJ: Prentice Hall, 1992.
 - [9] M. C. Hulse, T. A. Dingus, D. V. McGehee, and R. N. Fleischman, "The Effects of Area Familiarity and Navigation Method on ATIS Use," *Proc. Human Factors and Ergonomics Society 39th Annual Meeting*, Oct. 1995, pp. 1068–1071.
 - [10] W. B. Verwy and N. A. Kaptein, "Automatic Man-Machine Interface Adaptation to Driver Workload," *Proc. First World Congress on Applications of Transport Telematics and Intelligent Vehicle-Highway Systems*, Nov./Dec. 1994, pp. 1701–1708.
 - [11] J. Walker, E. Alicandri, C. Sedney, and K. Roberts, "In-Vehicle Navigation Devices: Effects on the Safety of Driver Performance," *Proc. IEEE Vehicle Navigation and Information Systems Conference (VNIS '91)*, Oct. 1991, pp. 499–525.
 - [12] B. Leiser and K. Carr (Eds.), *Analysis of Input and Output Devices for In-Car Use*, Traffic Research Center, University of Groningen, The Netherlands, Nov. 1991.
 - [13] A. Nojima, H. Morita, H. Kondo, and N. Komoda, "Development of Toyota ATIS In-Vehicle Equipment in VICS Demonstration Test," *Proc. IVHS America 1994 Annual Meeting*, Apr. 1994, pp. 664–670.
 - [14] A. Marcus, "Motorola Smart Car," in *Macromedia Director Design Guide*, C. Clarke and L. Swearingen, Eds., Indianapolis, IN: Hayden Books, 1994, pp. 66–76.
 - [15] M. F. Coleman, B. A. Loring, and M. E. Wiklund, "User Performance on Typing Tasks Involving Reduced-Size, Touch Screen Keyboards," *Proc. IEEE Vehicle Navigation and Information Systems Conference (VNIS '91)*, Oct. 1991, pp. 543–549.
 - [16] S. L. Fowler and V. R. Stanwick, *The GUI Style Guide*, Boston: AP Professional, 1995.
 - [17] M. Schadt, H. Seiberle, and A. Schuster, "Optical Patterning of Multi-Domain Liquid-Crystal Displays with Wide Viewing Angles," *Nature*, Vol. 381, May 16, 1996, pp. 212–215.
 - [18] "Sharp Realizes New Applications with Technology Breakthroughs," *Display Devices*, No. 11, Spring 1995, pp. 12–13.
 - [19] C. Ajluni, "FED Technology Takes Display Industry by Storm," *Electron. Des.*, Oct. 25, 1994, pp. 56–66.
 - [20] S. Matsumoto (Ed.), *Electronic Display Devices*, translated by F. R. D. Apps, New York: John Wiley & Sons, 1990.
 - [21] R. A. Perez, *Electronic Display Devices*, Blue Ridge Summit, PA: TAB Books, 1988.
 - [22] L. E. Tannas, Jr. (Ed.), *Flat-Panel Displays and CRTs*, New York: Van Nostrand Reinhold, 1985.
 - [23] J. M. C. Schraagen, "Information Presentation in In-Car Navigation Systems," in *Driving Future Vehicles*, A. M. Parkes and S. Franzen (Eds.), Washington, DC: Taylor & Francis, 1993, pp. 171–185.
 - [24] T. Ross and G. Burnett, "The Right Road to Take," *ITS International*, June 1996, pp. 54–55.
 - [25] P. Green, *Suggested Procedures and Acceptance Limits for Assessing the Safety and Ease of Use of Driver Information Systems*, Ann Arbor, MI: The University of Michigan, Transportation Research Institute, Technical Report No. UMTRI-93-13. Also published in McLean, VA: U. S. Dept. of Transportation, Federal Highway Administration, Report No. FHWA-RD-94-089, Dec. 1995.
 - [26] M. Watanabe, O. Nakayama, and N. Kishi, "Development and Evaluation of a Car Navigation System Providing a Bird's-Eye View Map Display," *SAE Paper No. 961007*, Society of Automotive Engineers, 1996.

- [27] M. Oikawa and M. Kato, "A Three Dimensional Navigation System '3D Navi'," *Abs. Third World Congress on Intelligent Transport Systems*, Paper #3129 (full paper in CD-ROM), Oct. 1996, p. 205.
- [28] L. R. Rabiner, "The Impact of Voice Processing on Modern Telecommunications," *Speech Comm.*, Vol. 17, Nos. 3-4, Nov. 1995, pp. 217-226.
- [29] A. I. Rudnicky, A. G. Hauptmann, and K.-F. Lee, "Survey of Current Speech Technology," *Comm. ACM*, Vol. 37, No. 3, Mar. 1994, pp. 52-57.
- [30] A. S. Spanias, "Speech Coding: A Tutorial Review," *Proc. IEEE*, Vol. 82, No. 10, Oct. 1994, pp. 1541-1582.
- [31] H. Kishi and S. Sugiura, "Human Factors Considerations for Voice Route Guidance," *SAE Paper No. 930553*, Society of Automotive Engineers, 1993.
- [32] J. R. Davis and C. M. Schmandt, "The Back Seat Driver: Real Time Spoken Driving Instructions," *Proc. IEEE Vehicle Navigation and Information Systems Conference (VNIS '89)*, Sep. 1989, pp. 146-150.
- [33] M. K. Krage, "The TravTeck Driver Information System," *Proc. IEEE Vehicle Navigation and Information Systems Conference (VNIS '91)*, Oct. 1991, pp. 739-748.
- [34] J. Smolders, R. Diller, and D. Van Compernelle, "Noise Robust Speech Recognition Makes In-Car Navigation Safe and Affordable," *Proc. Second World Congress on Intelligent Transport Systems*, Nov. 1995, pp. 601-604.
- [35] W. Fastenmeier, R. Haller, and G. Lerner, "Preliminary Safety Evaluation of Route Guidance Comparing Different MMI Concepts," *Proc. First World Congress on Applications of Transport Telematics and Intelligent Vehicle-Highway Systems*, Nov./Dec. 1994.

CHAPTER 8



WIRELESS COMMUNICATIONS MODULE

8.1 INTRODUCTION

Reliable communication is a vital component for further improvement of performance and increased functionality in vehicle location and navigation systems. It provides a very valuable opportunity to present relevant information (and invaluable safety features) to the vehicle and its occupants as well as to obtain data for transportation management systems. Many quality services can be provided to drivers using communications technology. For example, the drivers and in-vehicle navigation systems can receive updated traffic information to help them maneuver through traffic, the traffic control centers can obtain current traffic reports from vehicles for traffic management and travel time predictions, and traffic control centers could even locate and help the vehicle navigate in the road network.

Communications technologies are poised for major expansion in two key areas: wireless data communications and high-speed wireline (wire-based) communications supported by the Internet and ancillary networks. Ultimately it will be the marriage of these two key technologies that will bring about a revolution resulting in a new information society.

Wireless data applications play a critical role in making the vision of mobile computing a reality. Today's competitive and fast-paced business climate demands

tools that allow people to communicate at their own convenience and discretion. Wireless applications continue to be refined and will become commonplace in the near future. Data systems could develop into one of the fastest growing sectors of the wireless industry.

High-speed wireline infrastructures such as the Internet are a key element to providing a backbone for data delivery. This infrastructure has approximately doubled in size each year since 1988. A key factor to this dramatic growth has been the Internet's widespread connectivity and universal access, which often makes it quicker and easier for people to route their traffic through a local access point than to talk directly to each other. A national backbone with countless local access points gives the mobile community easy access to vast amounts of information and services not available on most private networks. This access has propelled the capabilities of wireless computing into the global arena.

The emphasis in this chapter falls more on wireless communications, and most of the technologies to be discussed are based on radio-frequency (RF) communications. Wireline communications are discussed spontaneously in connection with the wireless technologies when appropriate. To concentrate on our main topic of describing the existing communications technologies, we will not discuss how to use the radio signal for positioning here. However, we should keep in mind that many of the technologies discussed can be used for positioning of vehicles or mobile devices, which is addressed in Section 9.2. A large number of diverse technologies are available and space is limited, so it would be difficult to address all of these technologies in a single chapter. We therefore discuss only the major technologies currently used for vehicle location and navigation applications and direct readers to the references listed at the end of this chapter for further reading on other technologies. Section 8.2 discusses the basic attributes of communication subsystems. Section 8.3 provides an overview of each technology to quickly bring the reader up to date, but also contains important details, many useful summary tables, and comprehensive references to encourage further research.

8.2 COMMUNICATIONS SUBSYSTEM ATTRIBUTES

Vehicle communications require a seamless, wireless infrastructure for voice and data that can reliably and efficiently deliver real-time traffic and other information between a fixed infrastructure (base stations or repeaters) and mobile vehicles (mobile radios) or mobile devices (portable radios) or between the mobile vehicles themselves. Depending on whether the link is simplex (information is transmitted in one direction or both directions by alternating between transmitting and receiving on the same frequency), half-duplex (information is transmitted in both directions by alternating between transmitting and receiving but on different frequencies), duplex (information is transmitted in both directions simultaneously using different frequencies), a

matching transmitter and/or receiver must be used both in the mobile vehicles and in the infrastructure. In this book, when referring to two-way communication, we will mean either duplex or half-duplex depending on the context. Mobile communications can be both one-to-many and one-to-one. Broadcasting is a one-way, one-to-many communication, in which a single message can be received by all mobile receivers within range. For one-to-one communication, a communications channels (frequency) is reserved solely for use by two parties for the duration of the communication. A dispatch system provides two-way communications channels for a designated group (or fleet) of users. As is apparent, different applications have diverse communication requirements. Furthermore, certain key attributes are required for successful deployment. All of these requirements and attributes apply equally to the intelligent transportation system (ITS), which is an emerging mobile computing technology. For ideal widespread ITS deployment, these attributes should include excellent coverage, high capacity, low cost, full connectivity, excellent security, and public access. We now briefly discuss the key attributes that can be used to assist in the selection of proper technologies for vehicle applications.

8.2.1 Coverage

Coverage refers to the geographic area where wireless voice and data transmissions can be sent and received reliably. Specific site or station coverage may depend on a number of factors at both the fixed and mobile ends. Transmitter power, antenna characteristics, channel characteristics, receiver quality, interference, communications protocols, and geographic topography are generally the factors that affect site coverage the most.

Boosting transmitter power is one method that can be used to increase coverage, but the benefits are limited since most base stations/repeaters typically have a transmitter power of 75 to 100W. On the mobile or portable end, it is desirable to obtain the required coverage with a minimum amount of transmitter power. This is because the transmitter power of portables is usually limited to 3 to 5W. To overcome the limited transmitter power of portable radios, tower-top receiver amplifiers are often added to the base stations/repeaters for better signal reception. In any case, doubling the input power results in only a 40% increase in field strength. Proper selection of antenna characteristics can be more effective in increasing coverage. At the base station end, increasing the antenna height is usually a very good solution. However, antenna towers are expensive and their heights are often restricted by local regulations.

Gain and height are two of the most important characteristics of an antenna. Antenna gain is a measure of how well the antenna concentrates its radiated power in a given direction. It provides a measure of the antenna's efficiency. The gain of an antenna is measured as the ratio between the highest radiation intensity (in any

direction) from the antenna and the radiation intensity from a reference antenna (such as a dipole antenna) or a theoretical omnidirectional antenna driven by the same total power, that is, an isotropic source that radiates energy in all directions. Antenna gain is measured in decibels (dBs) relative to an isotropic antenna, that is, dBi, and may vary widely depending on the application and design of the antenna. Gain numbers can range from 0 to 50 dB (for example, 0 to 17 dB for antennas that are omnidirectional in the horizontal plane).

The second main characteristic of an antenna is its height. This explains why most transmitters are located at the tops of tall buildings or hills to get maximum coverage. If no buildings are available, a tower is sometimes constructed along with a protected enclosure to house the base station equipment. Optimum height is determined by the required coverage, range, antenna characteristics, local regulations, the trade-off between extra height and greater signal attenuation (loss), and cost. To keep signal losses in the transmission lines to a minimum, the base station equipment is generally kept as close as possible to the antenna.

Table 8.1 summarizes the properties of antennas in each frequency band, assuming that the noise density is higher in urban areas and interference is present from another antenna in the same area. This table gives us a broad picture of the relationship between frequency and coverage. Note that even though a high-gain antenna design can be more easily constructed at higher frequencies, too much directivity may not be acceptable for the application, and path loss is inversely proportional to the wavelength squared.

As Table 8.1 implies, channel characteristics such as path loss, fading, and penetration loss can seriously impact coverage. Path loss is the attenuation of the signal as it propagates from the transmitting antenna to the receiving antenna. In practice, the path loss is more severe than the inverse square law would predict from free-space calculations. The path loss for mobile systems generally falls off as distance is cubed (or even faster). However, it is important to note that these figures must often be empirically derived.

Table 8.1
Frequency Band Characteristics

<i>Characteristics</i>	<i>25-50 MHz</i>	<i>148-174 MHz</i>	<i>450-512 MHz</i>	<i>806-960 MHz</i>
Antenna size	Long	Short	Shorter	Shortest
Antenna gain	Low gain	Low-high gain	Low-high gain	Low-high gain
Urban range	Poor	Good	Excellent	Excellent
Suburban range	Good	Excellent	Good	Fair
Rural range	Excellent	Good	Fair	Poor
Urban building "fill-in"	Poor	Fair	Good	Excellent
Foliage path loss	Low	Low	Medium	High
Interference	Severe	Medium	Minimal	Almost none

As the signal travels along its transmission path, it passes through air and other environmental material. Oxygen and water vapor in the air may absorb energy, and the resulting attenuation depends mainly on the frequency, which means that the attenuation due to rain increases with frequency and amount of precipitation. Frequencies below 1 GHz are essentially unaffected by atmospheric moisture, while frequencies above 10 GHz are severely affected (or even become unusable) for long transmission paths. Finally as Table 8.1 indicates, the 806- to 960-MHz band has high foliage attenuation, but good urban fill-in coverage (between buildings). The 148- to 174-MHz band has low foliage attenuation, but poor urban fill-in coverage.

Fading is a phenomenon that leads to variations in signal level. A distinction is made between short-term and long-term fading. For the short-term fading, signals vary over short distances due to multipath propagation. This type of fading generally becomes more severe as the frequency increases. Long-term fading is a random process caused by shadowing. For radio frequencies of 450 to 960 MHz or higher, shadowing becomes the most important environmental attenuation effect where buildings or other ground terrain may create radio shadows that can result in received signal strength differences as great as 20 dB. These fading effects are most severe in dense metropolitan areas. On the other hand, high frequencies bounce off buildings, providing good fill-in coverage (as mentioned above), although the ability of the signals to penetrate through buildings is a function of frequency band and signal strength.

Receiver sensitivity is an important measure of receiver quality, and has been defined in various ways depending on the type of signal modulation. Modulation is a technique for transmitting information by varying the amplitude, frequency, or phase of a carrier wave or signal. Sensitivity is a measure of the smallest signal that can be successfully received. A receiver with high sensitivity is desirable, since a lower field strength is required for the receiver to respond correctly at the fringe of the coverage area. For the analog signals used by FM (frequency modulation) receivers, the oldest standard for sensitivity measurement standard uses the 20-dB quieting method. In this method, the sensitivity is measured by determining the unmodulated input signal level that results in the noise output of the receiver being reduced 20 dB below the noise output level with no input signal. Another widely accepted measure of the sensitivity uses the 12-dB SINAD method [SINAD refers to the ratio of (signal + noise + distortion) to (noise + distortion) expressed in decibels]. Digital communication systems specify the sensitivity in terms of the signal level required to attain a bit error rate (BER) not exceeding 1%, or in many situations, the maximum BER allowed for acceptable system operation. Signal strength generally is given in microvolts; however, in some situations it will be specified in decibels relative to one milliwatt (dBm).

Features such as error detection and correction, data interleaving and coding, and the retry control employed by the communications protocol may directly affect coverage. A trade-off is usually necessary between throughput and coverage in this

case. Error detection and correction work by adding information to the message (sometimes referred to as *redundancy* when multiple copies of certain data are transmitted). This method can be used to detect, and perhaps correct, errors in transmission. The more redundancy added to the message, the more reliably the receiver can detect and correct errors, but with a corresponding reduction in the effective message rate. In practice, the number of additional check digits required to achieve error correction is much larger than that needed for just error detection. Convolutional encoding may be used in the forward error correction process to produce an output bit stream representing the original message such that the number of bits between adjacent bits from the original message is larger than the number of bits affected by a single burst error. For example, if an error burst of up to seven bits in length does occur, it will affect only a single bit in each codeword rather than a string of bits in one or two codewords. If an error is detected but cannot be corrected, then an automatic repeat request (ARQ) can be transmitted back to the sending unit to retry the transmission. Another alternative is the selective ARQ (SARQ), which requests retransmission of only the corrupted piece of information rather than the entire digital transmission. Thus, error detection and correction can result in a lower end-to-end BER for the message as a whole.

Geometry and topography must be taken into account when determining site coverage because hills, buildings, and other geographic features may absorb, refract, or reflect signals. Many coverage prediction models contain detailed geographic data, including the locations of buildings and terrain features to be used in coverage calculations.

The overall network coverage depends on the concentration and distribution of network equipment such as base stations. For example, cellular services typically position their base stations to emphasize wide geographical coverage. At the other extreme, infrared and microwave equipment generally requires line of sight between transmitter and receiver. Some packet data network operators concentrate their base stations in urban areas to emphasize in-building coverage. Such mobile technologies typically only serve metropolitan areas where system operators have the best opportunity for recovering their investment. On the other hand, the requirement for wide-area coverage is important if seamless nationwide operation of ITS is to be provided.

8.2.2 Capacity

Capacity varies in terms of message size and effective throughput. For example, paging services generally have limited message size. The effective throughput depends on transmission speed and number of concurrent transmissions/connections. Although capacity is directly related to transmission speed, the transmission speed figures cited by wireless providers can be misleading. Protocol overhead associated with error correction and reliable message delivery can account for throughput rates

that are as much as 50% lower than the maximum over-the-air transmission rate. Also, some mobile technologies employ store-and-forward delivery to compensate for their limited radio bandwidth or service capacity. With such networks, messages can be delayed anywhere from a few seconds to hours but may have substantially lower operating costs. In general, communications technologies with higher capacity and throughput will have higher operating costs.

8.2.3 Cost

Equipment and operating costs vary widely according to device capabilities, transmission rates, device size, and power requirements. Paging technology is generally the least expensive (and least capable) technology to implement, whereas satellite and microwave technologies are the most expensive to deploy. To provide universal access to ITS as a national network, subscriber costs must be kept to a minimum.

8.2.4 Connectivity

Connectivity may be supported by simplex, duplex, and/or half-duplex communications to provide broadcast, dispatch (group), and individual communications within the network. Simplex connectivity can be used for simple data collection purposes where sensor data is delivered to a processing center. Two-way communications are usually required for interactive sessions. A mobile or portable radio user may converse with the communications center, an individual radio user, a group of vehicles, or the whole network to communicate and exchange information. The situation is similar for the operator in a communications center.

8.3 EXISTING COMMUNICATIONS TECHNOLOGIES

Vehicle communications networks need to support many different applications within a wide set of user services including traffic management, emergency management, public transportation operations, commercial vehicle operations, electronic payment, and advanced vehicle control. Each application has specific needs that may best be satisfied by a particular communications technology. Because no single communications technology can satisfy all of these requirements, a hybrid implementation of technologies is likely to be required for a regional or statewide communications network. Table 8.2 provides a high-level view of existing wireless technologies (after [1]). This matrix is intended as an initial guide for use at the beginning of the selection process. The user should perform a more thorough evaluation of the planned application and available technology prior to actual implementation of the system. Many good references are available to aid the user with this analysis (see [1,2] and the references contained therein).

Table 8.2
Communications Technology Matrix

Technology	Reliability	Coverage	Avg. Data Transfer Speed	Equipment and Airtime Costs	Security	Simplex or Duplex	Real Time or Store and Forward	Access
Paging	E	Metro, some rural	2.4–3.6 Kbps	E	E	Simplex/duplex	Store and Forward/Real time	Public
Cellular	F-E	Metro, some rural	9.6–19.2 Kbps	F-E	P-E	Duplex	Real time	Public
PCS	E	Metro	≥8 Kbps	E	E	Duplex	Real time	Public
Private land mobile radio systems	VG-E	Metro, rural	1.2–64 Kbps	F-E	P-E	Simplex/duplex	Real time	Private/public
Radio data networks	VG-E	Metro	2.4–256 Kbps	F-E	E	Duplex	Real time	Public
Broadcast subcarriers	E	Metro	1.2–19 Kbps	E	E	Simplex	Real time	Private/public
Short-range beacons	E	1–100m	64–1024 Kbps	F	VG	Simplex/duplex	Real time	Private
Satellites	E	Worldwide	2.4–21.33 Kbps	P-E	E	Duplex	Real time	Private
Cordless telephony	E	None in the U.S.	≥32 Kbps	VG	E	Duplex	Real time	Public
Radio LAN networks	VG-E	Indoors/outdoors; 40–11,263m	64–5,700 Kbps	VG	E	CSMA/CD	Near real time	Private
Infrared LAN networks	E	Indoors/outdoors; 9–6,436m	1,000–10,000 Kbps	F-E	E	CSMA/CD	Near real time	Private
Meteor burst	E	Worldwide	2–32 Kbps	E	E	Duplex	Store and forward	Private
Microwave relays	VG-E	Metro, rural; 40,000-m hops	8448–250 × 1544 Kbps	VG-E	VG-E	Duplex	Real time	Private

Note: P = poor; F = fair; VG = very good; E = excellent; Metro = metropolitan areas; CSMA/CD = carrier sense multiple access with collision detection.

Both analog and digital systems are available for mobile radios (the digital systems are essentially related to voice digitization and to digital transmission). The major advantages of the digital systems over the analog systems are increased spectrum efficiency, more consistent audio quality throughout more of the coverage area, inherent privacy from analog scanners, and greater data rate capability.

We now take a brief look at some of the transmission and switching methods used for transmission of information. Three major channel access methods exist: FDMA (frequency-division multiple access), TDMA (time-division multiple access), and CDMA (code-division multiple access) [3]. FDMA and TDMA can be implemented on either narrowband channels (e.g., 12.5 kHz) or wideband channels (e.g., 25/30 kHz), whereas CDMA is restricted to the wideband architecture. Two principal switching techniques are available: circuit switching and packet switching. Circuit switching requires a complete end-to-end connection over both the wireless and wired segments before any voice or data can be sent. Packet switching divides the information into packets and transfers packets of data between network nodes over different connections until the packets reach their final destination and are reconstructed for the user. The main difference between these two methods is that circuit switching reserves the required bandwidth in advance, whereas packet switching acquires and releases it on an as-needed basis. Two other switching techniques are store-and-forward (in which the information is first stored and then forwarded, one hop at a time) and hybrid (variants and mixed forms of circuit switching and packet switching) transmission.

In general, mobile services can be classified into conventional (nontrunked) mobile systems and trunked mobile systems. Each conventional mobile system has one channel available for each specific group of users. In other words, users are grouped so each group of users is assigned to a different channel. This system is less efficient than trunking because users can only communicate if their channel is free, regardless of whether the other channels are free. Since each group of users shares a common radio frequency, the users end up competing for air time. The trunked mobile system allows all available channels to be shared among a large group of users, so that no one is waiting as long as free channels are available. For instance, a typical trunked land mobile radio system can have up to 28 channels, one of which is a dedicated control channel that automatically assigns the open channels to users. When a user wants to transmit, both the transmitting radio and the receiving radios are automatically assigned to one of the 28 available radio channels. This system is better at conserving spectrum, since no individual radio user will have to wait to transmit if a trunked radio channel is available. Depending on how the available spectrum is used, these systems can also be classified as narrowband or wideband. In the narrowband architecture, the total frequency band is split into several narrowband channels. In the wideband architecture, most (or all) of the spectrum is available to each and every user.

Land mobile radio can be roughly divided into two separate classes of service: public and private. Paging, cellular telephony, and personal communications services (PCSs) fall under the classification of public land mobile radio, although some small paging networks may be private. We discuss each of these public service systems in turn in the next three sections. When cellular telephony is discussed, we also summarize the popular cordless telephony systems developed in Europe and Japan for comparison. We then turn to a discussion of private land mobile radio systems. Although some recent systems tend to support both private and public services, we discuss them together because many of them are evolved from early private systems. After discussing radio data networks, short-range beacons, and satellites communication technologies, we conclude this chapter with issues related to integration of communications subsystems. Information on the remainder of the technologies listed in Table 8.2 can be found in [4-7]. To save space, some system names, modulation methods, and open protocols contained in some tables are not spelled out, interested readers may contact the manufacturers for the full names of the systems or consult [2,8,9] for the full names of the technical terms listed.

8.3.1 Paging

Paging is the simplest mobile radio system. A basic paging system is shown in Figure 8.1, where PSTN stands for the public switched telephone network and BS stands for base station. It includes an input source, the existing wireline telephone

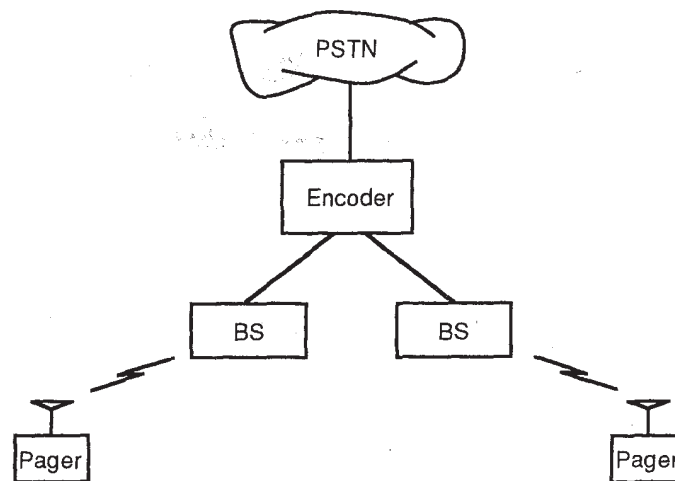


Figure 8.1 Paging network.

network (PSTN), the paging encoder equipment (Encoder), the base station (BS) controlled by the transmitter control equipment, and the remote pager. Input can be entered from a phone, a computer with a modem, a desktop page-entry device, or through an operator who enters the message. Once the message is entered, it is sent via PSTN to the encoder. The encoder checks the validity of the pager number and looks in the subscriber database for the pager address. It then converts the address and message into the paging signaling protocol (a special code). This encoded paging signal is then sent through the control system to the transmitters located at each base station. The base station broadcasts the signal on the specified frequency for its coverage area to beep the pager.

Paging was originally developed as a one-way communications service (from a base station to mobile or fixed receivers) that provides signaling or information transfer by tone, tone-voice, numeric or alphanumeric message display, etc. Recent advances in technology have led to the introduction of two-way paging. This makes paging technology even more attractive as a medium for vehicle communications.

Radio paging has many attractive features, which explains why public interest in the technology has grown by leaps and bounds during the last decade. Paging systems are fairly robust and most can reach customers located deep inside buildings. The technology is widely available, and pagers are becoming even more versatile and portable as paging systems continue to take advantage of digital signaling technology. Competition among service providers is also creating a host of options for consumers, ranging from basic tone-only models (which beep or vibrate when someone is needed) to alphanumeric devices that can display short messages or other advanced models. For instance, tone-and-voice models will notify the user of a call and then relay a voice message at the customer's convenience.

One interesting paging method is to superimpose paging signals on AM/FM radio or TV signals. This signal multiplexing technology is generally referred to as *broadcast subcarrier*. We discuss it in further detail in Section 8.3.6. As far as applications of paging in vehicle location and navigation are concerned, for example, a transportation authority could establish its own paging network in outlying areas by contracting with FM radio stations to use FM subcarrier signals to transmit short traffic messages along with the programming broadcast. FM radio listeners would never hear these pages unless they tuned to that specific subcarrier frequency with a proper radio receiver. Using such an approach, the authority could establish a rural network without having to construct its own costly radio towers and transmitters. Such a paging system could serve as an adjunct to other commercial services and would come into use whenever mobile workers traveled beyond the range of the privately run networks.

In spite of recent advances, paging is still primarily a one-way communications network. Its strengths lie more in transmission of short data messages. Paging is restricted by two main shortcomings. First, it is primarily an urban service, although some paging networks reach further into the suburbs than do cellular systems. Second

(and most critically), paging has traditionally been a one-way service. Users can receive brief messages, but they cannot acknowledge them unless they have a cellular phone or are near a wired telephone. Despite these drawbacks, paging is a highly affordable and widespread communications technology.

A multitude of paging companies and paging equipment is currently available on the market. For instance, Teletrac Inc. provides a paging service that operates in the 900-MHz band to provide two-way paging and tele-locator features. SkyTel's two-way paging service allows users to receive messages and respond with an acknowledgment via a button. This service takes advantage of wireless text messaging and adds the capability to send replies. Socket Wireless Messaging Service, or SWiMS, is a suite of services administered by Socket's National Dispatch Center and designed to turn an alphanumeric pager into a wireless link to the user's favorite communications resources: telephone, voice mail, fax, e-mail, modem, and paging system. Interested readers should consult the various trade magazines for wireless services based on radio paging technology and associated PCMCIA paging cards that are currently available. More information about various available paging equipment, devices, and protocols, can be found in [10,11].

8.3.2 Cellular

Cellular networks are full-duplex circuit-switched communication systems whose main components are of a set of mobile stations (MS, mobile or portable cellular phones), a base station (BS), a mobile switching center (MSC), and the public switched telephone network (PSTN) (Figure 8.2). The cellular system can provide both voice and data services over either analog or digital networks, although the main design objective of the analog network is to carry voice. Each coverage area in the network is divided into cells that are usually 1.5 to 30 km (0.93 to 18.65 miles) in diameter. The cell sites (BS) located at the center of each cell consist of a transmitter, receiver, control computer, and an antenna. The base station computer controls the frequencies, directs calls to open channels, and alerts the switching station when a call weakens. For simplicity, Figure 8.2 shows only two BSs associated with two cells. The switching station (MSC) connects the cellular phone to the local carrier and manages calls through frequency reuse, roaming, and hand-offs. Adjacent cells use different frequencies to avoid overlap, and cell site hand-offs occur automatically as each mobile station moves from cell to cell. A telephone carrier and its local office connects the call to the person being called and provides billing for the service.

The revolutionary concept behind cellular systems is the frequency reuse method proposed by Bell Telephone Laboratories. A certain frequency used by a channel in one area can be reused in another area. Each area consists of a cell. Cells using the same carrier frequency are located sufficiently far apart from one another such that interference can be within tolerance limits. As a result, each region initially

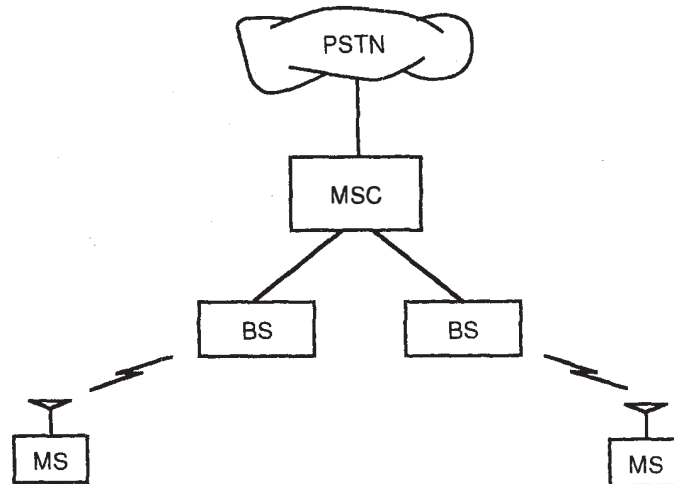


Figure 8.2 Cellular network.

served by a single base station before introduction of cellular systems can be divided into several cells. Each of these cells will have its own base station and its own set of channels. Therefore, more services can be provided within the same frequency range.

Several different cellular radio systems are used in different areas around the world. The characteristics of these systems are summarized in Table 8.3 (where some items have been quoted from [12]). Many books are available that discuss the principles and applications of these systems so we do not discuss them further in this section. More information on these systems and the underlying principles may be found in [9,13–16]. Note that analog cellular systems use FDMA for channel access, while digital cellular systems use either TDMA or CDMA and can still provide analog services using FDMA if designed to do so. Digital cellular systems promise similar advantages over analog cellular systems mentioned in the beginning of Section 8.3. In Table 8.3, m-b is the broadcast rate from mobile station to base station, b-m is the broadcast rate from base station to mobile station, ISDN stands for Integrated Service Digital Network, and CEPT stands for the Committee of European Posts and Telegraphs.

Various techniques and ideas for using cellular systems to provide broadcast data for vehicle communications exist even though cellular systems were originally designed for one-to-one communications. On analog voice networks, off-the-shelf modems must be used to modulate circuit data over cellular links as briefly discussed in Sections 8.4 and 10.2.2. These services are good for mobile users who need to transmit or receive large amounts of data in a single session without worrying about

Table 8.3
Characteristics of Cellular Radio Systems

<i>Name</i>	<i>Date of Operation</i>	<i>Channel Spacing</i>	<i>Frequency (MHz)</i>	<i>Number of Channels</i>	<i>Characteristics</i>
NAMT	1978	25 kHz	870-885 b-m	600	FDMA; increased to 1000 channels
NMT-450	1981	25 kHz	925-940 m-b 453-457.5 m-b 463-467.5 b-m	180	FDMA; low channel capacity; good radio coverage; suitable for rural areas
AMPS	1983	30 kHz	825-845 m-b 870-890 b-m	832	FDMA; city-based; higher capacity than NMT, but smaller cells
C-450	1985	20/10 kHz	451.3-455.74 m-b 461.3-465.74 b-m	222/444	FDMA
TACS plus ETACS	1985	25 kHz	890-915 m-b 935-960 b-m 872-888 m-b 917-933 b-m	1000 plus 640	FDMA; 50% greater capacity than AMPS, but smaller cells
NMT-900	1986	12.5 kHz	890-915 m-b 935-960 b-m	1999	FDMA; designated for cities; caters to hand-held portables
NAMPS	1992	10 kHz	825-845 m-b 870-890 b-m	2412	FDMA; one 100-bps digital channel; 14-character mobile terminated messages
GSM	1992	200 kHz	890-915 m-b 935-960 b-m	175 with 8 timeslots per channel (1400)	FDMA/TDMA; digital; ISDN capability; CEPT standard
TDMA (D-AMPS)	1993	30 kHz	825-845 m-b 870-890 b-m	832 with 3 users per channel (2496)	TDMA; digital; IS-54 standard
PDC	1993 1994	50 kHz 25 kHz interleave	824-849 b-m 940-956 m-b 1429-1453 m-b 1477-1501 b-m	1600 with 3 users per channel (4800)	TDMA, digital; for Japan only
CDMA	1995	1228 kHz	825-845 m-b 870-890 b-m	19-20 (could vary)	CDMA; digital; IS-95 standard

the extra call setup time and call disconnect time. In addition, a network overlaying the existing analog cellular network has been developed to address packet data applications. We discuss this network (CDPD) in comparison with other radio data networks in Section 8.3.5. See Section 10.4 for applications of the analog AMPS network (Table 8.3) in vehicle communications.

For comparison with cellular technology, Table 8.4 lists various cordless technologies that have been introduced in Europe and Japan. These systems are considered to be second-generation technology, mainly on the basis of the fact that they use digital signals and the fact that there are numerous base stations outside the home that can operate with a large number of individual handsets. By contrast, first-generation cordless technology is based on analog signals and only operates within residences between unique base station and handset pairs. This first-generation technology has found common use in the United States. Because of the relatively low cost and mobility, the new cordless telephony systems have become an integral part of personal communications services, as discussed in the following section.

8.3.3 Personal Communications Services

Personal communications services (PCS) have been defined by the U.S. Federal Communications Commission (FCC) to cover a broad range of radio communications services that free individuals from the constraints of the wireline PSTN and enable them to communicate when they are away from their home or office telephone [17]. Some literature refers to this technology using the term "personal communications network" (PCN). The PCS concept was developed out of the recognition that both cellular and cordless telephones have severe limitations. Therefore, PCS is supposed to combine the advantages of each system along with other added features to meet the communications requirements of people on the move. PCS has been proposed as a multienvironment, multioperator, and multiservice type system (Figure 8.3). Many people classify wireless systems based on analog technology as

Table 8.4
Characteristics of Cordless Telephony Systems

Name	Date of Operation	Channel Spacing	Frequency (MHz)	Number of Channels	Characteristics
CT2	1989	100 kHz	864–868 m-b 864–868 b-m	40	FDMA/TDMA; digital
DECT	1993	1728 kHz	1880–1900 m-b 1880–1900 b-m	10 with 12 users per channel (120)	TDMA; digital
PHP	1995	300 kHz	1895–1907 m-b 1895–1907 b-m	77 with 4 users per channel (308)	TDMA; digital

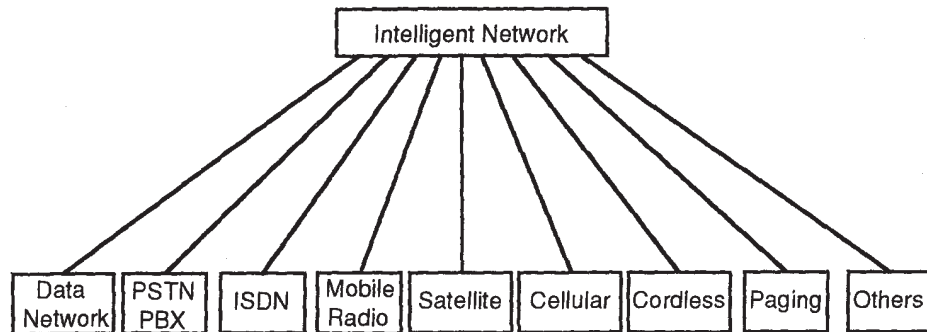


Figure 8.3 Proposed PCS architecture.

first generation, systems based on digital technology as second generation, and PCS the next major step in the evolution of communication systems, as third generation [18].

Although there is no exact definition or well-defined system architecture for PCS, the basic concept is to support highly competitive wireless communications services without any restrictions on providers' capabilities and imaginations. PCS frequencies have been divided into two main groups by the FCC: wideband (broadband) frequencies and narrowband frequencies. The wideband PCS frequencies lie in the 1.8- to 2.2-GHz band; these frequencies are intended for high-throughput voice and data applications. Mobile devices (or handsets) might be able to roam freely among homes, offices, and roads, even access the Internet, using devices evolved from cellular phones. Narrowband PCS frequencies have been allocated to the 900-MHz band; these frequencies are intended for short and burst voice and data applications. These mobile devices might be small, using devices evolved from two-way pagers, and other personal computing or communications devices. Both wideband and narrowband PCS should be able to support vehicle location and navigation.

Some believe that PCS can be thought of as a family or continuum of wireless communication services at lower power and with more (and smaller) cell sites than cellular systems. Instead of using large transmitter towers, PCS is envisioned as employing transmitter/receivers small enough to be mounted inconspicuously on the sides of buildings and on top of telephone poles. This results in much smaller transmission cells (often called *microcells*). The minimum PCS cell size is expected to be approximately 0.8 to 2.5 km (0.497 to 1.554 miles) in radius. Consequently, PCS will require a larger number of cells for the same coverage area. This results in dramatically lower power requirements (but probably worse interference between adjacent cell sites than occurs for cellular services). Because PCS devices will use lower power, the phones should be smaller, lighter, and able to run longer on a

single charge, perhaps even for weeks at a time. PCS is expected to be able to accommodate many more callers than cellular networks. As a result, promoters of this technology expect the rates for air time to be cheaper. In contrast, others think that there are other alternative architectures and implementations available. Some services will have more features and will presumably be more costly.

A wide variety of alternate versions of PCS have been proposed. Although the first U.S. PCS has been in commercial operation since November 1995, several other proposed systems are either still in test trials or in the process of being implemented. Many digital service specifications (including different voice-coding techniques and signal compression schemes to maximize spectrum usage) are being considered by rival PCS. Around the world, PCS has been allowed frequencies between approximately 800 and 3000 MHz. Proposed PCS services have included both licensed and unlicensed frequency alternatives [19]. The licensed services proposed for use include sophisticated pocket telephones, cellular phones at 2 GHz, and wireless fixed and short-distance links connecting the phones to microcells. Unlicensed services proposed for use include cordless business and home telephone systems, wireless local-area networks for computers, and wireless portable information devices. In short, PCS has the potential to improve mobile communications technologies existing today.

8.3.4 Private Land Mobile Radio Systems

As mentioned earlier, paging, cellular telephony, and PCSs are generally regarded as public land mobile radio services. In contrast, private land mobile radio services can be divided mainly into the categories of conventional, trunked, and specialized mobile radio (SMR). Although most SMR operations technically would fall under public services, the decision simply follows the private sector classification in the FCC frequency regulations (FCC: 47 CFR, Part 90). In the conventional radio system, users must share a common RF channel and compete for air time by monitoring the system for an open channel as described at the beginning of Section 8.3. Trunked radios are controlled by microprocessors, which act as automatic switchboards. Trunked radio system shares repeater(s), antennas, and channels. SMR systems were introduced to the United States in 1974 by the FCC to provide private land mobile services on a commercial basis. A SMR system can be either conventional or trunked, although most SMR systems are now trunked. These three types of systems can take different forms, such as individually owned, shared, voice only, data only, and integrated voice and data.

Although classified roughly under the title of this section, some recent systems intend to serve both the private and public sectors, such as the iDEN system, which is discussed later, and the European and Japanese systems listed later in Table 8.6. Because of these recent developments, the differences between private and public

land mobile radio systems are becoming smaller, which has led to a blurring of the distinctions between these two categories. Unlike cellular technology, it is difficult to find public references discussing certain recent systems such as iDEN. Therefore, we cover this system in this section. General and in-depth descriptions of various land mobile radio systems can be found in [20,21].

Traditionally, private land mobile radio activity has been in the areas shown in Table 8.5 (after [1]). There are many different types of vehicle applications. One example of private data systems for vehicle location applications is the Motorola Private DataTAC system, which has been used in the United States and has a gross data rate of 9.6/19.2 Kbps. Another example is the system used for the dynamic route guidance system discussed in Chapter 11. Most radio equipment currently in service uses analog voice and would require modems to modulate, transmit, and receive position and traffic data over the air. As new systems are implemented, many are moving to digital technology.

Many land mobile radio systems combine voice and data via a trunked radio system. Several such trunked radio systems are shown in Table 8.6. The last four are mainly used in Europe (MPT-1327 and TETRA) and Japan (MCA and DMCA). The remainder of the systems are available in American and other markets.

Trunked radios have different implementations. A private trunked system can consist of a single site or a large multiple-site system. Each site has multiple repeaters that can receive, amplify, and retransmit radio signals. One repeater at each site is a control channel, which transmits a continuous stream of data (digital codes). Mobile and portable radios communicate with the site over the control channel to request an available channel. In a trunking system, each radio is assigned an individual unit ID and is also programmed by the system administrator with one or more groups that are user selectable. These groups, usually referred to as *talkgroups*, are

Table 8.5
Traditional Private Land Mobile Radio Service Categories

<i>Land Transportation</i>	<i>Public Safety</i>	<i>Special Emergency</i>	<i>Industrial</i>
Auto emergency	Fire	Beach patrols	Business
Motor carrier	Forest conservation	Communications standby facilities	Forest products
Railroad	Highway maintenance	Disaster relief	Manufacturers
School buses	Local government	Emergency repair of public communications facilities	Motion picture
Taxicab	Police	Handicapped	Petroleum
	State and federal governments	Medical services	Relay press
		Rescue services	Special industrial
		Veterinarians	Telephone main-tenance
			Utility business

Table 8.6
Representative Trunked Voice and Data Radio Systems

<i>Company</i>	<i>System</i>	<i>Data Protocol</i>	<i>Modulation</i>	<i>Channel Access</i>	<i>Gross Data Rate</i>	<i>Voice</i>	<i>Channel Spacing</i>
Ericsson	EDACS	RDI	Binary FSK	FDMA	9.6 Kbps	Analog/ digital (Aegis)	25 kHz
E. F. Johnson	LTR	LTR	FSK	FDMA	0.3 Kbps	Analog	25 kHz
E. F. Johnson	Multi-Net	Multi-Net	FSK	FDMA	9.6 Kbps	Analog	25 kHz
Motorola	iDEN	LAPi	M-QAM	TDMA	64 Kbps	Digital	25 kHz
Open	APCO	CAI, IP	QPSK-C	FDMA	9.6 Kbps	Digital	12.5 kHz
Motorola	Project 25 ASTRO	CAI, IP, FLM	QPSK-C	FDMA	9.6 Kbps	Digital	25 kHz 12.5 kHz
Motorola	SmartZone SmartNet II	MDLC MDC-4800 RD-LAP, ..	FSK	FDMA	1.8/3 Kbps 4.8 Kbps 9.6/ 19.2, ...	Analog	12.5/25 kHz
Open	MPT-1327	MPT-1327	FFSK	FDMA	1.2 Kbps	Analog	12.5/25 kHz
Open	TETRA	TETRA	$\pi/4$ DQPSK	TDMA	36 Kbps	Digital	25 kHz
Open	MCA	MCA	MSK	FDMA	1.2 Kbps	Analog	12.5/25 kHz
Open	DMCA	RCR Standard-32	M-QAM	TDMA/ TDM	64 Kbps	Digital	25 kHz

groups of radios that will transmit and receive each other's communications. When a radio is set to a particular talkgroup, that radio will only be able to transmit and receive information from other radios in the same talkgroup. With talkgroups, each radio user will only hear the communications desired. One typical method for segmenting users into groups on a trunking system is according to functional responsibilities.

Requests for telephone interconnects are also handled through the control channel in similar fashion. A repeater is linked to a phone line and connected to the PSTN. This enables a radio system user to communicate with persons not on the radio system via standard telephone. Depending on the radio system, the conversation will either be in full duplex or half-duplex. Full duplex operates like a typical phone call, whereas half-duplex implies that the radio can only transmit or receive but not both simultaneously.

Some systems use a different approach. Instead of using a separate control channel to communicate with the site, they utilize slower and low-speed data on

the voice channels. Although the trunking operation is similar, some features such as "busy queuing and callback" are not available in a system design of that type.

As mentioned earlier, SMR systems were introduced in the United States in the early 1970s, although significant amounts of equipment and systems did not begin to appear until the early 1980s. Trunked SMR sites can be linked, and also include data to provide a nationwide voice and data communications network. In the early 1990s, new radio systems (enhanced SMR or ESMR) were developed to compete with and exceed the services offered by cellular radio carriers. Licenses of many independent private SMR service providers were purchased and used together to provide more capacity and services on a single large network that integrates voice dispatch, wireless phone, text messaging (paging), and data transmission capabilities into one handset with cellular-like coverage. Unlike cellular systems, these ESMR operators have agreed to a single standard—the one set by the iDEN system discussed next.

Integrated Dispatch Enhanced Network

The integrated Dispatch Enhanced Network (iDEN) is a digital communications system that combines the capabilities of a standard dispatch system with that of a cellular-like interconnect system. iDEN was developed by Motorola and is now in use in Argentina, Canada, Israel, Japan, the United States, and several other countries. iDEN uses an advanced proprietary modulation technology consisting of a speech compression scheme enabling six or three communication paths to be carried over a single 25-kHz RF channel. This technology is a radical change in the basic modulation techniques and hardware platforms that have dominated mobile radio for the past 40 years.

A key limiting factor for expansion of wireless communications products is global spectrum availability. By allowing six communication paths over a single channel, the iDEN system leads to a sixfold increase in efficiency. Furthermore, iDEN provides numerous other benefits including regional networking, full-duplex telephone interconnect, and integrated service. iDEN gives SMR operators the ability to integrate wireless phone, text messaging, two-way radio, and data transmission into one network using a single subscriber unit, referred to as a mobile station (MS, either mobile or portable). iDEN eliminates the need to carry a phone, a pager, a two-way radio, and a modem to stay in touch.

The digital technology used in iDEN provides additional benefits beyond integration of various services, including the following: more consistent voice audio quality throughout more of the coverage area, fewer dropped calls, improved security, and better system access.

Several other additional features of iDEN are listed below:

- Smaller hand-held portables;
- Full-duplex service (like cellular);
- Messaging services similar to digital paging that will allow rapid response;
- Superior data services to avoid lengthy verbal message exchanges and provide office mobility;
- Reduction of busy waiting time due to more efficient use of the RF spectrum;
- Advanced mobile telephone services;
- Advanced dispatch services (group communication with transparent networking);
- Flexible network scaling.

This technology allows fleets of users equipped with MS units to communicate with one another or the PSTN. The RF system consists of strategically located enhanced base transceiver systems (EBTSs), which are linked to base site controllers (BSCs) that direct the paths to either a dispatch application processor (DAP) for dispatch-type call processing or to a mobile switching center (MSC, or Northern Telecom Switch) for telephone interconnection (Figure 8.4). The BSC manages the RF channel, and the MSC maintains control over which users can enter the network, what their operating privileges should be, and where they are presently located within the system. Information previously entered into the home location register (HLR) database is used by the MSC for this maintenance. The EBTS site consists of one or more base radios (BRs), an RF distribution system, a BR frequency reference, a site synchronization receiver and antenna, a local-area network interface, and BR antennas. The EBTS provides the radio communications link between the land network and MS units. The BRs perform the communications with the MS units, sending both control information and compressed speech over a multiplexed radio channel. The EBTS can be configured to support multiple RF frequencies in an omnidirectional or sectorized configuration.

In Figure 8.4, VLR stands for the visitor location register, which contains and is updated with current information such as most recent location, and the feature provisioning table. AUC stands for the authentication center, which holds the authentication keys for all MS users. MPS stands for the metro packet switch, which is a collection of components providing wide-area communications services. These services include distribution of incoming packets and dynamic replication of voice packets for transport to the respective BSC and corresponding EBTS. The switch provides connectivity between iDEN sites during a dispatch call.

The gross radio-channel bit rate for an iDEN system is approximately 64 Kbps. Most earlier systems utilized FSK (frequency shift key) modulation. In contrast, the iDEN system uses M16-QAM (quadrature amplitude modulation). This linear modulation provides a highly desirable combination of modulation efficiency, channel sensitivity, acceptable carrier-to-interference ratio trade-offs, and low adjacent channel interference.

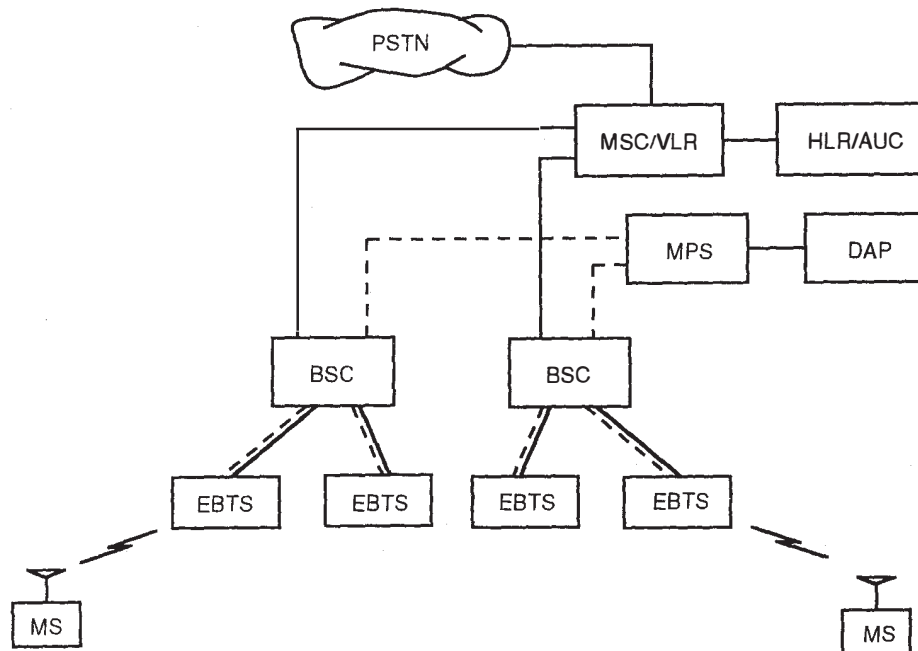


Figure 8.4 Major elements of an iDEN system.

The iDEN equipment uses frequencies of 806-821/851-866 MHz, with a carrier spacing of 25 kHz. The separation between transmitting/receiving pairs is 45 MHz. The ability to accommodate 12.5-kHz offsets is also included. iDEN equipment may also be configured to operate in higher frequency ranges, such as 1453-1465/1501-1513 MHz with 48-MHz transmitting/receiving pair separation.

The compression algorithm used for digitally encoding/decoding speech is known as Vector Sum Excited Linear Predicting (VSELP). VSELP is a member of a class of speech coders called Code Excited Linear Predictive Coding (CELP) and Stochastic Coding (SC) or Vector Excited Speech Coding (VESC). It is the vocoder used in iDEN.

The iDEN implementation divides the RF carrier into discrete time slots of 15 ms (including overhead for transmitter turn-on, training and synchronization, and propagation delay). Auxiliary data embedded within each slot provide for associated signaling in addition to the normal digitized voice and data traffic. A set of slots on one group of RF carriers is devoted to trunking control. Inbound control-channel access is obtained via a reservation protocol. TDMA is the channel access technology used by the iDEN system.