THE UNIVERSITY OF CALGARY

PROTOTYPE FOR A LAND BASED AUTOMATIC VEHICLE LOCATION AND NAVIGATION SYSTEM

by

CLYDE B. HARRIS

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF SURVEYING ENGINEERING

CALGARY, ALBERTA

December, 1989

ISBN 0-315-61640-7

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a dissertation, entitled "Prototype for a Land Based Automatic Vehicle Location and Navigation System", submitted by Clyde B. Harris in partial fulfilment of the requirements for the degree of Master in Engineering.

Gerard Lachapelle

Dr. G. Lachapelle (chairman),
Department of Surveying Engineering

Dr. E.J. Krakiwsky
Department of Surveying Engineering

MG Sideris

Dr. M. G. Sideris
Department of Surveying Engineering

JF Morrall

Dr. J. F. Morrall
Department of Civil Engineering

Date 15 December 1989

ABSTRACT

An Automatic Vehicle Location and Navigation (AVLN) System for stand-alone, land based, street applications provides drivers and navigators with real-time position and location of their vehicle while operating in a defined network. It also has the built in capability of supplementing route guidance advice to the driver in real-time, en route from the vehicle's starting location to a given destination.

The focus of this project was to design an AVLN prototype system and the goal was to gain experience and knowledge from building and assembling various components within. Investigations were made into positioning, location monitoring, database construction, and navigation operations, with the intent to integrate all of these.

A kinematic position filter which integrates dead reckoning by differential odometry, map matching, and GPS was designed and field tested. Data of the three positioning information sources was logged simultaneously over a test run and later fed through developed filter software. Results indicate that the filter works well in general but systematic error is introduced during some map matched azimuth updates. It is suggested that another direction sensor be integrated into the design to add redundancy and reliability to the system and also odometer quantization should be minimized.

A digital map database was built for a small, community size, test network. Specific details outlining all data constructs used to support the system's map matching, route determination, and route guidance functions are addressed.

The prototype system demonstrates in real-time, on an in-car computer map display, vehicle positioning and location monitoring by differential odometer dead reckoning augmented with map matched updates. The system also demonstrates GPS positioning of the vehicle in real-time, tracking the vehicle onto a digital map with fix updates every two seconds. Much knowledge and experience was gained from the field trials and thus lead to future recommendations.

Route determination and guidance software was constructed as a separate but compatible module built around the same system database. Proof of concept of the real-time route guidance advice is performed in lab by simulating vehicle location. Instructions include approaching intersection turning procedures, approaching intersection street names for user identification, and distance remaining. In this manner, three fundamental types of route guidance primitives namely, directional advice, landmark information, and trip duration status, were employed.

ACKNOWLEDGEMENTS

To begin, I would like to acknowledge and thank my supervisor Dr. Edward J. Krakiwsky, from whom I received knowledgeable advice, understanding, and technical support. His unprecedented perpetual desire, energy, and positive attitude will never be forgotten.

I would also like to pay tribute to my fellow colleagues, my professors in graduate studies, the Surveying and Civil engineering technical support staff, the first U of C AVLN test car - my 1976 Dodge Dart, and to those who played an active role on the University of Calgary's AVLN team, especially Hassan Karimi.

Externally, appreciation is sent to Mr. John Adams of Pulsearch Consolidated Ltd., for his technical consultation, and to Pulsearch, for financial assistance. Also thanks goes out to Statistics Canada for sponsoring City of Calgary digital map data. The NSERC operating grant of Dr. E. J. Krakiwsky provided me with a Graduate Student Research Assistantship. Thanks is also extended to Motorola's IVHS Group for their positive encouragement of the completion of this dissertation prior to my employment with the firm.

A special recognition and thanks is directed to my fiancé, Elise Honeyman, for typing and preparing this manuscript. Thank you Elise for your patient support, love, and gracious understanding during this uphill climb.

Finally, I would like to thank my family and especially dedicate thanks to my parents:

to my Mother, Doris E. Harris, for her years of love, patience, financial aid, and inspiration to always push onward and upward; and,

to my Father, Alan V. Harris, for his love, directional guidance, financial assistance, and fatherly advice.

Thank you for all of the sacrifices you two have made. In honour, I dedicate a lifetime of remembrance and appreciation to you both.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF NOTATION

a_e	semi-major axis of ellipsoidal earth
a	semi-major axis of error ellipse
α	number of arcs
$\boldsymbol{\alpha}'$	arc subset
$\boldsymbol{\alpha}$	arc set
A	first design matrix
ADJ_LIST	adjacency list array
AMAT	adjacency matrix
b_e	semi-minor axis of ellipsoidal earth
b	semi-minor axis of error ellipse
B	second design matrix
c	speed of light in a vacuum
$\mathbf{C}_\ell$	covariance matrix of observables
$\mathbf{C}_{odom}$	odometer sensor noise covariance matrix
$\mathbf{C}_{\hat{x}}$	covariance matrix of unknown parameters
$\mathbf{C}_{\hat{y}}$	covariance matrix of system noise
$\mathbf{C}_{\hat{\delta}.}$	system state covariance matrix
$\mathbf{C}_{\varepsilon}$	covariance matrix of system noise residuals
d_{ion}	ionospheric range delay
d_{trop}	tropospheric range delay
dt	GPS satellite clock error
dT	GPS receiver clock error

d^L	total distance travelled by left wheel
d^R	total distance travelled by right wheel
e_e	eccentricity of ellipsoidal earth
E	mathematical expectation
E	expansion node
h	ellipsoidal height
i	epoch i
$\mathbf{I}$	identity matrix
$\mathbf{J}$	Jacobian matrix
k_o	UTM scale factor
$\mathbf{K}$	Kalman gain matrix
lkdist	distance of a link
$\hat{\boldsymbol{\ell}}$	vector of observables
m	number of arc attributes (i.e. properties)
M	meridian plane radius of curvature
n	number of nodes
$\mathbf{n}$	node set
$\mathbf{n}'$	node subset
N	transverse radius of curvature
NODE_ARC	node-arc incidence array
p	GPS pseudo-range
$\mathbf{p}$	arc property node set (i.e. node attribute set)
r	radius of curvature of circular arc
$\hat{\mathbf{r}}$	residual vector
R_e	radius of the earth

R^L	radius of left wheel's curvilinear path
R^R	radius of right wheel's curvilinear path
R	Root node
s	satellite
s_a	circular arc length
s	start node
S	successor node
$\boldsymbol{S}$	successor node set
t	terminating node
TRACK	perpendicular length between wheel planes along axle
x	UTM Easting
$\hat{\mathbf{x}}$	vector of unknown parameters
x_o	UTM false Easting
y	UTM Northing
$\hat{\mathbf{y}}$	vector of system noise
X	ECEF Cartesian reference frame X coordinate
$\mathbf{w}$	misclosure vector
Y	ECEF Cartesian reference frame Y coordinate
Z	ECEF Cartesian reference frame Z coordinate
α	UTM grid azimuth heading
$\hat{\delta}$	system state vector
δt	difference between satellite signal transmission time and receiver signal reception time

$\delta\hat{x}$	state correction element to x
$\delta\hat{y}$	state correction element to y
$\delta\hat{\alpha}$	state correction element to azimuth α
$\Delta\alpha$	change in heading azimuth
Δd^L	change in distance travelled by left wheel
Δd^R	change in distance travelled by right wheel
$\overline{\Delta d}$	mean change in distance travelled
Δx	change in UTM Easting
Δy	change in UTM Northing
$\nabla\Delta d$	cross TRACK difference of left and right wheel distance changes
$\Delta\phi$	change in geodetic latitude
$\Delta\lambda$	change in geodetic longitude
ε	system noise residual vector
ϕ	geodetic latitude
λ	geodetic longitude
θ	orientation azimuth of error ellipse's semi-major axis
θ_a	arc angle swept by curvature radius
ρ	geometrical range distance between GPS satellite and the receiver
σ	standard deviation
$\varnothing$	null array
(+)	filter update
(-)	filter propagation

LIST OF ACRONYMS

AEC	Automatic Error Correction
AMF	Area Master Files
AVLN	Automatic Vehicle Location and Navigation
CIO	Conventional International Origin
CM	Central Meridian
CPU	Central Processing Unit
CRT	Cathode Ray Tube
CT	Conventional Terrestrial
DMI	Distance Measuring Instrument
DOP	Dilution Of Precision
DR	Dead Reckoning
ECEF	Earth Centered Earth Fixed
GIS	Geographic Information System
GPS	Global Positioning System
HDOP	Horizontal Dilution of Precision
HUD	Head Up Display
INIT	Initialization
LCD	Liquid Crystal Display
LOP	Line of Position
LORAN	LONg RANge positioning system
MM	Map Matching
MMI	Man Machine Interface
NAD27	North American Datum 1927
NAD83	North American Datum 1983

NAVSTAR	NAVigation Satellite Timing And Ranging
PDI	Periodic Distance Interval
RMS	Root Mean Square
SEP	Spherical Error Probable
TANS	Trimble Advanced Navigation System
U of C	University of Calgary
UTM	Universal Transverse Mercator
WGS84	World Geodetic Datum 1984

1. INTRODUCTION

1.1 Historical Overview

For centuries, Automatic Vehicle Location and Navigation (AVLN) technology for land based applications has slowly been unfolding, emerging from ancient civilizations of Egypt, Greece, Rome, and China, only to dawn now, during mankind's celebrated age of space and computer technology, here at the end of the 20th century. Ranging from ancient mechanical odometers to modern electronic satellite receivers, this technological evolution is marked by significant breakthroughs described by colorful and fascinating accounts which have stood the test of time [French,1986a].

Classical origins of AVLN technology root back nearly 2000 years ago. The first AVLN related device to emerge was the hodometer, the ancient forefather of the present day automobile odometer; the word hodometer originating from the Greek words *hodos*, meaning 'way', and *metron*, meaning 'measure'. Heron of Alexandria (circa 60 A.D.), one of the most noted engineers of the ancient University of Alexandria (Greek at that time), wrote of an ancient hodometer which employed a reduction gear train interlocked to the wheel of a cart [Strandh,1979]. Distances journeyed were recorded by dropping stone pebbles into a container at regular intervals representing perhaps the first AVLN related analogue-to-digital format ever used. Other great men associated with early developments of the hodometer were as follows: Ctesibius (circa 300 - 230 B.C.), the first great engineer of the University of Alexandria [Bedini and de Solla Price, 1967]; Archimedes (circa 287 - 212 B.C.), the renowned Greek scholar [Sleeswyk, 1981]; and Vitruvius (circa 20 A.D.), the famous military architect and engineer of the Caesars of

Rome [Sleeswyk, 1981]. Interestingly, Sleeswyk [1981] speculates Vitruvius' hodometer may have been used as an ancient data collector to compile contemporary Roman travel guides such as the tabular Peuteringiana which still exists today.

Outside of the Mediterranean regions, parallel developments materialized at roughly the same time in ancient dynasties of the Chinese empire. Chinese hodometers were known as chi li ku chhê (automatic mile-measuring drum chariots), ta chang chhê, or chi tao chhê. Some operated in the following manner: "at the end of each li traversed" (one li approximately 500 metres) "a wooden figure struck a drum while at the end of each ten li another figure struck a gong" [Needham, 1965]. Initial developments of these mile-measuring drum chariots date back to sometime between the 100 B.C. and 300 A.D. evolving from musical drum carriages used as band wagons in colorful parades during imperial state processions.

Shortly after the hodometer, the Chinese developed the remarkable ancient south pointing carriage, truly a classic in AVLN antiquity. On top of this double wheeled carriage rode an ornamental statue whose outstretched arm was believed to have always pointed southward (the ancient Chinese cardinal direction). According to ancient legend, the carriage was constructed for two main reasons: one, to guide the forces of the imperial Chinese army through smoke screens during battle against the great rebel Chhih-Yu; and two, to guide the Yueh-Shang ambassadors during the time of Chou Kung (circa 1000 B.C.) back home to the far south. Estimates of the carriage's true origins, however, do not date back this far and credit is directed towards Chang Hêng (circa 120 A.D.) and Ma Chün (circa 255 A.D.) of the later Han periods [Needham, 1965]. Historians once believed the statue's pointing capability was based on principles of magnetism making the carriage a decorative, magnetic compass. Contrary to this belief its pointing capability is now recognized to be a property of mechanical differential wheel gearing, marking the first related usage of differential odometers in the history of AVLN technology.

Spanning the years from the ancient world up until the advent of the automobile, AVLN technology quietly evolved. During this era, the magnetic compass was invented (circa 1111 A.D.) [Needham, 1965]. As well, hodometer developments were revised by the great Italian inventor Leonardo DaVinci (circa 1452 - 1519 A.D.) [Hart, 1925] and by the versatile German monk and natural scientist Athanasius Kircher (circa 1602 - 1680 A.D.) [Strandh, 1979]. During the latter half of the 16th century, Europe's preoccupation with contemporary map making efforts gave rise to the development of an instrument which integrated a hodometer with a magnetic compass (i.e. dead reckoning on land) [Bedini and de Solla Price, 1967]. This integrated device could automatically record distances and directions simultaneously by imprinting a spiked compass needle at regular intervals of distance into a hodometer synchronized, paper strip roll [Price, 1957]. Following these hodometer revivals, numerous distance measuring applications appeared ranging from civic taximeters determining cab fares to the recording of marched journeys by British troops commanded by General Arthur Wellesley during the Peninsular War (circa 1808 - 1814 A.D.) [Sleeswyk,1981]. Even Thomas Jefferson, former president of the United States, adopted the hodometer on numerous occasions during the early 1800's, feeling that recording the distances he travelled might be important in the future [Bedini, 1988].

Over the last century, since the advent of the automobile, a surge in AVLN technological developments has occured. Inventions of the early 1900's were highlighted by the Jones Live-Map in 1909 and the Chadwick Road Guide one year later [French, 1987a]. Years ahead of its time, the Chadwick concept mechanically resembled modern AVLN route guidance, featuring on-board alarm bells and signal arms controlled by programmed route discs (i.e. punched cards) to guide the driver accordingly. Transpiring at approximately the same time period, between the years of 1900 to 1924, an American, E. Sperry, and a German, H. Aschütz, made initial developments of the gyrocompass for contemporary naval navigation [Schwarz, 1986]; technology which today has found its way into many navigational developments including land-based applications. During the second World War, the U.S. military constructed a vehicular odograph (jeep's position plotted on paper map), Doppler radar techniques for positioning were tested [Sonneberg, 1978], and the electronic radio navigation system LORAN (LOng RANge positioning system) was born [Appleyard et al., 1980]. As well, from 1940 to 1946, the first operational inertial guidance system was developed [Schwarz, 1986]. In the 1960's satellite positioning was pioneered; roadside proximity beacons were implemented; and in the early 1970's, a clever and economical position and location update concept entitled map matching for on-road AVLN applications was conceived [French 1986b].

Recently, in the 1980's, research into AVLN systems has increased tremendously. This popularity is primarily due to recent advancements in microcomputer technology, Geographic Information Systems (GIS), mobile communications, and vehicular positioning equipment allowing for practical, on-board, state-of-the-art navigation and information processing. Out of these advancements, much AVLN interest has grown both locally and abroad [Krakiwsky et al., 1987a; Krakiwsky et al., 1987b; Case, 1986]. Upon review of these current interests, it is clear that the mid 80's has truly marked the dawn of AVLN systems [Krakiwsky et al., 1987a], shining a promising light towards the future of AVLN systems in the 21st century.

1.2 Definition of AVLN Systems

The term AVLN is quite general in nature and may conceivably be applied to any vehicular environment be it land, sea, air, or space. As the focus of this research is specifically aimed at land based AVLN systems with emphasis on on-road operations, reference to the term AVLN throughout the report will be made in this context.

The concept of AVLN systems for on-road, land based applications has been defined in several ways. This pluralism is primarily due to the wide spectrum of system classes. Six major AVLN classes based upon digital communication links between the vehicle and a possible information source exist [Karimi and Krakiwsky, 1988a]. Of these six classes two general divisions prevail, namely: stand alone systems (e.g. private motorists) and dispatch systems (e.g. fleet management).

Krakiwsky et al. [1987a] state that an AVLN system allows a land based user to:

> "(a) position a vehicle using signals from satellites and information from on-board differential positioning devices;
> (b) plot the position on a CRT or flat panel display;
> (c) call up a digitized-electronic map of the area and see the vehicle's position relative to a desired location(s); and,
> (d) obtain instructions (visual and audio) using an expert system on how to proceed from the present location to the desired location."

Skomal [1981], on the other hand, defines such systems as:

"an assembly of technologies and equipment that permits centralized and automatic determination, display and control of the position and movement of multiple vehicles throughout an appropriately instrumented area."

No matter how one defines such systems they all possess one common objective. This objective involves the automatic and real-time integration of positioning devices with relevant information systems which enable the users to benefit in some form or another by relating vehicle location to surrounding environment [Krakiwsky et al., 1988].

Both the terms location and navigation in the acronym AVLN were carefully selected in the context of this research. Location not only suggests knowledge of the vehicle's position but more so to the vehicle's position relative to a particular environment; in this case the digitally defined street network. It extends one step further than just the term positioning which implies coordinate determination only [Harris et al., 1988]. Both position and a digital map database are required for location determination, a fundamental prerequisite for automatic, real-time street navigation. Navigation on the other hand regards the provision of directional information, optimal or otherwise, on how to travel from the vehicle's present location to a given destination or set of way points. Intrinsically this entails both route determination and route guidance. Together, the terms location and navigation describe a system with the capabilities to monitor present vehicle positions and locations as well as, provide route guidance advice recommending how to manoeuvre and navigate a vehicle appropriately to a prespecified destination.

1.3 Application Sectors

All AVLN applications source from at least one of four major incentives, namely:

1) economic gain;

2) security enhancement;

3) safety improvements; and,

4) recreational benefits.

Four broad applications sectors of AVLN systems have been categorized by Krakiwsky et al. [1987a] according to vehicle type. These sectors are:

1) commercial;

2) private;

3) civic; and,

4) military.

Case [1986], French [1987a], and Covil et al. [1987] discuss future applications of AVLN systems. With a little imagination, it is easy to envision numerous AVLN applications when considering the various functional capabilities of such systems. For example, rent-a-car agencies, delivery and courier fleets, municipal services (e.g. street maintenance and waste removal), security and armoured car fleets, emergency services (i.e. police, fire, and ambulance), taxi companies, continental bus and trucking lines, and even the private automobile owner touring on vacation may benefit from such systems, to name only a few. Truly, it is the application itself which inherently defines the system's specific requirements and limitations and likewise governs the various navigational and informational strategies to be incorporated.

1.4 Present AVLN Methodologies

Current on-road AVLN methodologies are governed by attempting to resolve the problem of locating and navigating a vehicle in an ever growing, complex maze of city streets and highways as opposed to the more conventional, open terrain navigation problem. Due largely to the development of the silicon chip, affordable and pragmatic, on-board electronic positioning and processing has only recently become possible, opening the doors to a wide variety of new and practical AVLN system methodologies and configurations. Present market driven systems host a multitude of hardware and software features [Cooke, 1985; Krakiwsky et al., 1987a] including numerous positioning devices, microprocessors, computers, input and output facilities, storage mediums, programming languages, algorithms, etc. Of the many configurations possible, three underlying components fundamental to any AVLN system prevail, namely: positioning methods, navigation strategies, and digital map databases. The selection and interrelation of these three components forms the backbone of any AVLN system and will dictate its overall performance and market potential.

Positioning a vehicle may be accomplished by relative or absolute positioning methods or a combination of both. Positional accuracies demanded by most AVLN systems are in the general range of 30 to 50 metres. Relative positioning is performed by dead reckoning techniques which determine vehicle coordinates relative to a known starting point by simultaneously keeping track of distance and directional changes made. A dead reckoning technique gaining considerable attention at the present time is differential odometry. By averaging and differencing measurements made by two odometers, one mounted on each wheel of the same wheel pair, distance and direction changes may be observed and in turn vehicle coordinates mat be determined. Other feasible but less suitable dead reckoning possibilities include various combinations of odometers (wheel or drive shaft sensed), ground speed, Doppler based velocimeters (electromagnetic and acoustic signaled), accelerometers, magnetic compasses, and gyrocompasses. Inertial positioning systems (i.e. accelerometers plus gyrocompass) are accurate but presently much too expensive for the application addressed herein. Magnetic compasses provide absolute directions (i.e. azimuths) as opposed to relative directional changes but are too sensitive to external magnetic anomalies, which, unlike internal magnetic fields generated within the vehicle, are difficult to account for. The performance of Doppler based velocimeters are subject to problems associated with: undetected, non-horizontal velocities; specular reflection which depends upon the compatibility between varying dimensions of different surface type (e.g. ice, snow, mud, gravel, and asphalt) irregularities and the transmitted wavelengths used [Petrov, 1976]; and, signal interference due to sensor windows plagued by dust and mud.

Differential odometry, like all other relative positioning methods, requires periodic positional updates to control error accumulation. To solve these error growth problems, several systems currently employing dead reckoning devices (such as Etak's Navigator and Bosch Blaupunkt's EVA) make use of map matching techniques to provide the necessary coordinate update information [French, 1986b]. By recognizing turns made at identifiable intersections and road curvatures, such systems correlate or match a vehicle's path to a digital street map. When a match is made, corresponding map coordinates and azimuths are retrieved from a database and used to update the position, location, and heading of the vehicle.

Although augmented dead reckoning techniques (i.e. dead reckoning augmented with map matching [Honey and Zavoli, 1985]) provide an economical and straightforward means of positioning, one obvious limitation still exists. In order to control error accumulation, map matches must be made. Thus if turns at intersections do not occur periodically, errors will accumulate to such a degree that a lost situation will be inevitable. Honey et al. [1986] report that Etak's Navigator experiences a lost situation after approximately 400 kilometres of normal travel on average. Undoubtedly, on long straight roads and highways forming municipal, provincial, and federal transportation networks, this problem will become more acute [Krakiwsky et al., 1988]. Upon instances when the system becomes lost, the user will be forced to manually reinitialize; an inconvenience most users would not appreciate from an 'automatic' vehicle location and navigation system. In addition to lost situations arising from infrequent map matching, systems employing augmented dead reckoning techniques will also require periodic manual reinitializations whenever significantly large, vehicular movements occur which are undetected by the dead reckoning sensors (e.g. ferry crossings, towing, etc.)..

Absolute positioning methods seen in modern AVLN systems include: roadside proximity beacons; hyperbolic radio positioning systems such as LORAN-C; and, satellite positioning systems such as the Navstar Global Positioning System (GPS). Of these, the GPS satellite positioning system displays the most potential overall. Systems which rely on road-side proximity sensors to position a vehicle and relay guidance instructions to a driver possess two major infrastructure problems, namely limited operating regions and large implementation costs. These two problems led to the downfall of such attempts made during the 1960's and early 1970's. Hyperbolic radio positioning systems do not require signposts at every intersection to position a vehicle but do not provide full continental coverage at this time. The widely used coastal radio positioning system LORAN-C has limited inland coverage world-wide [Fuentes, 1987] but future expansions are projected to accommodate the entire continental United States and major parts of Canada. Unfortunately, even then, only positional accuracies in the 200 to 300 metre range are achievable with LORAN-C techniques [French, 1987b]. LORAN-C positioning using differential techniques can increase accuracies considerably (approximately 50 metre level) and properly calibrated (i.e. location errors) shows repeated accuracies of 20 to 50 metres. These accuracies depend upon numerous factors (e.g. proximity between roving and control stations, in-land LORAN-C infrastructure) and falls short to differential GPS satellite positioning with accuracies one order of magnitude better. For certain applications especially for position monitoring, LORAN-C may be a reasonable and economically alternative until GPS is in full operational force.

The Navstar GPS satellite positioning system is a multi-user system providing all-weather, real-time, single point kinematic positions to within 30 metres of accuracy [Wells et al., 1986], and in differential mode to within 5 metres or less [Cannon, 1987; Lachapelle et al., 1986]. Unfortunately, GPS is presently at its prototype stage of development and world-wide, 24 hour coverage will not be available until the full satellite constellation is complete sometime in the early 1990's. Consequently, present GPS equipment costs are relatively expensive which has persuaded many AVLN system designers not to implement GPS technology, especially those attacking present markets. GPS equipment costs have, however, dropped considerably over the past few years and it is strongly believed that when the GPS is fully deployed, this trend will continue, due to increased demands and widespread use, driving equipment costs down to more affordable and marketable levels. Both GMC (General Motors Corporation), Chrysler and some Japanese companies have anticipated such GPS price reductions and are projecting to future markets by incorporating GPS positioning into their AVLN system designs. Concern also exists regarding GPS Selective Availability controlled by the U.S. Department of Defense which may come into affect upon completion of launching and constrain non-military users to accuracies on the order of 100 metres. Differential GPS can, however, suppress the affects of Selective Availability considerably to levels required by AVLN systems and will be necessary.

Unlike relative dead reckoning methods, GPS positioning is not subject to error accumulation over time passed or distance travelled. To compute a two dimensional position fix, however, users require direct line of sight to at least three transmitting satellites passing overhead (four satellites for 3-D fix). Thus, when travelling in and about natural or man-made obstructions such as mountains, tunnels, and skyscrapers, necessary signal reception will be interrupted prohibiting the use of GPS throughout such shadowed regions.

Combinations of absolute and relative positioning methods are obviously complimentary in nature. GMC, realizing the problematic shadowing effects intrinsic to positioning their system on land by GPS and LORAN-C, have included into their test and demonstration vehicle, dead reckoning using an odometer and a compass [Dork, 1986]. Krakiwsky et al. [1988] discuss the advantages of blending differential odometer dead reckoning, map matching, and GPS positioning techniques via a Kinematic filter and present covariance analyses of the filter's performance along a simulated test route in the N.W. quarter of the city of Calgary. By combining the complementary differential odometry, map matching, and GPS positioning techniques, a vehicle's position and location can be continuously maintained within the framework of a road transportation network at all times at the 20 - 60 metre level . While travelling in shadowed regions such as urban canyons where GPS is masked, dead reckoning updated by map matching will predominate, and when journeying on long stretches of road where map matched updates are scarce, GPS will provide the necessary updates to the system's dead reckoning component.

In conjunction with position and location, AVLN navigation options are featured in some presently developing systems (e.g. Phillips CARIN) while others do not provide route guidance advice at all - they are only concerned with position reporting and monitoring of vehicle positions superimposed on a digital map. Some systems, such as Etak's Navigator, cannot provide route guidance advice but will highlight destinations and direct line pointing arrows relative to the vehicle's location on a digital map, leaving route selection and street navigation entirely up to the user.

Harris et al. [1987] explain the use of Dijkstra's shortest path algorithm in the context of AVLN best route determination and investigate a divide-and-conquer strategy entitled the 'patch-quilt' concept. This involves the partitioning of digital road networks into polygons to better attack the problem of best route determination in large transportation networks. Another divide-and-conquer approach was proposed by Sugie et al. [1984] and implemented into CARGuide, an on-board system which provides route instructions only not integrated with vehicle position and location. To date, knowledge-based route guidance systems have also been investigated and design concepts are presented by Karimi and Krakiwsky [1988b] and by Karimi [in prep.].

AVLN databases may range from basic coordinate files used as screen backdrops for systems which merely superimpose vehicle position onto a digital map, to carefully planned network and relational structures which support more sophisticated system functions such as map matching, route determination, and route guidance, to knowledge-based concepts for coupled system approaches Karimi [in prep.]. Guzolek and Koch [1989] list database contents included in a real-time route planning system presently being developed by Navigation Technologies Corporation in California. AVLN database requirements have been discussed in detail by Neukirchner and Zechnall [1986] and White [1987] and digital map dependent functions of AVLN systems have been illustrated by Harris et al. [1988].

1.5 Research Objectives

The primary objective of this research is to design an on-road, stand-alone, prototype, AVLN system and research various functions within. Emphasis is placed on designing and field testing a kinematic position filter which integrates differential odometer dead reckoning, map matching, and GPS satellite positioning, extending out from the simulation study performed by Krakiwsky et al. [1988] at The University of Calgary. As well, foundations for a navigation component featuring optimal route determination and guidance tied to real-time vehicle location are laid out in conjunction with the system's design. Simulation software for this component is developed for proof of concept purposes. To anchor the system and satisfy the needs of the above system components, a digital map database is built for a test region of a manageable size containing the required information to support map matching, route determination, and route guidance investigations. An urban community just north of The University of Calgary campus containing a mixture of both residential and industrial districts is selected for the performance test site.

It was not the intent of this project to focus on any one particular application but rather to construct a generic prototype model, which, with future extension, could be customized to fit various application sectors and could be extended to operate in a dispatch (i.e. multi-vehicle) mode as well. In light of the above research objectives, it was realized that completion of this project would create a solid foundation upon which future AVLN research and development could be built. In honour of initial AVLN research developments in this field at The University of Calgary and the group's projection of wide scale usage of AVLN systems by the year 2000, the prototype system is aptly named AVLN 2000™.

2. SYSTEM DESIGN

2.1 System Architecture

The architectural design of the AVLN 2000™ prototype system is laid out in Figure 2.1. Two natural divisions of the system's architecture are partitioned out, namely, the system's hardware component and the system's software component. This specific design follows the initial and more generic designs contained in Krakiwsky et al. [1987a] and Karimi [in prep.].

Major hardware equipment includes: a laptop microcomputer; a differential odometer configuration; a GPS receiver with antenna; and, a 12 volt automobile battery. The software component designed contains three modules: a database module; a position and location module; and, a navigation and information module.

2.2 System Hardware

The test vehicle employed for prototype development was a front wheel drive, four door sedan, Chevrolet Caravelle. For the prototype vehicle, a customized mount was constructed to house all of the necessary hardware. The assembly rests between the driver and the passenger on the floor in front of the seat and beneath the dashboard. A photograph of the housing assembly is given in Figure 2.2.

Located in the centre of the console is a GRiDCase EXP laptop microcomputer. Portable and compact, this microcomputer acts as the system's central processing unit, interacting with all other hardware peripherals and storing all software and databases required. It features two internal expansion slots, an RS-232C serial interface port, a 3 1/2 inch floppy diskette drive, an

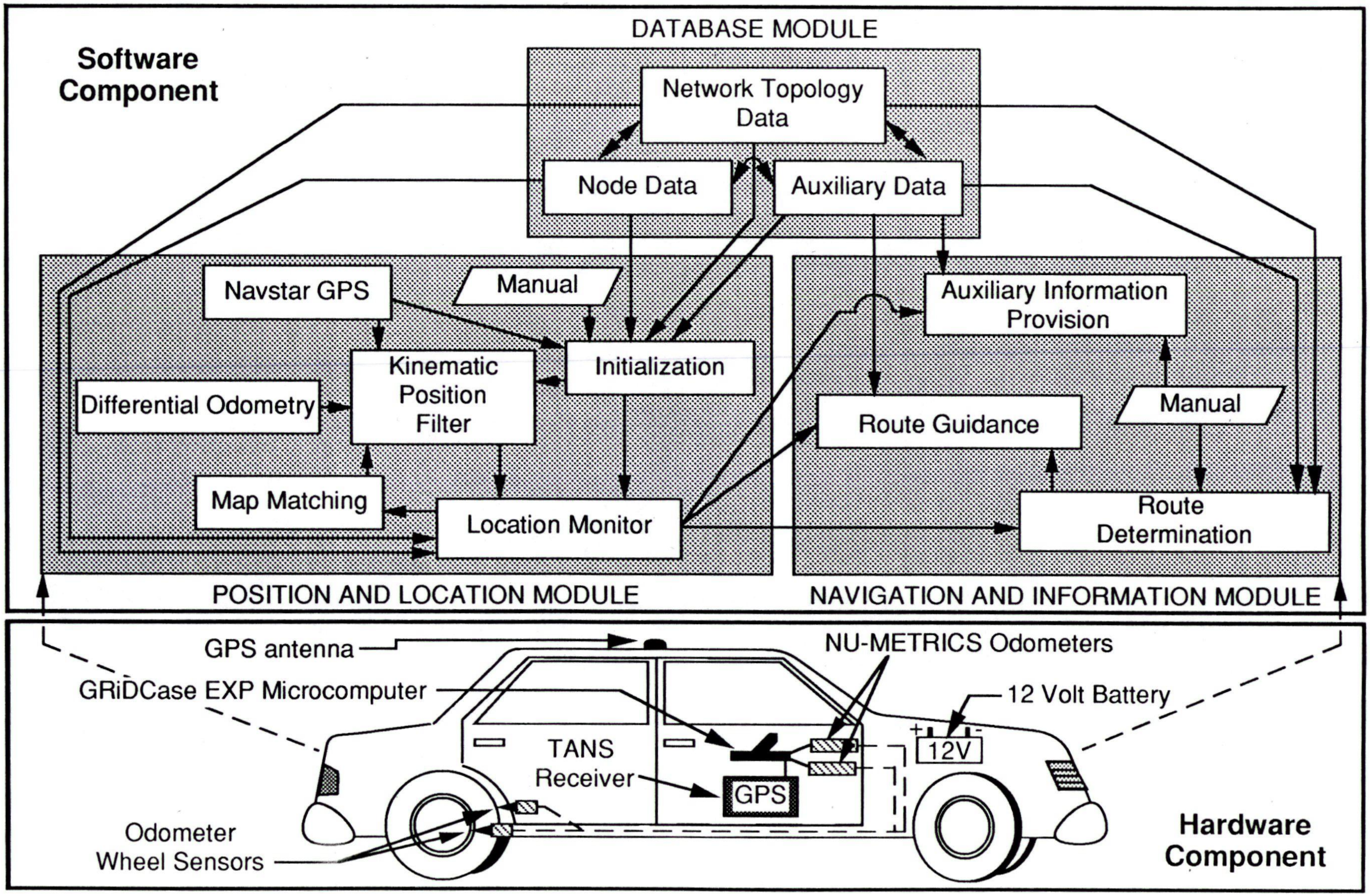


Figure 2.1 - Prototype System Architecture

Figure 2.2 - AVLN 2000™ Equipment and Housing Assembly

LCD (Liquid Crystal Display) screen, and is IBM compatible. The CPU (Central Processing Unit) of this GRiDCase computer is an Intel 8087 (i.e. Intel 8086 plus math coprocessor). One attractive feature of the GRiDCase EXP, in reference to this project, is its ability to be powered by 12 volt battery supply. For further information on the GRiDCase EXP refer to the GRiDCase EXP Owner's Guide [1987] and GRiDCase Technical Reference Manual [1985].

Above the GRiDCase EXP microcomputer in Figure 2.2, situated side by side, at the top of the assembly are two NU-METRICS Roadstar 40A Distance Measuring Instruments (DMI's) (i.e. odometers). As each instrument only measures distances travelled by one wheel only, two are required for differential odometry. To avoid problems related to slippage of power driven wheels due to varying surface types (e.g. ice, gravel), the non-driven wheel pair of the test vehicle (in this case the rear wheels) are utilized. As the vehicle moves the wheels rotate and eight, equally spaced, metal reflector targets attached to the inside rim of each wheel pass by an inductive proximity sensor one at a time. When a target is in close proximity to a wheel sensor, an electrical pulse is generated which is mathematically converted to represent distance travelled. Initial calibration is required on a level baseline of known length. The manufacturer's quoted relative accuracy of this device is one metre in one kilometre (i.e. 1/1000) and the smallest resolution attainable using the NU-METRICS micro-processor output is 0.304 metres (i.e. one foot). Compensation for signal errors generated by the sensory equipment in response to various vehicle dynamics is provided for by the AEC (Automatic Error Correction) software feature of the NU-METRICS instrument; an internal raw data preprocessing facility. The primary function of the AEC feature is to detect missing pulses by observing and predicting velocity and time and further to compensate for these missed signals by generating synthetic pulses. Unfortunately, at very low, creeping speeds the AEC function may not work appropriately due to potential, non-uniform pulse patterns, and in such instances, as a result of slow transition periods of targets passing by the proximity sensor, pulses can be physically missed. As each odometer stands alone, PDI (Periodic Distance Interval) pulses are generated from each device and fed into a 32 bit, two channel configured, up-down counter, microcomputer add-on board (CONTEC - CNT16-4M) inserted in one of the two GRiDCase EXP expansion slots. This configuration accommodates the interface required between the odometers and the microcomputer allowing differential odometry to be performed. Power to both odometers is supplied by the automobile's 12 volt battery. A more enhanced description of the NU-METRICS Roadstar DMI's is found in the NU-METRICS Roadstar Distance Measuring Instruments Operating Manual [1987]. As well, greater detail of the CNT16-4M add-on board is seen in the CONTEC IBM-PC/XT/AT Add-on Board CNT16-4M (PC) 16-Bit 4-Channel Up/Down Counter User's Manual [1986].

To the side of the assembly housing on the right, is a TANS (Trimble Advanced Navigation Sensor) GPS satellite receiver. It is normally mounted on the side of the equipment stand under the right odometer flange but is laid out here for better inspection. The Tans receiver is a C/A code, dual channel sequencing receiver which operates on L1 frequency and can track up to six satellites in view. Physically it is compact (127 millimetres (mm) x 207 mm x 51 mm) and light weight (1.27 kilograms (kg.)), quite appropriate for in-car applications. Position and velocity vector updates with accuracies of 25 metres SEP (Spherical Error Probable) and 0.2 metres per second RMS (Root Mean Square) error respectively are available approximately once every second but no less, and when requested are transmitted to the GRiDCase EXP via an RS-422 serial connector and an RS-422/RS-232C adaptor. Placed on top of the housing assembly is a TANS GPS antenna. It, like the receiver, is quite small for present GPS hardware markets (dimensions - 96 mm x 102 mm x 14 mm) and is lightweight weighing only 0.17 kg. Mounting of the antenna to the roof of the automobile is accomplished by a Trimble built, magnetic GPS antenna platform. As was the case for the NU-METRICS EXP microcomputer configuration, and the NU-METRICS odometers, the automobile's 12 volt battery supplies the required power to all TANS equipment. Further reference to the TANS GPS equipment is directed towards the TANS System Definition and Program User's Guide [1988] and the TANS GPS Receiver Specification and User's Manual [1988].

2.3 System Software

Three major modules combine to form the AVLN 2000™ prototype system software component: a database module; a position and location module; and, a navigation and information module. As seen in Figure 2.1, the software is represented by a system block diagram. Housed within each module are numerous blocks. Data blocks are found in the database module and function blocks are contained in the position-location and navigation-information modules. Beyond the function blocks, two manual inputs boxes, one in each of the latter two modules, are also included to illustrate the necessary human interface component in the design. Flow arrows indicate the dependencies between functions and data. A similar but much more general diagram depicting digital map dependent functions for AVLN systems is illustrated by Harris et al. [1988].

Contained within the database module are three primary sources of data: node data (e.g. x, y - UTM (Universal Transverse Mercator) map coordinates); network topology data (e.g. adjacent nodes defining arcs, one-ways); and, auxiliary data (e.g. street names, parking lots). Combined, these three sets of data provide the necessary information about a digital street network which accommodates the various system functions housed within the other two modules. Data blocks are shown interconnected in Figure 2.1 and are related to one another by node identifier numbers.

Encased within the position and location module are six system functions.: initialization, GPS positioning, differential odometry, map matching, kinematic position filtering, and location monitoring. Each is identified by a rectangular block in Figure 2.1.

The role of the initialization function is to provide a starting point for both the kinematic position filter and the location monitor. To accomplish this knowledge of vehicle position and heading are required. This starting point information can conceivably be provided by the GPS component which automatically provides position (ϕ - latitude, λ- longitude, h - height) and azimuth derived from two consecutive GPS fixes while the vehicle is in motion. When GPS is not available due to shadowing effects or outages, manual initialization is required. In order to initialize, all three data types are required: node map coordinates; network topology; and, auxiliary data, the latter being required for manual street name intersection initialization only.

The Navstar GPS positioning function not only provides information for the initialization function to perform automatically but also provides positional updates to the kinematic position filter. The differential odometry function provides distance and distance difference (i.e. yielding change in vehicle heading) measurements of both the left and right rear wheels to the the kinematic position filter. The kinematic position filter is the heart of the system's positioning feature. It blends differential odometry, map matching, and GPS function information to provide optimal estimates of three states which are corrections to these three parameters: two UTM coordinates (x, y), vehicle forward heading azimuth (α). The filter requires state initialization and feeds optimal estimates of position and heading to the location monitor.

The location monitor is initialized via the initialization function and receives information from the kinematic position filter function (i.e. coordinates, heading, distance travelled) to continually determine and monitor the location of the vehicle while it manoeuvres within a transportation network. Location is maintained by comparing and correlating vehicle movement to the digital street map. In order to carry this out, knowledge of the street topology and associated node coordinates are required and are obtained from the database module. When the monitor is confident that a definite match to the street map can be made, whether it be an intersection turn or street azimuth, a map match update may be made and fed into the filter. As one can see, a data flow loop exists between the filter and the location monitor. The filter feeds the changing vehicle states to the location monitor and in turn the location monitor provides periodic updates by matches to the map (i.e. map matching).

Map matching is isolated from location monitoring in Figure 2.1 to show that route guidance functions in the navigation and information module must receive location information much more frequently than map matching rates (e.g every intersection turn) in order to perform properly. Also this separation is to illustrate that the location monitor may locate the vehicle in the network but may not necessarily be able to provide an update to the kinematic position filter. For example, it may identify the automobile's location as being in a parking lot but this knowledge of location does not possess enough information to provide the filter a reliable map matched update (i.e. x, y, α). The location monitor is a crucial function to the success of any AVLN system as it is this function which reaches across to all functions of the navigation and information module, bridging the necessary common ground to automatically relate vehicle position to surrounding environment (i.e. street network) represented digitally by the contents of the database module.

Housed within the navigation and information module design are three defined functions: a route determination function; a route guidance function; and, an auxiliary information provision facility. Route determination involves computation of a best route to follow from the vehicle's present location (supplied by the location monitor) through the digital street network (described by the network topology data) and to a given destination (manually selected by street address and automatically converted to node identifiers using geocoded auxiliary data). Once a best route has been computed, the route guidance function uses the node to node defined list along with vehicle location and heading to instruct the driver on how to direct the course of the vehicle by means of intersection negotiation advice in real-time so as to follow the chosen optimum route. As approaching intersections are designated in the real world by street names, the auxiliary information contained in the database provides street names of approaching street crossings to the route guidance function to allow additional information to be given outside of just turning instructions alone (e.g. turn left at 32 Avenue). Auxiliary information provision is designated as a separate function to exemplify the system's future capability of providing digital map, auxiliary information to a driver or navigator outside of tasks related to optimal routing and even possibly vehicle location. To operate, it draws necessary information from the auxiliary data files and will involve human interaction.

Software programs used for development and testing are written in the C programming language, specifically Turbo - C for microcomputer environments. The next four chapters are devoted to each module of the designed prototype's software component, namely: the database module (Chapter 3); the position and location module (Chapter 4); the kinematic position filter (Chapter 5); and, the navigation and information module (Chapter 6).

3. DIGITAL MAP DATABASE MODULE

3.1 Graph Theory

Graph theory is used extensively in AVLN development to characterize street networks. This section describes the generic components and characteristics of graphs and networks pertaining to AVLN systems. Reference to Smith [1982] is made for a more detailed discussion on a broad range of general graph theory concepts.

One major part of a network is a graph. A graph (**n**, **a**) consists of a set of nodes (**n**), sometimes called points or vertices, and a set of arcs (**a**), often refered to as links. There are 'n' nodes in the set (**n**) and 'a' arcs in the set (**a**). Arcs represent connections between node pairs and are sometimes referred to as links. One arc contains one start node (a tail node) and one end node (a head node). The standard convention of flow for an arc(*i*, *j*) is that the start node is node *i* and the end node is node *j*, with the direction of the arc flowing from *i* to *j* as seen in Figure 3.1. Node *i* and node *j* are said to be adjacent to one another.

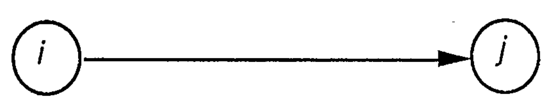


Figure 3.1 - Arc Flow Convention

For one arc there only exists one node pair, but one node pair may define more than one arc. If two or more arcs have the same node pair they are termed parallel arcs. It is possible for the arc set **a** to be an empty set (i.e. **a** = {ø}), resulting in a graph of individual, unconnected nodes. The node set **n** in a graph may but does not necessarily contain nodes which are all linked together. A graph in which all the nodes are linked together is called a connected graph. When unlinked nodes or separated node-arc subsets exist in the graph, then the graph is classified as a disconnected graph. It is assumed no such graphs exists in street networks, however, they should be addressed here as partitioning of larger street networks into smaller ones can result in disconnectivity within the smaller, newly defined regions.

Graphs which only contain uni-directional arcs (i.e. one way streets) are called directed graphs or digraphs for short, and those which only consist of bi-directional arcs (i.e. two way travel flow) are called undirected graphs. If a graph contains both directed and undirected arcs it is referred to as a mixed graph which real-world transportation networks are categorized under. A small mixed graph is illustrated in Figure 3.2 and typifies a set of connected streets. Directed arcs are indicated with directional flow arrows where as undirected arcs have no arrows at all.

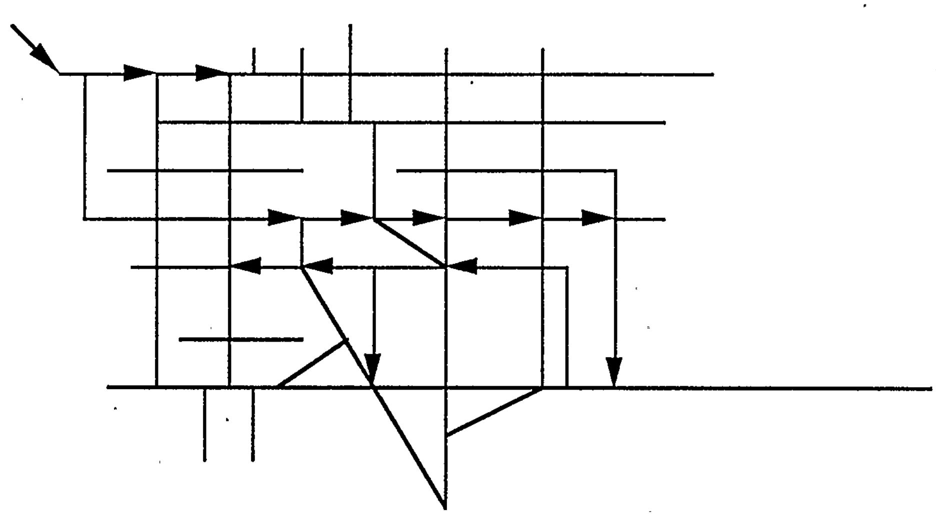


Figure 3.2 - Example of a Mixed Graph

Arc sets **a** are commonly used in graph theory applications and are discussed in Smith [1982]. Four such sets are chains, paths, cycles, and circuits. A chain is a set of arcs which in series join a source node (s) and a terminating node (t). A chain thus contains a set of intermediate nodes between s and t. These arcs may flow in either direction from s to t, or from t to s, and do not all have to flow in the same direction. A path is a special type of chain in which the chain's arcs all flow in the direction from s to t. A cycle is also a special type of chain whose start and end nodes are one of the same. A circuit is both a path and a cycle combined. When there exists paths and chains which contain no cycles in them they are called simple paths and simple chains respectively.

In many instances, a graph is a portion or subset of a larger graph. For example a community contains a set of streets which in turn is a section or subset of a larger set of city streets. Subsets of graphs can be divided into two categories; partial graphs and sub-graphs. A partial graph (**n**, **a**') is a graph extracted from a larger graph (**n**, **a**) that contains the same set of nodes (**n**) as its parent graph but has a smaller number of arcs (**a**') in its arc set (**a**). A subgraph, on the other hand, is a graph whose nodes (**n**') are a subset of a larger graph's node set (**n**) and whose arc set (**a**') are all those arcs which contain the nodes found in (**n**'). There also may exist a combination of these known as a partial subgraph (**n**', **a**').

A tree is a special type of connected graph whose nodes and arcs are structured in a hierarchical fashion. Often trees are partial subgraphs (**n**', **a**') of larger graphs (**n**, **a**). A commonly used example of a tree is family genealogy (or family tree) but it is noted that this example may not always exemplify true hierarchy. At the source of this tree is the root node. Most nodes have a parent (ancestor) node and a child (descendant) or set of children nodes. Only the root node (R) has no ancestors. A node with no children is called a leaf. Trees are commonplace in optimal route determination theory and when properly applied, find powerful application in real-time map matching and as well route guidance.

A tree whose nodes possess values or labels are called labelled trees. These labels are separate from node numbers. Trees whose labels are ordered are called ordered trees. Trees may be structured in numerous ways and can be stored quite efficiently depending on the application. One common characteristic of trees is that one node can have no more than one parent node. Trees can be classified into many categories according to their structure. For example, a partially ordered tree is a tree in which no child has a node label which is greater than its parent's label.

Consider a specific tree locally defined by approaching a node along a link from another node; case in point, a car approaching an intersection. If travel is constrained to the graph then only a limited number of possibilities or opportunities exist to proceed onward. By branching outward or expanding the potential links of travel stemming out of the approaching expansion node (as defined by the graph's topology), a first-order tree (by depth) in this local region may be formed. Such a tree will be refered to here as a successor tree of opportunity and consists of a root node R, an expansion node E, and a set of successor nodes $\boldsymbol{S}$. An example of the entire configuration of such a successor tree is illustrated in Figure 3.3. When one approaches node 2 (the expansion node E) from node 1 (the root node R), the opportunities of travel emanating out of node 2 (E) are towards the successor nodes (or leafs) 3, 4, and 5, which are the successor nodes S_{left}, $S_{forward}$, and S_{right} respectively, defining end points of alternatives links of travel. This particular example excludes the notion of u-turn manoeuvres which may be handled by assigning R as a successor node S_{back} as well. All nodes including R, S_{left}, $S_{forward}$ and S_{right} in this example are adjacent nodes to node E.

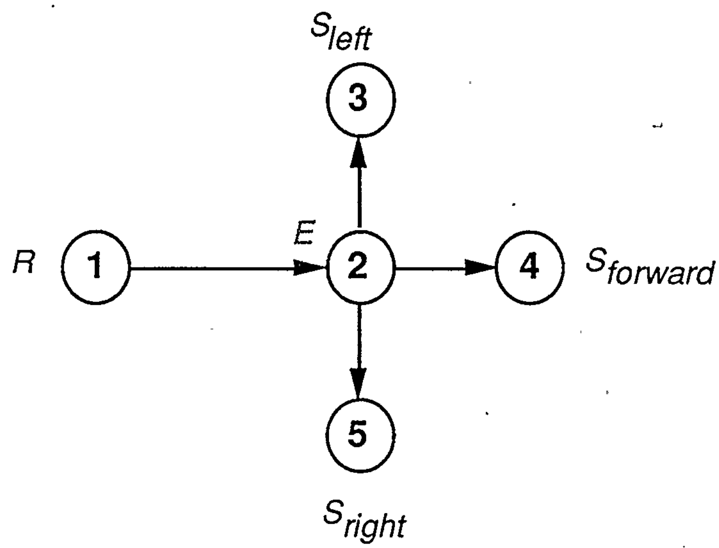


Figure 3.3 - Successor Tree of Opportunity

Networks (**n**, **a**, **p**) are simply graphs (**n**, **a**) whose arcs have additional properties (**p**) other than the arc's defined start and end nodes. A good example of an additional arc property is the length of the arc. It should be noted that in order to have a network there must exist at least one such additional arc property pertaining to each arc in the graph. Such street segment attributes or properties could include link distance, maximum permitted velocity, street name, road surface type, scenic value, etc.

The number of arcs in a network can be quite important in certain graph theory applications. A complete network is a network in which every node in the network is connected to every other node in that network. For a complete network,

$$a = n(n-1) \cong n^2 , \tag{3.1}$$

and for a near complete network,

$$a \cong n(n-1) \cong n^2. \tag{3.2}$$

On the other hand, when

$$a << n(n-1) \cong n^2, \tag{3.3}$$

the network is identified as a sparse network which is most characteristic of automobile transportation networks.

Networks are not always in two dimensions. In fact, networks may take on any dimension. For instance, imagine a network that possesses a time dimension. That is to say that some property (or properties) changes with time. A static network is a network whose defined properties are independent of time. These type of networks are also called non-time dependent networks and involve properties such as link distances. A dynamic network is opposite to a static network in that certain properties of the network vary with time. For example, impedance values relating to traffic flow can dramatically change with time of day, being quite 'resistive' during peak rush hour times and quite 'small' in the middle of the night. Dynamic networks are also termed time dependent or temporal.

Two other types of networks that exist are probabilistic networks and deterministic networks. Probabilistic networks are networks whose properties are not known with absolute certainty. These properties usually follow some form of probability distribution function. Deterministic networks, as opposed to probabilistic networks, are networks whose properties are known with absolute certainty.

Networks are usually visualized in a schematic form. This form of representation is good for imagining the structure of a network, however, for computational purposes, graphical diagrams, such as the one seen in Figure 3.2, have their shortcomings. Network diagrams, for computational purposes, are commonly represented in a numerical sense. This form of representation involves the creation of a relational structure relating the nodes, arcs, and arc properties in a meaningful and useful format.

There are three general structures often used in numerical network representation and they all have advantages and disadvantages. The three structures are the node-arc incidence matrix, the adjacency matrix, and the adjacency list.

The first method of representing the network numerically is the node-arc incidence matrix (**NODE_ARC**). This matrix is structured in the following manner. Each row of the matrix represents one node in the network and each column represents one arc. Thus the dimensions of **NODE_ARC** is n rows by α columns. In each column there contains all zero elements except those rows that correspond to the start and end node of that particular arc. To represent the directional flow of the arc, any arbitrary convention may be adopted. For example, undirected arcs may store the arc property value in both node pair rows and directed arcs may store a *-1* value in the start node row and the arc property value in the end node row. Two advantages of this form of network representation is that parallel arcs can be stored easily and the arc orientated structure of the matrix may be beneficial in certain applications. A strong disadvantage is that for networks with large node sets (**n**), many zeros may be stored in the incidence matrix, which may warrant an undesired matrix dominated by zero elements. This is definitely the case for automobile networks and thus the incident matrix is not recommended for AVLN implementation.

The adjacency matrix (**AMAT**) is an n by n dimensional matrix whose rows (*i*) and columns (*j*) correspond directly to the various nodes in the network. Similarly, the elements of **AMAT** (i.e. **AMAT**[*i*, *j*]) correspond to the arc property **p**[*i*, *j*]. If a node *i* is connected to another node *j* by one arc, then the property of that arc is stored in the location **AMAT**[*i*, *j*]. On the other hand, if node *i* is not directly linked to node *j* by one arc, then an infinite value ∞ (i.e. computationally a very large number) is stored in the location **AMAT**[*i*, *j*]. The advantage of using an adjacency matrix lies in the characteristic of the matrix itself. Only the arc properties need be stored numerically. The structure of the matrix inherently accounts for the adjacent node relationships of each arc. For complete or near complete networks, this is an efficient way to represent the network. Drawbacks can occur when the network contains parallel arcs. A single adjacency matrix can't contain two or more undirected arc properties for the same node pair unless another dimension to the array are introduce. In this case we now have an adjacency array. This may pose a storage waste problem, especially when there are only a small number of parallel arcs compared to n^2.

The third possibility of street network representation is the adjacency list array **ADJ_LIST**. This list is a table of arc node pairs (i, j) and corresponding arc properties $\mathbf{p}[i, j]$. The main advantage to this method is storage efficiency when working with sparse networks (i.e. automobile road transportation networks). If, for example, there are m additional arc properties besides the node pair for each arc in a network, then the decision threshold when deciding on an adjacency list **ADJ_LIST** over an adjacency matrix **AMAT** is,

$$a\,(2 + m) < m\,n^2\,, \tag{3.4}$$

where

$a\,(2 + m)$ is proportional to **ADJ_LIST** storage,

$m\,n^2$ is proportional to **AMAT** storage.

So if (by rearranging equation 3.4)

$$a < \frac{m\,n^2}{(2 + m)}\,, \tag{3.5}$$

then one should select an adjacency list for storage optimization. Depending upon the application various methods can be introduced regarding the order of storage of adjacent nodes in an adjacency list. One method of storing an adjacency list is called the forward star form. This method uses a pointer list, an end node list, and an arc weight list and requires storage proportional to n and a, namely:

$$\text{storage} \therefore n + 2a, \tag{3.6}$$

[Dial et al., 1979]. As the arc weights are sorted by ascending order, it is termed forward star form. It recommended that a similar method of network storage to this be used for AVLN applications, as was the case for this project.

3.2 Prototype Test Network

The transportation region used for prototype development is illustrated in Figure 3.4. The selected area is located just north of The University of Calgary's (U of C) campus and is a blend of both residential and industrial districts. The network itself is composed of 118 nodes and 179 arcs and internally features numerous street characteristics such as: one ways; cul-de-sacs; T-intersections; off ramps; high speed arterials; collector and secondary roads; school and playground zones; and, driving loops, providing a realistic yet manageable size network for the test project. One polygon is also connected to the network and it represents a parking lot located on U of C grounds. As both one way and two way streets (i.e. arcs) exist in the network its structure is a mixed graph. Specific nodes relating to discussions in the following sections are represented by bold dots in Figure 3.4.

Most vehicles operate in much larger networks geographically (e.g. those found on municipal, regional, provincial and nation wide scales). Due to the large storage requirements imposed by these larger networks and the real-time processing demanded by AVLN systems, it is envisioned that divide and conquer strategies which partition the network into subnetworks will prevail in the future.

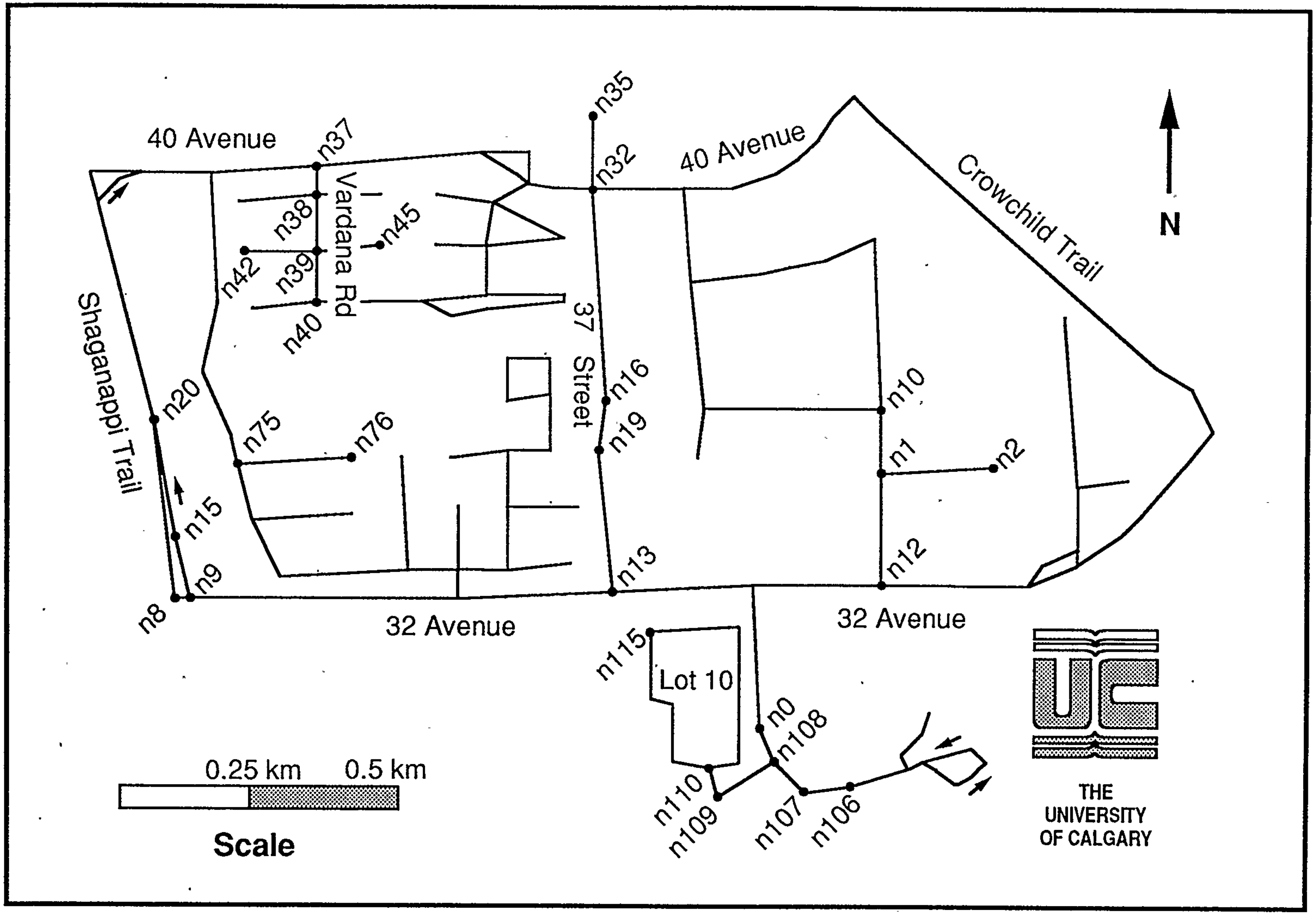


Figure 3.4 - Prototype Test Network

Network partitioning implies larger networks being represented by a mosaic of smaller subnetworks demarcated by polygons, providing more rapid and efficient real-time system operation and performance with manageably sized working frames to operate within. The test area selected is representative of a small subnetwork (i.e. a community) within a larger network (i.e. city of Calgary).

Extraction of street intersection coordinates from the Area Master Files (AMF) provided by Statistics Canada supplied initial digital map information. Customization and enhancements were carried out to more appropriately suit the needs of the project (e.g. topological structure reformatting). For the geographic regions where AMF data coverage does not extend (e.g. U of C campus) and for those street intersections where AMF coordinates are in gross error, a small GPS campaign both supplemented and corrected all the updates required.

Vector (i.e. street segments) as opposed to raster (i.e. bit map) format forms the foundation of the test network's database. AVLN network operations such as map matching to streets, route determination, and route guidance are more appropriately performed for on road AVLN operations with vector formats than with raster formats. Vector formats provide a storage efficient method of naturally representing a graph or network digitally. As AVLN operations are mostly concerned with sparse street networks, vector files save tremendous amounts of storage space when compared to raster (i.e. pixel by pixel) data files. Raster map representation is best envisioned as a matrix storage technique filling in all picture elements of a computer screen and stored this way for future display. Unfortunately, all topological data and coordinate data that was required to build the bit map of streets is buried in the dot matrix. As most route determination algorithms require vector information describing a network's topology numerically, complex and run-time expensive algorithms would be required to extract the necessary vector knowledge back out of such matrices enabling route determination algorithms to operate. This large incompatibility with route determination algorithms prevents the practical implementation of raster techniques alone into AVLN systems offering optimal route guidance options. Compared to raster, vector format is also favourable in light of map matching functions as map correlation to streets can be made much more quickly and more efficiently with vector based structures than with raster.

Comparatively speaking, vector format is clearly the prefered selection for most AVLN system applications. Raster format does have one major advantage over vector techniques. Implementing raster based data permits significantly faster displaying of digital map images to a computer screen than vector data does [Klesh, 1989]. This is due to the fact that vector techniques involve segment by segment line drawing operations compared to a single screen graphics dump of a raster image. As time is a key factor in any real-time system, some combination of both (e.g. highlighted vector best route overlayed onto bit map background) may be a novel solution for future AVLN generation systems which contain both map display and route guidance options. As it was not the intent of this project to develop enhanced AVLN graphics functions, only vector format is implemented for functional reasons discussed above. For a more in depth analysis on this subject see Karimi [in prep.].

3.3. Node Data

Each node in a network can be represented by a unique node identifier. Many graph theory related problems are not concerned with a node's geographic representation (i.e. physical coordinates) but only of its relation to other nodes in the network (i.e. topology). In an AVLN environment, however, this is not the case and node coordinates play an essential role providing the relational medium in which vehicle location within the network may occur. Without such information, no physical relationship between computed geographic positions of the vehicle and locations contained within the digital map (of which all information is tied to) could be made. For the test project, a street node data file was created of the test network seen already in Figure 3.4.

Illustrated in Figure 3.5a is the node data record structure. It is of fixed length and contains three fields of attributes. The first two fields of each node record are reserved for the x and y UTM coordinates of that node. The last field is designated to contain a code typifier number which indicates node type. To expand on this last field, every node used to describe a street map digitally may not pivot a typical cross (+) intersection. For example, some nodes anchor T intersections whereas others lie within a sequence of segments representing road curvature. Knowledge of node types are used for route guidance operations, aid in map matching techniques, and assist reading variable length topological records.

x UTM Coordinate	y UTM Coordinate	Node typifier code

Figure 3.5a - Node Data Record Structure

An example of the node file is displayed in Figure 3.5b. The file can be envisioned as an array with each row containing one node record corresponding to a particular node in the network and each column corresponding to a particular field. Node identification is carried out by assigning each node a node identification number which directly corresponds to the row of the array in which that particular node is described. Note that this is not the same as the node typifier code. The number of nodes in the array is coincident with the number of nodes contained in the network. For future work involving very large networks it is realized now that it may be more appropriate to add a new field in the node record reserved specifically for a node identifier number. Such an addition would allow for easier maintenance and upkeep of the database. As mentioned above, node identifier numbers in this file correspond directly to the listed order of each node record in the node file. Each field is now addressed here in further detail beginning with coordinates and followed by node typifier codes.

Numerous coordinate types can conceivably describe a node's position. Differences exist between varying combinations of coordinate systems and reference frames (e.g. ECEF - Earth Centred Earth Fixed Cartesian X,Y,Z; geodetic curvilinear φ, λ, h; and, x,y - UTM mapping plane). The acquired Canadian AMF coordinate data, exist only in reference to the UTM, two dimensional mapping plane which is based upon the geodetic NAD27 (North American Datum of 1927); the standard generally used in Canadian GIS industry. UTM coordinates used for the test network data conform directly to the AMF data available. Working in this system allows positioning, locating, and map matching to be performed in a two dimensional Cartesian world, making computations quite straight forward; an appealing characteristic to a real-time, dynamic system. Another favourable quality regards visual presentation. The two dimensional reference frame conforms directly to two dimensional computer display screens. Two problems arise, however, when UTM mapping coordinates referenced to NAD27 are incorporated.

Node Identifier	x UTM	y UTM	Type
node 0 (row 0)	700579	5662329	0
node 1 (row 1)	700780	5662759	3
	I	I	I
node 8 (row 8)	699540	5662540	2
	I	I	I
node 39 (row 39)	699795	5663127	4
	I	I	I
node 76 (row 76)	699855	5662784	1
	I	I	I
node 110 (row 110)	700490	5662263	5
	I	I	I
node 115 (row 115)	700387	5662486	6
	I	I	I

Figure 3.5b - Node File Example

The first is more of a consideration than a problem. Any UTM projection, regardless of the geodetic datum used, involves a two dimensional map projection based upon six degree longitudinal zones. This zonal restriction ensures that map projection distortions are controlled. Unfortunately, when one crosses between two zones, a discontinuity exists in coordinates as the new zone is referenced to another central meridian. As the test region selected is quite small relative to six degrees of longitude, it falls entirely within one UTM six degree zone and no problems of this type were of concern to the project. One must be aware, however, of this non-global trait for future development where UTM zone boundary crossings will be inevitable.

The second problem is more serious in nature and involves the discrepancy between the nongeocentric, geodetic NAD27 (North American Datum - 1927) which many existing digital map data are referenced to, and the geocentric, geodetic NAD83 (equivalent to WGS84 - World Geodetic System 1984) in which GPS coordinates are derived. The UTM projection is a standardized transverse Mercator projection [Frankich, 1985], which, like most map projections, approximates a portion of the earth's ellipsoidal surface in two dimensions. The coordinates of the projection subsequently depend upon the curvilinear coordinates of latitude and longitude (defining points on the earth's surface) which in turn depend upon how the earth's spheroid is defined [Vanicek and Krakiwsky, 1986]. The difference between NAD27 and NAD83 datums is quite significant and it is not sufficient to simply convert NAD83 GPS latitude and longitude coordinates to the UTM mapping plane and relate these to NAD27 derived UTM street coordinates. The difference in Alberta is expected to be in the 100 metre range for geographic coordinates and in the 200 metre range for UTM coordinates [Walker and Usher, 1988]. As GPS is an absolute positioning system referenced to NAD83 and is being incorporated into the prototype, appropriate transformations must be made.

Unfortunately, absolute point conversions based on geometric coordinate system transformations [Krakiwsky and Wells, 1971] between NAD83 to NAD27 do not account for regional non-linear distortions caused by the independent datum parameter adjustments. Three readjustment possibilities exist: rigorous recalculation technique using original field observations which is very costly; common correction factor throughout a locality which is boundary limited; and, rubber sheeting which by comparison between specific points computes non-linear correction parameters for all other points from NAD27 to NAD83. Preliminary rubber sheet testing has shown very good results for such conversions [Walker and Usher, 1988] and will be more economical than the rigorous recalculation efforts and more flexible than boundary limited readjustment. The test network (Figure 3.4) selected lies within a small enough region that a common correction factor can be applied converting directly NAD27, UTM coordinates to NAD83, UTM coordinates. The shift parameters are a -76 metre correction to x(NAD27) and a 222 metre correction to y(NAD27); showing no detectable discrepancies between GPS and map once applied. These parameters were computed using NAD27 and NAD83 geodetic coordinates of four different control stations situated around and about the prototype test network and were found constant to within 0.3 metres across the region defined; clearly within the tolerance needed for AVLN. The control information was supplied by the Geodetic Survey Division of the Department of Energy, Mines and Resources, Canada.

The last field in the node data record is the node typifier code. Seven node typifier code numbers, 0 through 6, are implemented to typify the nodes forming the test network and define variable record lengths of the network topology data. These node types are exemplified in column 3 of Figure 3.5b in conjunction with the digital map plotted in Figure 3.4. A 0 code indicates a node which is used to describe curvilinear features of a street bend (e.g. node 0). The code number 1 describes a node which is a cul-de-sac end point (e.g. node 76). A code 2 expresses a node anchoring a near right angle turn intersection (e.g. node 8) differing by a 0 code in that a significant identifiable change in turn direction can be used for coordinate update map matching. Codes 3 and 4 are attached to T and + intersection nodes respectively such as node 1 and node 39. Codes 5 and code 6 are reserved for parking lot nodes with the distinction of the former specifying entrance-exit gateway nodes (e.g. node 110) and the latter boundary nodes only(e.g. node 115). Further node types will undoubtedly be required for unique network configurations not contained in this test network.

It is realized that many node attributes not included in the prototype's database would further enhance later generation systems but have yet to be included in many GIS street databases. An obvious node feature missing is height information. This would accommodate for slope reduction of observed odometer distances onto the two dimensional mapping plane and also aid in magnetic compassed based systems allowing a means to better estimate the compass plane's incidence with the earth's magnetic field. Unfortunately, height information is not contained within the AMF sources acquired and unfortunately does not exist in developing AVLN databases to date. It is recommended to AVLN map developers to digitize this third dimension into developing databases so as to provide information to better resolve a vehicle's position and location (e.g. three dimensional map matching) and lend additional knowledge to route determination and guidance functions housed within navigation and information modules.

An example of position enhancement from height information is a GPS position fix using only three satellites by constraining the height (retrieved from the map database) in the solution. Other node attributes envisioned to assist significantly pertain to traffic control, such as traffic lights, stop signs, no turn restrictions, and yield signs. This information will provide additional impedance values for best route computation. Landmark information in the near proximity of particular nodes will also contribute to route guidance functions by alerting drivers more appropriately of important 'land marked' intersections ahead. For each new attribute classification created, each node record (see Figure 3.5a) would be expanded by one field.

3.4 Network Topology Data

Representation of the topology of a street network is fundamental to the AVLN prototype system [Krakiwsky et al., 1987b]. This is due to the fact that the map matching, location monitoring, route determination, and route guidance functions are all dependent upon knowledge of the topological structure of the road network.

Topology of a network is the connectivity relations between nodes contained within the network. Each node is connected to one or more nodes. For example, consider a standard grid-like framework often associated to an urban street network. Each internal node is connected to four other adjacent nodes via arcs (i.e. streets) providing four potential routes for travel advancement (i.e successor tree of opportunity) from that point.

The record structure of the network topology data is exemplified in Figure 3.6a. The knowledge fields contained within each variable length topology data record are:

a) each nodes adjacent node relation;
b) the street name identifier number of each link defined; and,
c) the link distance between two adjacent nodes.

Variable lengths of each record are known from the node typifier code number contained in the node data file (section 3.3).

Adjacent node	Street name identifier	Distance	---	---	---

Figure 3.6a - Network Topology Data Record Structure

Similar to the node data file, each row (i.e. record) of the array structure, seen in the example topology file of Figure 3.6b, corresponds to one node. Contained within each row are the adjacent node identifier numbers of that particular node distinguished by node identifier numbers (i.e node file row numbers). Inserted after each adjacent node number, which completes the definition of an arc, are two arc attributes (i.e. $m = 2$). The first is the link's distance weight and the second is the street name identifier code of the link. Combined the three fields form a field set, one set for each case of adjacency. A specific example is seen in row 39 which corresponds to node 39. It is an + intersection node with four adjacent nodes: nodes 38, 45, 40 and 42. The arc defined by end node 39 and 40 has the street name identifier 18 (corresponding to Vardana Road, see Figure 3.4 and Figure 3.7b) and a distance of 89 metres. Conceivably any information type pertaining to each arc (e.g. legal permitted velocities, lane widths, etc.) can be stored just as arc lengths and arc street names are.

```
Node Identifier

node 1 (row 1)          | 10  33 103 |  2   6 201 | 12  33 189 |

node 8 (row 8)          | 20   9 297 |  9  32  35 |

node 9 (row 9)          |  8  32  35 | 15   9 113 | 14  32 467 |

node 15 (row 15)        | 32  36 352 |  9   9   0 | 13  32 267 |

node 39 (row 39)        | 38  18  90 | 45  26 113 | 40  18  89 | 42 12 127 |

node 76 (row 76)        | 75  27 203 |

node 107 (row 107)      | 106  0  56 | 108  0   6 |

node 110 (row 110)      | 109  4  52 |
```

Figure 3.6b - Network Topology Data File Example

Not all nodes are directly adjacent to four other nodes. This is apparent from inspection of Figure 3.6b and quite obvious when one thinks about it. For instance in Figure 3.4, cul-de-sac ending nodes (code 1) have only one adjacent node (e.g. node 76). Thus only one adjacent node (node 75) and corresponding set of arc descriptors are found in row 76 of the topology data file example (Figure 3.6b). Code 0 nodes (nodes in curvilinear sequence -shape nodes) and code 2 nodes (near right angle turn nodes) both have two adjacent nodes. Examples of these are nodes 107 and 8 respectively. T-intersections (code 3) have 3 adjacent nodes (e.g. node 1) and parking lot gateway nodes (code 5) (e.g. node 110) are unique in the sense that they are connected to at least one street node and to two parking lot boundary nodes (code 6). A separate file was created to handle the one parking lot in the test network (i.e.U of C parking lot 10). This file contains a stream of node numbers defining the parking lot polygon boundary including the gate way node 110, allowing a relation to exist between the street topology nodes and parking lot polygon.

The network topology file is also where one way street information is stored. If an arc is directed (i.e. allowable travel flow is one way only) then the length of that arc which represents the opposite flow (i.e. against the one-way) is flagged by a zero weight element (representing ∞ - converted in the program to a very large weight). The adopted orientation of flow is defined by the row node as the start node flowing to the corresponding adjacent node, the end node. An example is the off ramp arc defined by nodes 9 and 15 (see Figure 3.4). From node 9 to node 15 traffic may exit 32 Avenue and feed onto Shaganappi Trail. Traffic is prohibited to flow from node 15 to node 9. Thus, in the network topology file, the length of arc 9 to 15 found in row 9 (Figure 3.5) has the correct weight, 113 metres, however, its compliment length found in row 15 (arc 15 to 9) is assigned a zero weight (representing ∞ - converted in the program to a very large weight) indicating a one-way restriction.

It is clear that the topology date file contains the information of adjacency list described in section 3.1, the most appropriate selection for AVLN databases due to the inherent sparseness quality of automobile networks. For this particular network recall $n = 118$ and $a = 179$. Plugging these values into equation 3.5 yields the true inequality

$$179 < \frac{2 \cdot 118^2}{2 + 2} = 6962, \qquad (3.7)$$

which numerically proves the reason for the adjacency list selection from a storage point of view. It is similar to a forward star form, having $n + 2\alpha(m + 1)$ storage (2α representing one street segment stored twice, once for each flow direction possibility). The listing of adjacent nodes are not ordered in an ascending or descending fashion by arc attribute quantity, however, but rather are ordered by adjacency node orientation about the particular expansion node at hand. The order is defined by clockwise orientation in this particular case. For code 4 nodes (+ intersections) it does not matter which adjacent node leads the sequence stored in a row, but more importantly, that they are all stored in a clockwise fashion about the central node.

From an AVLN point of view, such order aids in location monitoring, map matching, and route guidance. If the system has identified what arc it is on and what node it is approaching on that arc, then from this data structure not only the successors nodes of the up and coming expansion node can be accessed quickly but furthermore which successor node is to the right, which is forward and which is to the left upon intersection approach. For code 2 and code 3 nodes, the lead adjacency node in the corresponding row is important to maintain in the orientation strategy allowing the left, forward and right successor nodes (whatever the case may be) to be easily identified. For code 2, the node which defines the top of the uppercase "L" of the intersection is listed first and clockwise sequence is maintained for all other adjacent nodes following. For code 3 nodes, the upper right most node of the uppercase "T" is listed first with all others from this point ordered in a clockwise sequence. Code 0 and code 1 need no orientation for the obvious reason that the successor node set $\boldsymbol{S}$ is empty for cul-de-sacs (code 1) (excluding the root node R) and only contains one node for code 0 nodes, leaving only one possibility of street travel neglecting mid street, u-turn movements. Without doubt, successors node orientation could be resolved by real-time azimuth computations and comparisons upon each intersection approach but this format allows for quick relative orientation without computation.

The topological adjacency structure is probably the most important component of the database module. Unfortunately it is also the single most factor missing in many existing digital map GIS road databases.

3.5 Auxiliary Information Data

Viewed from a user's perspective, auxiliary information contained within this file provides the necessary data link between numerically represented street networks and symbolic street names, block address endings, points of interest, etc., with which drivers and navigators identify with. The data file for the test system was a list of street names ordered alphabetically. Each row is reserved for one name of a road be it a street, avenue, parking lot entrance, etc., and the chain of nodes associated with that street. In this way, alpha-numeric street names are geocoded directly to node identifiers and further to chains of arcs.

An example of the prototype system auxiliary data record structure is given in Figure 3.7a and an example of the auxiliary file compliments this in Figure 3.7b. Relations can be made to the node attribute data file via node identification numbers and to the network topology data file by means of node identification numbers and street name identification numbers attached to each defined arc. These relations accommodate a two way access, namely: they allow the system to quickly access the street name for navigation presentation when driving on a particular street or approaching a specific intersection; and, opposite to this, they allow the system to rapidly retrieve a node identifier number when the navigator types in a location by alpha-numeric street address (in order for appropriate numeric ties to be made necessary for various system functions).

Street name	number of nodes (n)	n_1	n_2	---	n_n

Figure 3.7a - Auxiliary Data Record Structure

Street Name Identifier			
street 18 (row 18)	Vardana Road	4	37 38 39 40
street 39 (row 39)	37 Street	5	13 19 16 32 35

Figure 3.7b - Auxiliary Data File Example

It has been envisioned by Case [1986] that tremendous amounts of useful auxiliary information could be stored into future AVLN systems similar to what one might find in the yellow pages of a telephone directory and Lee et al. [1989] discuss various route related information types that will be useful to scores of future AVLN applications. These visions will soon become a reality, giving a tremendous amount of power and practicality to AVLN systems in years to come.

3.6 Remarks

Many developmental man-years still lie ahead in digital mapping before the practical implementation on AVLN systems into our society will occur. Development areas concerning: cartography for computerized land vehicle navigation [Glick and Johnson, 1984; Hayes, 1986] including automatic label placement [Monmonier, 1982; Ahn and Freeman, 1984]; zooming, panning, and scrolling operations for in-car digital map displays; database modelling and structuring in light of all system functions; real-time retrieval speeds, update flexibility, and storage limitations; geocoding and digitizing efforts involving hundreds of resources to represent the entire nation's streets and highways appropriately; update and maintenance of street networks transforming over time (i.e. dynamic networks); and, quality data checking are among the hurdles AVLN mapping presently faces.

If digital street networks stored in cars are poorly represented and incomplete, whether it be by input error or more likely by neglected update, AVLN systems will be subject to failure while operating in affected regions. The more such instances exist, the less reliable the system will be. It is imperative that both these issues be addressed and taken very seriously with the utmost respect. This is true not only from a system integrity point of view but also from a consumer market point of view. Interestingly, even since the project began, the prototype network has physically undergone changes. What was once a traffic intersection controlled by stop lights at Crowchild Trail and 32 Avenue (see Figure 3.4) is now a major overpass and off ramp thoroughfare. Real world transportation transfigurations and transformations clearly must be dealt with.

In the years to come digital mapping will experience many advancements. Rising technologies such as CD-ROM (Compact Disc-Read Only Memory) storage devices capable of storing more than 500 megabytes of data [Cooke, 1985], enough to store a digital map of every highway and street in Canada and the United States [Case, 1986] and network transformation detection by high resolution satellite imagery techniques [Li et al., 1989], will dictate direction of the industry's approach to solve the pressing problems at hand. Considering the magnitude of the digitization and update processing which prevails, industry mapping standards must be defined as soon as possible. Scores of creative AVLN mapping ideas and methodologies remain to be unveiled and soon cartographers and mappers facing the future must make significant decisions (e.g. redundancy in exchange for speed) regarding AVLN related databases as they see their industry transform progressively from a map production industry to an information industry [Haywood, 1986].

The chapter presented here outlines what data structures were incorporated for the prototype development over the test region selected providing the necessary information to various, real-time system functions. The intent was to present the kinds of knowledge that must be embedded into AVLN system databases and describe the working database installed in the AVLN 2000™ prototype .

4. POSITION AND LOCATION MODULE

4.1 Initialization

To begin operating, an AVLN system must have some knowledge of its present starting position in order to furnish the kinematic position filter with an initial point of expansion and additionally to commence resolving the vehicle's location with respect to the network. The function which handles the provision of such knowledge is entitled the initialization function. Initialization for an AVLN system may be accomplished by a number of alternatives with three primary methods including:

1) manual input (e.g. menu selection);
2) automatic GPS position link fitting; or,
3) saved information carried over from the previous journey.

Manual input implies exactly that. Numerous methods of input are possible ranging from standard keyboard entry to dedicated keypad menu selection to touch sensitive screens displaying a digital map. By whatever means the information is entered, the decision being strictly ergonomical, it is clear that location not position information will be supplied in the most part. Users will have no idea nor care for that matter what their exact latitude or longitude (i.e. position) is, but they will, however, be able to supply knowledge of where they are situated in the transportation network by observing intersection street names read from corner signposts and addresses marked on nearby buildings (i.e. location). Given such location information, the system can then automatically position the vehicle by retrieving corresponding node and link identifiers from the auxiliary database and then in turn map coordinates from the node database. In this manner location by user identifiable street names are entered and numeric location and position is retrieved enabling the AVLN system to begin operations.

Although manual input provides a very reliable method of initialization, it falls short of perfection due to its tedious nature. Users would surely grow impatient rather quickly with a system requiring frequent manual updates. In this light, clearly one of the advantages of implementing GPS positioning into the system involves its innate ability to compute a position automatically without human involvement. Only position is provided however not location. The task to identify the corresponding network location given the position is the responsibility of the initialization function. Although not implemented into the AVLN 2000™ test software, a method of performing this operation worthy of mention was conceived during the project and is explained here.

Observation of two consecutive GPS fixes of a vehicle in motion defines a directed arc or link with two map coordinate end points x_1, y_1 and x_2, y_2.

By definition, the link is directed from point 1 to point 2 in the azimuth direction defined by the planar azimuth equation

$$\alpha_{12} = \tan^{-1}\left[\frac{x_2 - x_1}{y_2 - y_1}\right] , \tag{4.1a}$$

$$= \tan^{-1}\left[\frac{\Delta x_{12}}{\Delta y_{12}}\right] , \quad (4.1b)$$

where

$$\Delta x_{12} = x_2 - x_1 , \quad (4.2)$$

$$\Delta y_{12} = y_2 - y_1 . \quad (4.3)$$

Figure 4.1 illustrates the GPS computed link. Note that α_{12} is taken as the estimated vehicle heading in this case as well.

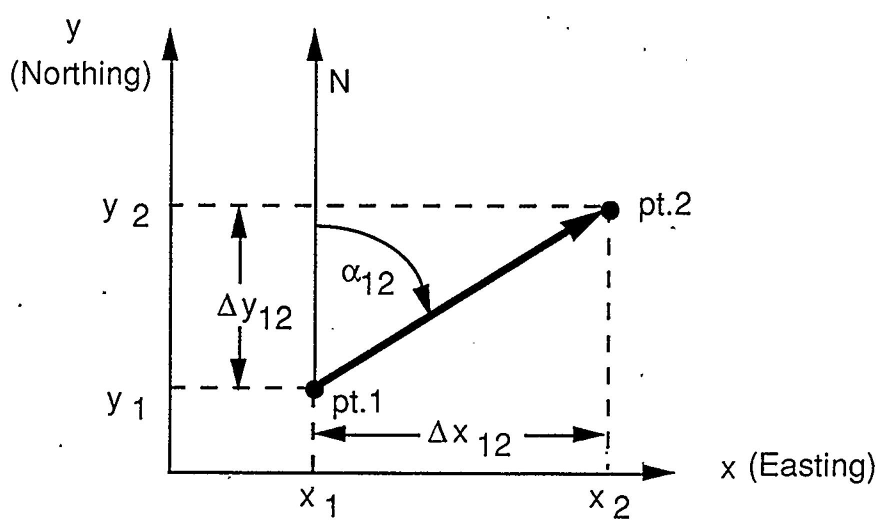


Figure 4.1 - Grid Azimuth Geometry

If the vehicle is travelling on a street, similarity exists between the observed GPS link and the street's link defined digitally in the map database. If a unique correlation can be found between a link in the database and the GPS link, then the location of the vehicle can be identified.

The correlation procedure to match links is termed 'link fitting'. To narrow the problem of link comparisons down tremendously, only street data contained in a prespecified window about the GPS coordinated link should be considered. Suitable window sizes should be considered based on existing discrepancies between observed position and digital map data.

An illustrated example of link fitting is displayed in Figure 4.2. Visually, it is clear link $\overline{3\ 5}$ would be the appropriate choice for a match. Notice that once the correct street has been identified, with the heading already determined (equation 4.1a), the next node en route (i.e. node 5) in the network can be quickly identified and all successor nodes of this node retrieved quickly from node topology data structure. Interpolation procedures using the last GPS fix (i.e. GPS 2) could then be implemented to estimate how far along the identified link the vehicle is situated and how far, in distance, is the approaching intersection from the vehicle's present location.

As it is inevitable that off road travel will occur whether it be through a field or in a parking lot (both defined as polygons in the digital map database) the term location must be defined as to include polygon occupancy (i.e. location is defined as being situated within the boundaries of a defined region). In these situations, location must be determined by solving a standard point-in-polygon problem. Techniques to solve the point in polygon problem are discussed by Feuchtwanger and Lodwick [1986] as applied in GIS.

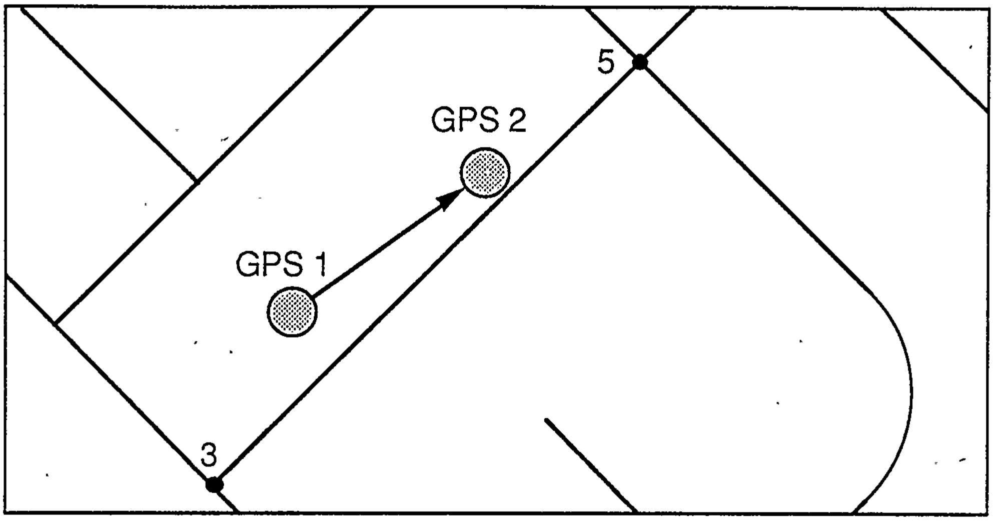


Figure 4.2 - GPS Link Fitting Concept

Clearly, an AVLN system which can provide the least amount of human interface during initialization or reinitialization periods will be the most favorable from a user's perspective. GPS obviously has the potential to automatically perform these duties as discussed above but due to shadowing problems leading to satellite signal blockage, manual reinitialization features should be designed into the system. This will ensure an alternative solution for reinitialization during location loss situations in GPS shadow zones.

Practically speaking, reinitialization should rarely occur. One clever approach to decrease reinitialization substantially is to utilize static RAM facilities enabling the system to remember all pertinent position and location information after power down just waiting to be used when the vehicle begins operating at some later time in the future. Of course, when initialization is required and it will be required from time to time, GPS can provide both position and location updates automatically reducing the option of manual initialization even further with the beauty of never letting the driver know he or she was temporarily lost in the first place. For the test project, no fancy initialization techniques were developed. Only manual input of system location defined by raw link and node identifier numbers was incorporated to allow for rapid field trials and development. A final product however would require extensions to this, such as the ones discussed above.

4.2 Differential Odometry

To obtain dead reckoned coordinates of a vehicle using odometers alone requires a minimum of two odometer sensors, each sensing independent wheels of the same wheel pair. By averaging and differencing the left and right wheel measurements over specified intervals, position vectors defining relative changes in vehicle position may be observed. To compute a set of coordinates (x_{i+1}, y_{i+1}) of a vehicle at time t_{i+1} both the previous set of coordinates (x_i, y_i) at time t_i and the position vector components $(\Delta x_{i,i+1}, \Delta y_{i,i+1})$ over the interval t_i to t_{i+1} are required. The general mathematical expressions for this is simply

$$x_{i+1} = x_i + \Delta x_{i,i+1}, \tag{4.4}$$

and

$$y_{i+1} = y_i + \Delta y_{i,i+1}. \tag{4.5}$$

To begin this recursive operation a set of initial starting coordinates and heading are required since the position vector implies knowledge of not only magnitude (i.e. distance) but direction (i.e. azimuth) as well. This gives rise in part to the initialization function described in Section 4.1.

Once a starting point and an initial heading azimuth have been defined, positional vectors in reference to these are computed by observing changes in both distance and azimuth. Over a small interval t_i to t_{i+1}, changes in distance and azimuth as functions of both left and right wheel odometer measurements are computed in the following manner. A change in left wheel distance is computed by

$$\Delta d^L_{i,i+1} = d^L_{i+1} - d^L_i, \tag{4.6}$$

and similarly for the right wheel by

$$\Delta d^R_{i,i+1} = d^R_{i+1} - d^R_i, \tag{4.7}$$

where

d^L_i distance measured at time t_i by left wheel odometer,

d^R_i distance measured at time t_i by right wheel odometer,

d^L_{i+1} distance measured at time t_{i+1} by left wheel odometer,

d^R_{i+1} distance measured at time t_{i+1} by right wheel odometer.

The estimated total distance travelled of the vehicle over the specified interval is the mean of equations 4.6 and 4.7 above defined by

$$\bar{\Delta d}_{i,i+1} = \frac{\Delta d^{L}_{i,i+1} + \Delta d^{R}_{i,i+1}}{2}, \tag{4.8}$$

where

$\bar{\Delta d}_{i,i+1}$ mean distance travelled over interval t_i to t_{i+1}.

A change in azimuth is also computed using odometer equations 4.6 and 4.7 along with the TRACK width of the vehicle's wheel path (from wheel centre to wheel centre of same wheel pair). Azimuth deflection as a function of left and right odometer observations is expressed as

$$\Delta \alpha_{i,i+1} = \frac{\Delta d^{L}_{i,i+1} - \Delta d^{R}_{i,i+1}}{\text{TRACK}}, \tag{4.9a}$$

$$= \alpha_{i+1} - \alpha_{i}, \tag{4.9b}$$

where

α_i Azimuth heading of vehicle's forward direction at epoch t_i,

α_{i+1} azimuth heading of vehicle's forward direction at epoch t_{i+1},

$\Delta \alpha_{i,i+1}$ change in heading of vehicle over interval t_i to t_{i+1}.

The equation for the change in azimuth $\Delta\alpha_{i,i+1}$ (equation 4.9) is shown to be true by viewing Figure 4.3 and differencing the following two arc length equations

$$\Delta d^{L}_{i,i+1} = R^{L} \Delta\alpha_{i,i+1}, \tag{4.10a}$$

$$\Delta d^{R}_{i,i+1} = R^{R} \Delta\alpha_{i,i+1}, \tag{4.10b}$$

where

R^L Radius of the left wheel's curvilinear path,

R^R Radius of the right wheel's curvilinear path.

Subtracting equation 4.10b from equation 4.10a gives

$$\Delta d^{L}_{i,i+1} - \Delta d^{R}_{i,i+1} = (R^{L} - R^{R})\, \Delta\alpha_{i,i+1}. \tag{4.11}$$

Isolating $\Delta\alpha_{i,i+1}$ in equation 4.11 yields

$$\Delta\alpha_{i,i+1} = \frac{\Delta d^{L}_{i,i+1} - \Delta d^{R}_{i,i+1}}{R^{L} - R^{R}}, \tag{4.12a}$$

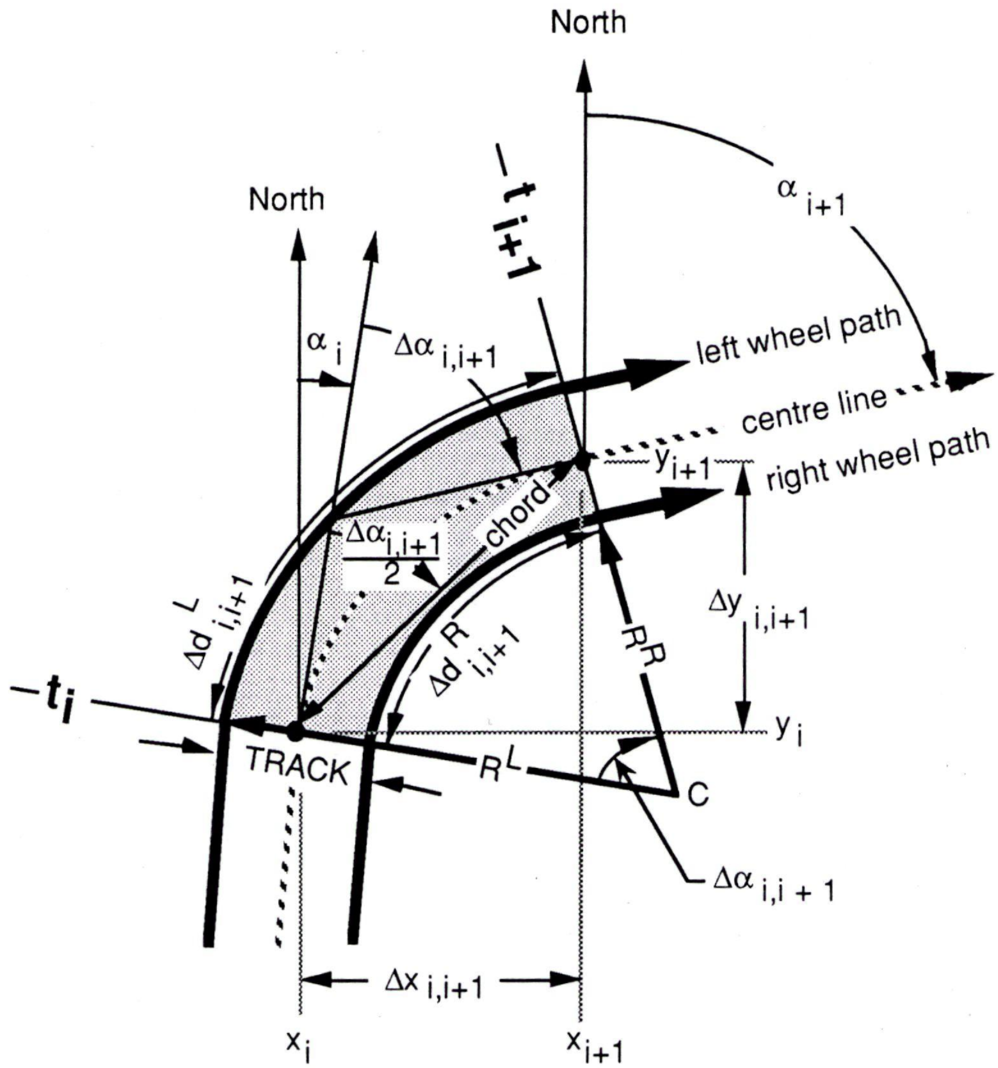


Figure 4.3 - Differential Odometer Geometry

which can be further expressed as

$$\Delta\alpha_{i,i+1} = \frac{\nabla \Delta d_{i,i+1}}{\text{TRACK}}, \tag{4.12b}$$

with the substitution

$$\nabla \Delta d_{i,i+1} = \Delta d^{L}_{i,i+1} - \Delta d^{R}_{i,i+1}, \tag{4.13}$$

and

$$TRACK = R^{L} - R^{R}. \tag{4.14}$$

For the prototype, the TRACK parameter is a constant since the employed rear wheels are fixed always remaining parallel to one another and forward pointing. For situations in which front wheels are employed this is not the case due to the vehicle's Ackerman steering control mechanism [Newton, et al., 1983; and, Hillier and Pittuck, 1975]. In this scenario, the TRACK is not a constant value but rather changes with the cosine of the angle of the wheel plane from its forward pointing direction (i.e. steering angle) [VNLST, 1986].

With knowledge of x_i, y_i $\Delta\bar{d}_{i,i+1}$, $\Delta\alpha_{i,i+1}$, and α_i, the parameters x_{i+1} and y_{i+1} from equations (4.4) and (4.5) may be calculated by solving the equations for change in coordinates which are

$$\Delta x_{i,i+1} = \Delta\bar{d}_{i,i+1} \sin\left[\alpha_i + \left(\frac{\Delta\alpha_{i,i+1}}{2}\right)\right], \tag{4.15}$$

and

$$\Delta y_{i,i+1} = \bar{\Delta d}_{i,i+1} \cos\left[\alpha_i + \left(\frac{\Delta\alpha_{i,i+1}}{2}\right)\right]. \tag{4.16}$$

The change in azimuth $\Delta\alpha_{i,i+1}$, is divided in half in equations 4.15 and 4.16 to approximate the chord of the arc path travelled (see Figure 4.3).

In order to compute the displacements $\Delta x_{i,i+1}$ and $\Delta y_{i,i+1}$ (equations 4.15 and 4.16) from interval to interval, the azimuth of the vehicle's heading must calculated every cycle, namely

$$\alpha_{i+1} = \alpha_i + \Delta\alpha_{i,i+1}. \tag{4.17}$$

Reduction of slope distances observed by the odometers to horizontal distance components are not carried out in the AVLN 2000™ prototype system as no tilt sensors were incorporated nor was there intersection height information available from the database. As the test region (Figure 3.4) contained no severe grades, this did not pose a problem. Future generation systems, however, should consider reducing observed distance data to the horizontal plane by multiplying the observed slope distance by the cosine of the tilt angle from the local horizon.

As well, no reverse wheel detection was automatically built into the system, however, the NU-METRICS odometers are capable of counting down by manual switching. It is suggested for automatic forward and backward motion detection that at least one additional sensor be placed on one of the wheels already odometer dedicated, mounted at 90 degrees out of phase with its partner odometer. With this configuration, the leading pulse coming from each sensor will reveal the direction of travel.

The accuracy of a computed set of coordinates x_{i+1}, y_{i+1} is obviously a function of the accuracies of $\Delta d^{L}_{i,i+1}$, $\Delta d^{R}_{i,i+1}$ and the previous coordinates x_i, y_i (see equation 4.1 to 4.17). The manufactured NU-METRICS odometer accuracies are $1 \cdot 10^{-1}$ of the distance travelled. Accuracies of x_{i+1} and y_{i+1} are also dependent on the interval length t_i to t_{i+1} and quantization errors due to the linear approximation involved.

The prototype retrieves odometer data at a rate of approximately 10 Hertz. This allows Δd quantities to be reasonably controlled at high velocities (i.e. 3 metres per interval at 100 km h^{-1}).

Quantization error unfortunately for the NU-METRICS case is more problematic. Only 0.235 metres (0.771 feet) intervals of distance travelled can be detected; this number reflects the ground distance travelled between two pulses observed on the test vehicle by the ROADSTAR 40A calibration procedures. In distance this does not pose a problem, but in azimuth change ($\Delta\alpha$) it is much more expensive, especially if a pulse is missed and goes undetected. Such instances can occur at slow creeping speeds which are common during intersection turns while waiting for traffic. By partially differentiating equation 4.12, the maximum $\Delta\alpha$ quantization error can be algebraically derived and an estimate computed.

Neglecting any error in the TRACK parameter, this partial differential equation is

$$\delta(\nabla\Delta\alpha_{i,i+1}) = \frac{1}{(\mathrm{TRACK})}\,\delta(\nabla\Delta d_{i,i+1}). \tag{4.18}$$

The TRACK width for the Chevrolet Caravelle was measured to be 1.43 m. For one missed pulse, a quantization error in distance difference (i.e. $\delta(\nabla\Delta d_{i,i+1})$ = 0.235 m) corresponds directly to a quantization error in $\Delta\alpha$ of

$$\delta(\nabla\Delta d_{i,i+1}) = \frac{1}{1.43\ \mathrm{m}}\ (0.235) \tag{4.19a}$$

$$= 0.16434 \text{ radians} \tag{4.19b}$$

$$= 9.4 \text{ degrees.} \tag{4.19c}$$

Clearly, a missed pulse by either NU-METRICS odometer employed can be an expensive contributor to error in coordinates resulting from error in azimuth of 9.4 degrees per missed pulse.

Estimates for changes in latitude and longitude ($\Delta\phi$, $\Delta\lambda$) as functions of changes in UTM coordinates (Δx, Δy) for future AVLN generations requiring latitude and longitude determination, may be computed according to the following equations which are functions of equations 4.15 and 4.16 above. A small change in latitude may be estimated by [Vanicek and Krakiwsky ,1986]

$$\Delta\phi = \frac{\Delta y}{R_e}, \tag{4.20}$$

and for a change in longitude

$$\Delta\lambda = \frac{\Delta x}{R_e \cos \phi}, \tag{4.21}$$

where

R_e is the radius of the earth.

For an ellipsoidal earth, R_e is not constant and the above two equations are enhanced to

$$\Delta\phi = \frac{\Delta y}{M + h}, \tag{4.22}$$

$$\Delta\lambda = \frac{\Delta x}{(N + h) \cos \phi}, \tag{4.23}$$

where

h is the ellipsoidal height of the vehicle,

M is the meridian plane radius of curvature,

$$M = \frac{a_e (1 - e_e^2)}{(1 - e_e^2 \sin^2 \phi)^{3/2}}, \tag{4.24}$$

a_e is the semi-major axis of the earth defining ellipsoid,

e_e is the earth's eccentricity,

$$e_e = \frac{a_e^2 - b_e^2}{a_e^2}, \tag{4.25}$$

N is the transverse radius of curvature (prime vertical section radius of curvature: perpendicular to meridian plane),

$$N = \frac{a_e}{(1 - e_e^2 \sin^2\phi)^{1/2}} . \tag{4.26}$$

The proof is made by utilizing the popular arc length equation

$$s_a = r \cdot \theta_a, \tag{4.27}$$

where

s_a is the arc length,

r is the radius of the arc, and

θ_a is the angle swept,

in application to the geometry of the earth's ellipsoid (see Krakiwsky [1973]). The ellipsoidal height h should be included, if available, in equations 4.22 and 4.23, to increase accuracy even more.

4.3 Location Monitor and Map Matching

The vehicle location monitor is essential to any AVLN system and with it tasks outside of vehicle position monitoring are performed. As mentioned previously, location not only relates to the knowledge of the vehicle's position but further to the vehicle's situated position relative to the digital street map. To ensure correct functional operation of an AVLN system, location must be consistently and continually monitored and maintained whether the vehicle is situated on a street, in a shopping mall parking lot, or off road in a field. Without this function performing well, the system would be lost, cutting all ties between vehicle and map database. The navigation and information module would have no inter-relation with the position and location module and thus prevent automatic location related navigation and information functions to perform (e.g. real-time route guidance instructions). Further map matching could not be carried out for positional updates without knowledge of vehicle location.

Map matching, as it is defined here, is a method of providing position and azimuth updates to the system by observing existing correlations between the observed path driven by the vehicle and the digital road network. It differs slightly from the location monitor in that this function is only invoked periodically, only when the location monitor has, to a certain degree of confidence, significant position and heading knowledge (i.e. azimuth) of the vehicle which both may be used as updates to the kinematic position filter. Map matching is truly a sub-function of the location module as its performance relies entirely on the monitor but is distinguished separately to highlight its significant interaction with the position filter.

Map matching origins, approaches and applications have been investigated in detail and presented by French [1989]. Such techniques have evolved over the last two decades and correlation matching ranges from bit map pattern recognition to vector based, window constrained probabilistic decision making. Presented here is the location monitored map matching technique built into the AVLN 2000™. A very straight forward map matching approach was taken in order to supply update information for testing of the kinematic position filter. Matches to the map are made by recognizing driven turn patterns at identifiable map defined intersections. This is performed in real-time, requiring network constrained, location monitoring to be carried out. The employed methodology is described here in conjunction with Figure 4.4.

At any time during a vehicle's journey, knowledge of the vehicles location and heading on a link in the network is known, maintained after initialization by the location monitor. The location monitor works strictly in a local sense, being concerned only with the particular link presently occupied by the vehicle. Distances travelled along the occupied link and direction of travel are observed. As the vehicle is approaching the end node's position of the current link, a successor tree (see section 3.1) is formed, involving all potential routes of advancement branching out of the up and coming intersection. The successor tree may be built quickly by examining all adjacent nodes contained in the topology record file corresponding to the approaching intersection's node. The tree's orientation is resolved by identification of that adjacent node in the topological record which corresponds to the root node of the successor tree being formed (i.e. the start node defining link of present travel). All remaining adjacent nodes thus define links of potential travel, all with unique azimuths. By composing and storing the successor tree well in advance of the next intersection en route, the system is capable of making pattern matches of observed vehicle path to the digital street map. When a match is made position and azimuth may be updated, and the location of the vehicle is maintained through knowledge of the newly identified link. This procedure is carried out progressively, with a new successor tree developed at every street segment the vehicle's path is matched to.

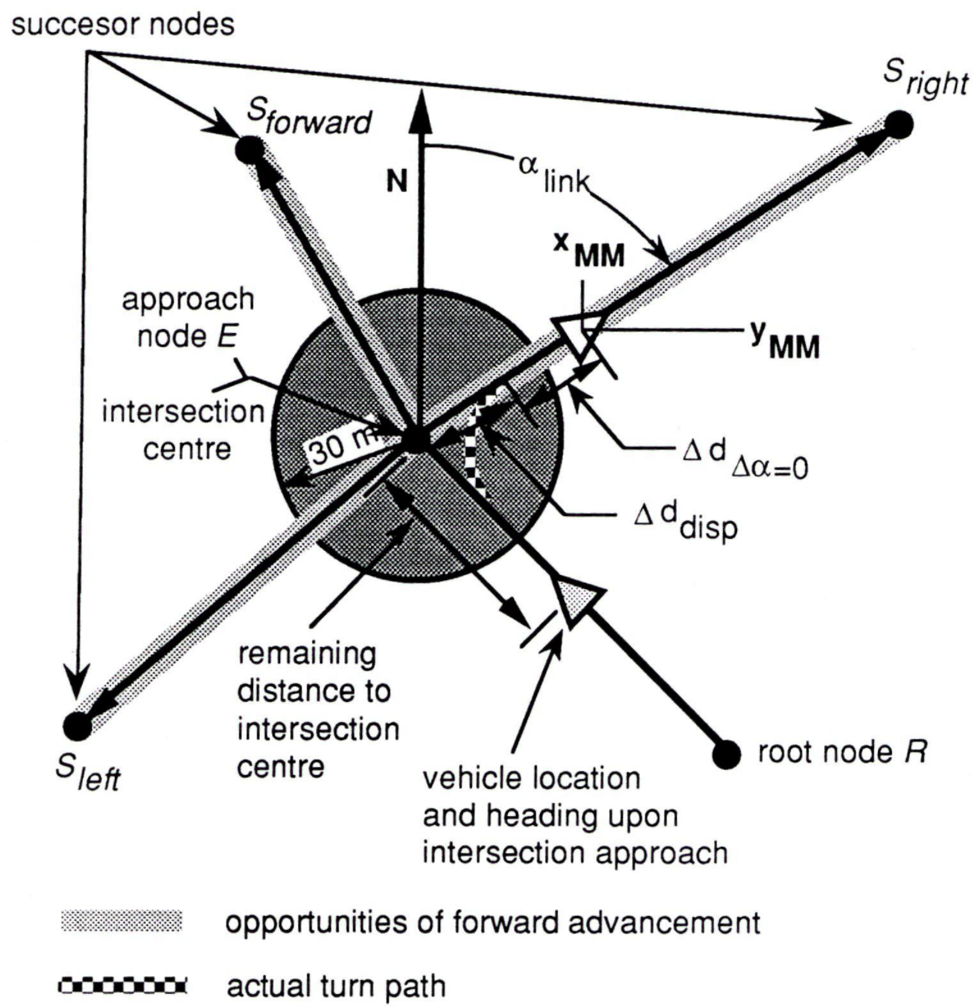


Figure 4.4 - Intersection Turn Map Matching

Prior to intersection proximity the system determines all opportunities of advancement. When the vehicle comes within a particular distance from the centre of the approaching intersection, the map matching function is alerted. A circular window with radius 30 metres is used as a measure of proximity. This accounts for any discrepancies and errors existing between the computed position and the digital map.

If the vehicle is in close proximity to an intersection (i.e. lies with the circular region defined) and an azimuth change is sensed which closely resembles one of the potential routes of the extension branches of the successor tree (e.g. within +/- 30 degrees), a match is made. It is noted that the update occurs after the turn has been made. To ensure that a correct azimuth update occurs, information from the differential odometer configuration is extracted and observed over time. Through the process of making a regular turn at an intersection, the trail of an automobile's path first follows along a straight path (i.e. on road approaching the intersection), then transforms into an arc (i.e. intersection turn), and returns to a straight line again (i.e. on new road). This particular pattern can be traced numerically by observing the quantity $\nabla\Delta d$ (equation 4.13) during the entire process.

Upon straight line approach, $\nabla\Delta d \cong 0$, then during the turn $\nabla\Delta d \neq 0$, and upon turn completion $\nabla\Delta d \cong 0$ again. When proceeding through an intersection or node and no turn is made, $\nabla\Delta d \cong 0$ through the entire proximity zone. If the vehicle is located in the proximity zone and indeed makes a turn, a map match is made on the new link of travel. Updates do not occur until straight line motion is detected by observing $\nabla\Delta d \cong 0$ over a period of time while the vehicle is moving. This ensures to a certain degree that the turn is complete and that the azimuth of the vehicle corresponds generally to the azimuth of the newly matched link (i.e. the vehicle movement is relatively parallel to the map link).

Map matched coordinates x_{MM} and y_{MM} (see Figure 4.4) are not the intersection centre coordinates defined in the database but rather fall a certain distance along the newly occupied street. This distance is a combination of two lengths. As the curve to tangent point of the vehicle's driven path through a turn does not generally pass through the centre of the intersection precisely, a nominal translational displacement (Δd_{disp}) of 20 metres for right turns and 25 metres for left turns is used in the computation of these map matched coordinates. Also, as distance ($\Delta d_{\Delta\alpha=0}$) is traversed for the period of time that the system is observing straight line motion (i.e. $\nabla \Delta d \cong 0$ which is equivalent to $\Delta\alpha \cong 0$) to ensure the turn has been completed (as discussed above), it is included in the determination of x_{MM} and y_{MM}. The mathematical expressions to compute the map matched updates are as follows:

$$x_{MM} = x_{Inter} + (\Delta d_{disp} + \Delta d_{\Delta\alpha=0}) \sin(\alpha_{link}), \tag{4.28}$$

$$y_{MM} = y_{Inter} + (\Delta d_{disp} + \Delta d_{\Delta\alpha=0}) \cos(\alpha_{link}), \tag{4.29}$$

$$\alpha_{MM} = \alpha_{link} , \tag{4.30}$$

where

x_{MM} is the map matched x coordinate,

y_{MM} is the map matched y coordinate,

α_{MM} is the map matched azimuth,

x_{Inter} is the digital map database retrieved intersection centre x coordinate,

y_{Inter} is the digital map database retrieved intersection centre y coordinate,

Δd_{disp} is the displaced distance from intersection resulting from turning,

$\Delta d_{\Delta\alpha=0}$ is the distance travelled while observing straight line motion.

Using this map matching procedure, intersection coordinate updates to within the 20 metre level (one sigma) were achieved. The extent of updates to the map included all intersection turns and 180 degree cul-de-sac turns.

The map matching technique described above integrated with dead reckoning performs reasonably well during real-time, field missions and served its purpose in gaining familiarity with AVLN map matching updates. It also provided map matched updates for the kinematic position filter tests which follow this chapter. Experience from numerous field runs revealed that this particular technique could brake down on off ramp shoots where the ramp angled away quite gradually and also in areas where intersections were closer than 30 metres apart, rendering the system lost. It is noted that as the system's operational area will be much larger than the development site selected for this project and that the location monitor must operate in real-time, a RAM buffered, sliding or switching window structure containing network data loaded from the main database should be developed for fast access and retrieval of surrounding data [Karimi, in prep.]. The prototype took advantage of fast RAM access but due to the size of the network, this was only done internally within the program by array pointers. In a sense, the test network itself is such a window from the prototype's point of view and adjacent windows should be loaded into the suggested buffer when ever the system crosses boundaries or sometime just prior to this. From field experience, development of more sophisticated and logic intensive location monitoring and map matching functions is a must for future system extensions to ensure system product integrity.

Further advanced map matching techniques are presently under investigation and development by both Etak [Honey et al., 1989] and Phillips [Thoone and Bruekers, 1987]; two systems which rely entirely on map matching to provide system updates. These efforts are based upon utilizing knowledge of estimated rectangular windows of position error encapsulating the vehicle's position. All digital map links which fall within the boundaries of the error window are potential candidates for street location. By means of simple decision logic and processes of elimination one particular road segment is recognized as the most probable possibility of location. Of course, the system will not always resolve just one location (i.e. an unambiguous situation). For example, if a left turn is made and the window of error contains two left turn possibilities then the system is unsure which street of the two it is truly on. In this case more sophisticated decision techniques are required to automatically re-locate the vehicle.

If time is not of the essence, the vehicle's driven path may be stored from this point in a historical fashion and correlated to the map some time into the future using all street segments in question as starting points in an effort to resolve a unique location. Perhaps similarities between driven velocity and speed limits, vehicle tilt and hill slope information, depending upon available information contained in the data base may aid in the decision process as well. The application of these techniques reduce the likelihood of required manual reinitialization and should be employed in any system, however, they don't eliminate the chances entirely. Obviously, the ultimate solution is to control error to a degree where no ambiguous location scenarios may arise (assuming high digital map quality) or at least not for very long, cutting the real-time location monitors tasks to a tracking mode only. This significantly reduces the load on the system's CPU giving it more time to run other system subprocesses.

The concept of using error window to assist map matching is one step further in complexity than the circular proximity technique used to identify intersection turns for the prototype . Room for enhancement in the design has been left for future developments towards this end. By employment of the kinematic filter, confidence regions (i.e. error windows) geometrically defined by error ellipses (in two dimensions) may be rigorously computed from the covariance information of the state vector.

4.4 NAVSTAR GPS

The NAVSTAR (NAVigation Satellite Timing And Ranging) Global Positioning System (GPS) is a 24 hour, worldwide, all weather absolute positioning system presently under development by the United States Department of Defense. When completed (estimated 1991), the entire GPS satellite constellation will contain 21 satellites plus 3 spares, all orbiting the earth in inclined orbits twice a day at a nominal altitude of 20 183 kilometres. The system is a passive multi-user system, transmitting two radio frequencies

L_1 on 1575.42 MHz and L_2 on 1227.6 MHz. Three main observation types exist namely: pseudo-range observations; carrier beat phase observations; and, continuous carrier phase [Wells et al., 1987]. Only the pseudo-range observables are utilized for prototype operation.

To solve for the 3-D coordinates of a vehicle equipped with a GPS receiver and antenna, simultaneous pseudo-range observations to at least four satellites must occur. The pseudo-range observation equation as given by [Wells et al. 1987] is

$$p = \rho + c(dt - dT) + d_{ion} + d_{trop}, \tag{4.31}$$

where

p	is the measured pseudo-range,
ρ	is the true geometrical distance between satellite and receiver,
c	is the vacuum speed of light,
dt	is the satellite clock error from GPS time,
dT	is the receiver clock error from GPS time,
d_{ion}	is the range error due to ionospheric delay,
d_{trop}	is the range error due to tropospheric delay.

The pseudo-range observable p is equivalent to the product

$$p = c \cdot \delta t\ , \tag{4.32}$$

where

δt is the difference between the time of signal transmission from the satellite (in the satellite time scale) and the time of signal reception by the receiver (in the receiver time scale).

The geometrical distance ρ from satellite to receiver is a function of both the receiver's and satellite's terrestrial coordinates and may be computed by employment of this three dimensional distance equation

$$\rho = \left[(X^s - X^r)^2 + (Y^s - Y^r)^2 + (Z^s - Z^r)^2 \right]^{1/2} . \tag{4.33}$$

Inserting equation 4.33 into 4.31 yields the parametric equation

$$\rho = \left[(X^s - X^r)^2 + (Y^s - Y^r)^2 + (Z^s - Z^r)^2 \right]^{1/2} + c(dt - dT) + d_{ion} + d_{trop} . \tag{4.34}$$

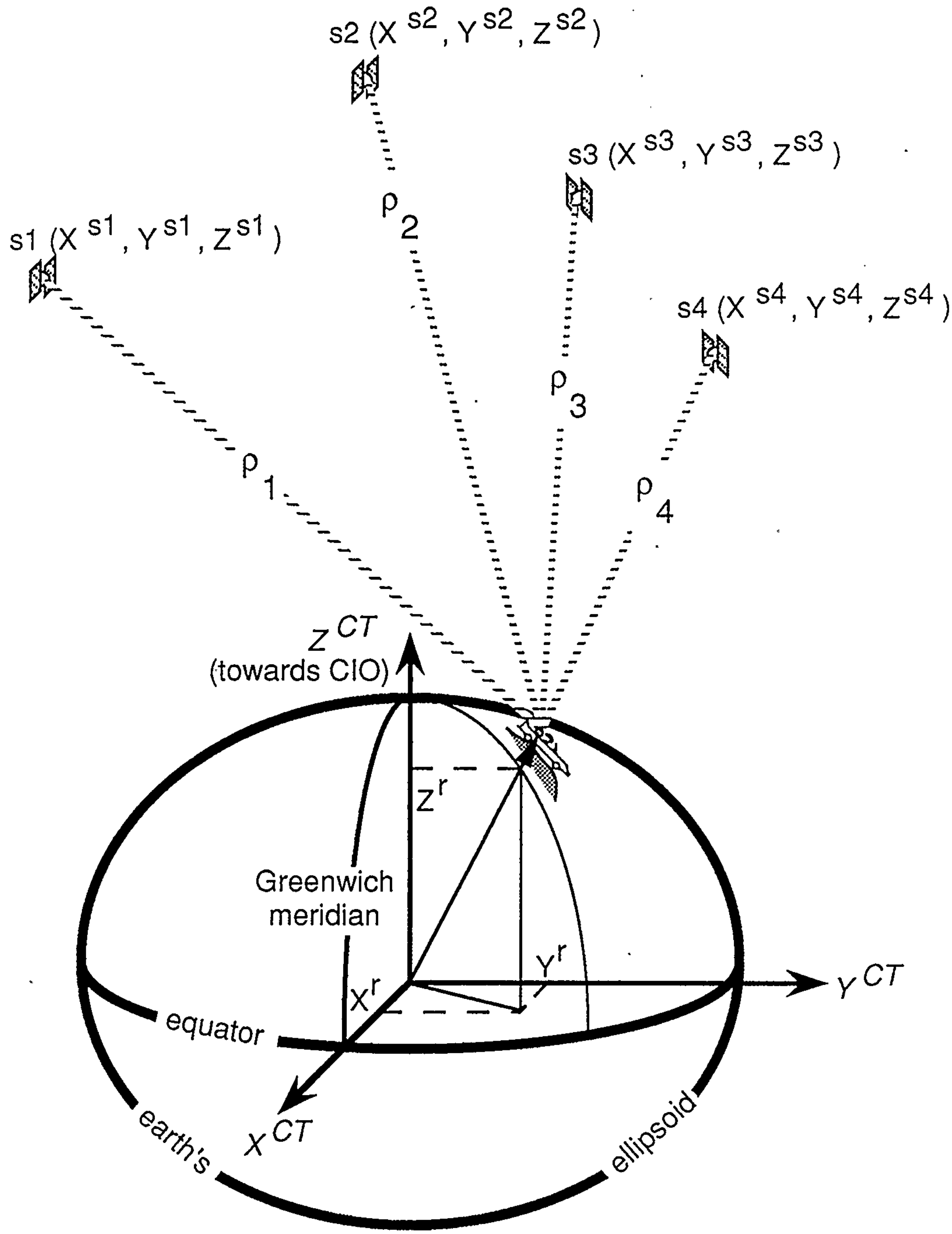
Four unknown parameters exist in this equation, these are: X^r, Y^r, Z^r (the coordinates of the receiver's antenna mounted on the roof of the vehicle) and dT (the receiver clock offset). All other parameters are known or observed. The satellite (s) coordinates X^s, Y^s, and Z^s are computed with knowledge of the Keplerian orbit parameters as obtained from the satellite transmitted ephemeris data, the solution of which is detailed by Van Dierendonck [1980]. The parameter dt is also transmitted from the satellite and the atmospheric delay corrections d_{ion} and d_{trop} are computed by the TANS receiver also using satellite transmitted information and standard atmospheric constants. The only remaining parameter is p the pseudo-range which is measured.

To uniquely solve for the four unknown parameters (X^r, Y^r, Z^r and nuisance parameter dT) requires the simultaneous solution of pseudo-range equations corresponding to observations made to four different satellites (e.g. s1, s2, s3, and s4). Figure 4.5 illustrates graphically the mathematics of the range intersection problem. The four range equations are:

$$p_1 = \left[(X^{s1} - X^r)^2 + (Y^{s1} - Y^r)^2 + (Z^{s1} - Z^r)^2\right]^{1/2} + c(dt^{s1} - dT) + d_{ion} + d_{trop}, \tag{4.35}$$

$$p_2 = \left[(X^{s2} - X^r)^2 + (Y^{s2} - Y^r)^2 + (Z^{s2} - Z^r)^2\right]^{1/2} + c(dt^{s2} - dT) + d_{ion} + d_{trop}, \tag{4.36}$$

$$p_3 = \left[(X^{s3} - X^r)^2 + (Y^{s3} - Y^r)^2 + (Z^{s3} - Z^r)^2\right]^{1/2} + c(dt^{s3} - dT) + d_{ion} + d_{trop}, \tag{4.37}$$



CIO (Convetional International Origin)

CT (Conventional Terrestrial)
- ECEF (Earth Centred Earth Fixed) Cartesian system

Figure 4.5 - GPS Pseudo-Range Intersection Geometry

$$p_4 = \left[(X^{s4} - X^r)^2 + (Y^{s4} - Y^r)^2 + (Z^{s4} - Z^r)^2 \right]^{1/2} + c(dt^{s4} - dT) + d_{ion} + d_{trop}. \qquad (4.38)$$

Once X^r, Y^r, and Z^r coordinates of the vehicle are resolved, latitude (ϕ), longitude (λ), and height (h) curvilinear coordinates may be computed [Vanicek and Krakiwsky, 1986; and, Heiskanen and Moritz, 1987]. Note that the GPS reference coordinate system conforms to the WGS84 datum which is equivalent to the NAD83 datum.

As the TANS GPS receiver used in the prototype computes ϕ, λ, and h independently from the AVLN system software every one to two seconds, these quantities are taken advantage of and are directly retrieved from the receiver and are used to compute UTM coordinates x and y of the vehicle. The following two sets of equations are employed to compute the parameters x and y and are applied in the kinematic environment. The mathematics to compute x as a function of ϕ and λ are:

$$x = f_x(\phi, \lambda), \qquad (4.39)$$

$$x = x_o + k_o \cdot (x_1 + x_2), \qquad (4.40)$$

where

$$x_1 = N \cdot \Delta\lambda \cdot \cos(\phi), \qquad (4.41)$$

$$x_2 = N \cdot \Delta\lambda^3 \cdot \frac{\cos^3(\phi)}{6} \cdot (1 - \tan^2(\phi) + \eta^2), \qquad (4.42)$$

k_o is the UTM scale factor which equals 0.9996,

x_o is the UTM false Easting which equals 500 000.0 metres,

$\Delta\lambda = \lambda_{vehicle} - \lambda_{CM}$,

λ_{CM} is the longitude of UTM zone Central Meridian - for test region $\lambda_{CM} = -(117/180) \cdot \pi$ radians (i.e. 117 degrees West),

$$\eta = \left\{ \left[\frac{a_e^2 - b_e^2}{b_e^2} \right] \cdot \cos^2(\phi) \right\}^{1/2} . \tag{4.43}$$

To compute the y coordinate, the following equations are used:

$$y = f_y(\phi, \lambda), \tag{4.44}$$

$$y = k_o \cdot (y_1 + y_2 + y_3), \tag{4.45}$$

where

$$y_1 = a_e \cdot [a_0\phi - a_2\sin(2\phi) + a_4\sin(4\phi)], \tag{4.46}$$

$$y_2 = N \cdot \frac{\Delta\lambda^2}{2} \sin(\phi)\cos(\phi), \tag{4.47}$$

$$y_3 = N \cdot \frac{\Delta\lambda^4}{24} \sin(\phi)\cos^3(\phi)\,[5 - \tan^2(\phi) + 9\eta^2 + 4\eta^4], \tag{4.48}$$

with

$$a_0 = 1 - \frac{e_e^2}{4} - 3\frac{e_e^4}{64} - 5\frac{e_e^6}{256} - 175\frac{e_e^8}{16384}, \tag{4.49}$$

$$a_2 = \frac{3}{8}\left(e_e^2 + \frac{e_e^4}{4} + \frac{15}{128}e_e^6 - \frac{455}{4096}e^8\right), \tag{4.50}$$

$$a_4 = \frac{15}{256}\left(e_e^4 + \frac{3}{4}e_e^6 - \frac{77}{128}e_e^8\right). \tag{4.51}$$

The above equations for x and y are truncated versions of a standard mathematical series which can be found in Thomas [1952] and Krakiwsky et al. [1977] and transform ϕ and λ into x and y (UTM) to the nearest metre.

Accuracy estimates of the TANS determined latitude, longitude and height are also available from the TANS receiver but only in the form of DOP (Dilution of Precision) quantities. DOP factors reflect the geometric component of the overall positioning accuracy σ_p. From Wells et al. [1987], a general equation for the standard deviation of the determined position as a function of both geometry (DOP) and measurement accuracy may be formed as

$$\sigma_p = \text{DOP} \cdot \sigma_m, \tag{4.52}$$

where

σ_p	is the positioning accuracy,
DOP	is the Dilution of Precision (geometric),
σ_m	is the measurement accuracy.

It is clear from this equation the DOP factor is the ratio of positioning accuracy to the measurement accuracy. Numerous varieties of DOP factors may be computed all of which are traces of diagonal elements combination of the position's variance-covariance matrix. As only two dimensional positioning was considered for the AVLN prototype, only the HDOP (Horizontal Dilution of Precision) Factor was retrieved from the TANS receiver. For each position fix performed by the TANS receiver an associated position accuracy estimate was computed by using equation with a nominal measurement accuracy value on pseudo-range observations of seven metres. This yields

$$\sigma_{position} = \text{HDOP} \times 7 \text{ metres.} \tag{4.53}$$

Unfortunately, the full positional covariance matrix is not provided by the TANS receiver but it is felt that for the application at hand, the above equation provides enough information for second moment computations to be carried out for AVLN applications.

One source of significant error that AVLN systems will be quite susceptible to due to frequent urban activity is caused by multipath affects. Multipath effects are caused by reflection of the satellite signals off of large obstacles such as building enabling signals to travel from satellite to receiver by more than one path. These multipath affects cause signal interference and confuse the receiver. If the wrong signal is used in the solution, it may affect the position estimate significantly. Fortunately, a method does exist to detect multi-path related errors in the AVLN 2000™. As x and y coordinates of the vehicle are computed independently from GPS by augmented dead reckoning, these may be converted to ECEF (Earth Centred - Earth Fixed) coordinates and used to compute rough estimates of geometric range to each satellite. Comparison of ranges, both observed and estimated, will allow for detection, in first approximation, of any gross observational blunders such as those associated with multipath affects. In such instances, these GPS solutions can be disregarded, or the ranges rejected.

5. KINEMATIC POSITION FILTER

5.1 Introduction

The primary role of the position and location module's kinematic position filter is to mathematically integrate differential odometer dead reckoning, map matching, and GPS positioning. Employment of a filter leads to optimal estimates of both the system's state parameters (i.e. statistical first moment) and the corresponding state covariance (i.e. second moment) information.

The term filtering falls under a broad classification entitled estimation theory, in which the terms prediction and smoothing also reside. If the instant at which an estimate is required coincides directly with the time at which the last update measurement was made, then the estimation procedure is designated as filtering [Schwarz, 1987]. If an estimate is required at a time before a measurement is made, this is termed prediction. If an estimate is required after the last measurement has been made, somewhere along the time line prior to the last measurement, then this form of estimation is termed smoothing. As an AVLN system is concerned with real-time estimation, smoothing plays no part in the estimation strategy, however, filtering is of utmost importance.

The beauty of incorporating a position filter for real-time applications is that it naturally and quite neatly carries forward all information up to and including the present time. This is all performed at no additional cost to storage due to the intrinsic recursive nature of the filter. In addition, it accommodates heterogeneous and non-uniform data coverage (i.e. measurements varying in type and content received at varying epochs not necessarily equally spaced in time).

It is stressed that the kinematic positioning filter described here does not determine vehicle location (as defined in Chapter 1) but rather computes best estimates of position related states and corresponding covariance matrix. This is a very desirable feature of the system as it allows the system's vehicle position to be maintained optimally regardless of where the vehicle is situated in the road network, whether it be on or off the road.

5.2 Integration Concept

Imagine a standard journey driven by a vehicle through a real-world, road transportation network (see Figure 5.1). Now envision you are given the task to maintain automatically the real-time position of this vehicle continuously and optimally as it manoeuvres through city streets and country highways at various times (t). The journey may very well begin at one's house. Initially, at time t_k^{INIT} the system requires a set of starting coordinates. This may be obtained, automatically, from a GPS position fix or recalled from memory of the last position estimate stored prior to system shut down, or manually, from user input. As the car sets out through surrounding urban communities position and heading estimates are calculated using differential odometer DR (Dead Reckoning). To control dead reckoned error accumulation both GPS and MM (Map Matching) provide necessary updates periodically (e.g. t_{k+1}^{GPS}, t_{k+2}^{MM}) in accordance with availability and need. Proceeding onward into the city centre the system temporarily experiences discontinued satellite reception as GPS satellite transmitted signals are blocked out due to shadowing affects caused by sky scrapers and tall building interference. In these shadowed zones, dead reckoning error is controlled by MM updates only (e.g. t_{k+3}^{MM}, t_{k+4}^{MM}, t_{k+5}^{MM}).

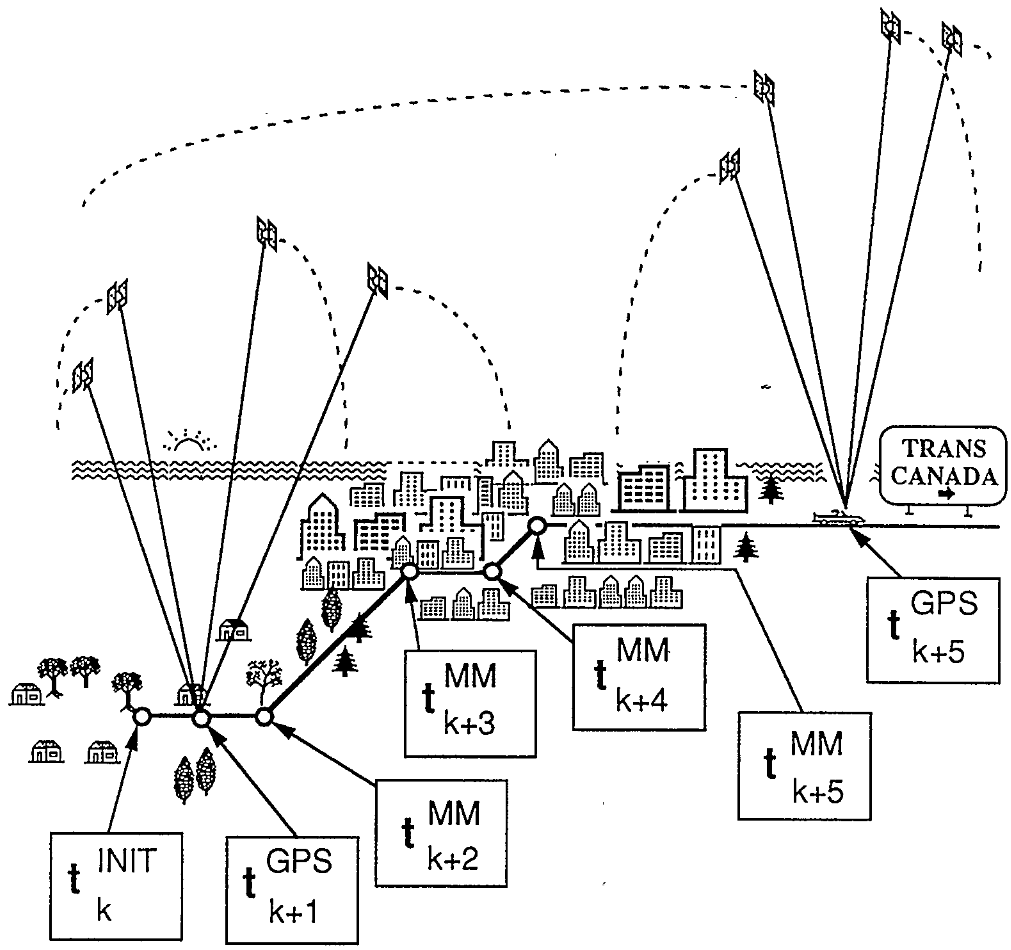


Figure 5.1 - Integration Concept

Exiting the city's downtown core, the vehicle heads towards the countryside for suburban or rural highway travel. On the long, open highway stretches where MM update occurrences will be few and far between, GPS fixes (e.g. t_{k+6}^{GPS}) will provide necessary coordinate updates to the differential odometer dead reckoning component.

The concept of combining differential odometer dead reckoning, map matching, and GPS satellite positioning has also been conceived recently by Nippondenso Co. Ltd. in Japan. Nippondenso engineers have designed into their new navigation system, MAPIX-III, such a combination along with electronic positioning sign posts situated on roadsides providing additional absolute position information [Ikeda et al., 1987]. The extent of the integration, however, only involves switching between augmented dead reckoning (i.e. differential odometer plus map matching) and absolute devices (i.e. GPS and sign post positioning). All are treated independently from one another (i.e. switching). The filter presented here is not based on simply switching from one to the other but rather blends all available information and knowledge together using covariance information to compute optimal estimates of both position states and related covariance matrix.

5.3 Mathematical Models

The integration concept of blending differential odometry DR, MM , and GPS positioning presented in the previous section can be represented mathematically. Three models may be conceived; two primary models, f_k and f_{k+1}, and one secondary model $g_{k,k+1}$. Formally, these may be written as [Krakiwsky et al., 1988]

$$f_k(\hat{x}_k, \hat{\ell}_k) = 0; \qquad \mathbf{C}_{\ell_k}, \tag{5.1}$$

$$f_{k+1}(\hat{x}_{k+1}, \hat{\ell}_{k+1}) = 0; \qquad \mathbf{C}_{\ell_{k+1}}, \tag{5.2}$$

$$g_{k,k+1}(\hat{x}_k, \hat{x}_{k+1}, \hat{y}_{k,k+1}, t) = 0; \quad \mathbf{C}_{y\,k,k+1}. \tag{5.3}$$

The primary model $\mathbf{f}_k$ is a function of both the unknown parameters $\hat{x}_k$ (e.g. coordinates) and the measurement update vector $\hat{\ell}_k$ at time t_k and carries with it knowledge of the covariance matrix $\mathbf{C}_{\ell_k}$. The cap (^) symbol denotes final adjusted values. The primary model $\mathbf{f}_{k+1}$ is a function of both $\hat{x}_{k+1}$ and $\hat{\ell}_{k+1}$ with covariance matrix $\mathbf{C}_{\ell_{k+1}}$ at time t_{k+1}. In accordance with Figure 5.1 these models represent a pair of updates in chronological sequence whether GPS or MM or one of each and may be successively applied to pairs of epochs beginning with t_k^{INIT} to t_{k+1}^{MM}, until all epoch pairs up to t_{k+5}^{MM} to t_{k+6}^{GPS}

have been covered. The secondary model $\mathbf{g}_{k,k+1}$ mathematically represents the differential odometer dead reckoning component which relatively positions the system over the interval t_k to t_{k+1}. It is a function of both the unknown vectors $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_{k+1}$, a model error vector $\hat{\mathbf{y}}_{k,k+1}$ (i.e. system noise) with covariance $\mathbf{C}_{\mathbf{y}_{k,k+1}}$ and time t.

The main objective of this modelling process is to solve for the state vector $\hat{\mathbf{x}}_{k+1}$ with corresponding covariance matrices $\mathbf{C}_{\hat{\mathbf{x}}_{k+1}}$. This is achieved by the simultaneous solution of the above three mathematical models (i.e. equations 5.1, 5.2, and 5.3) which is detailed by Krakiwsky [1989]. In this work both $\hat{\mathbf{x}}_{k+1}$ and $\mathbf{C}_{\hat{\mathbf{x}}_{k+1}}$ are algebraically derived from first principles of least squares and a direct analogy to the popular Kalman [1960] equations is made. The remainder of this section is devoted to describing both the general and detailed mathematical formulation and design incorporated into the prototype's kinematic positioning filter.

Frequently, the models above contain equation sets of the the non-linear variety. The first step is to linearize these. For the parametric case (i.e. $\boldsymbol{\ell} = \mathbf{f}(\mathbf{x})$)

$$\hat{\mathbf{r}}_k = \mathbf{A}_k \, \hat{\delta}_k + \mathbf{w}_k; \qquad \mathbf{C}_{\ell_k}, \tag{5.4}$$

$$\hat{\mathbf{r}}_{k+1} = \mathbf{A}_{k+1} \, \hat{\delta}_{k+1} + \mathbf{w}_{k+1}; \qquad \mathbf{C}_{\ell_{k+1}}, \tag{5.5}$$

$$\hat{\varepsilon}_{k,k+1} = \hat{\delta}_{k+1} - \Phi_{k,k+1}\,\hat{\delta}_k; \qquad \mathbf{C}_{\mathbf{y}_{k,k+1}}, \tag{5.6}$$

where

$\hat{\mathbf{r}}_k$ is the residual vector of measurement $\boldsymbol{\ell}_k$ at time t_k,

$\hat{\mathbf{r}}_{k+1}$ is the residual vector of measurement $\boldsymbol{\ell}_{k+1}$ at time t_{k+1},

$\hat{\varepsilon}_{k,k+1}$ is the system noise residual vector over t_k to t_{k+1},

$\Phi_{k,k+1}$ is the transition matrix transforming δ_k to δ_{k+1},

$\mathbf{A}_k$ is the first design matrix at time t_k determined by

$$\mathbf{A}_k = \left.\frac{d\mathbf{f}_k}{d\mathbf{x}_k}\right|_{\mathbf{x}_k^{\circ}}, \tag{5.7}$$

$\mathbf{A}_{k+1}$ is the first design matrix at time t_{k+1} equal to

$$\mathbf{A}_{k+1} = \left.\frac{d\mathbf{f}_{k+1}}{d\mathbf{x}_{k+1}}\right|_{\mathbf{x}_{k+1}^{\circ}}, \tag{5.8}$$

$\mathbf{w}_k$ is a misclosure vector at time t_k equivalent to

$$\mathbf{w}_k = \mathbf{f}_k(\mathbf{x}^{\circ}) - \boldsymbol{\ell}_k, \tag{5.9}$$

$\mathbf{w}_{k+1}$ is the misclosure at time t_{k+1} computed by

$$\mathbf{w}_{k+1} = \mathbf{f}_{k+1}(\mathbf{x}^{\circ}) - \boldsymbol{\ell}_{k+1}. \tag{5.10}$$

The superscript (°) denotes approximate values.

All unknowns can be solved in the above linear equations but only $\hat{\delta}_{k+1}$ with corresponding covariance matrix $\mathbf{C}_{\hat{\delta}_{k+1}}$ is of main concern due to the equation

$$\hat{\mathbf{x}}_{k+1} = \overset{\circ}{\mathbf{x}}_{k+1} + \hat{\delta}_{k+1}, \tag{5.11}$$

which is the least squares estimates of $\hat{\mathbf{x}}_{k+1}$ (the sought after solution) expressed linearly as truncated Taylor series expansions with expansion point $\overset{\circ}{\mathbf{x}}_{k+1}$ and correction vector $\hat{\delta}_{k+1}$. The strategy is to solve for the unknown correction vector $\hat{\delta}_{present}$ at time of update $t_{present}$ and apply it to an expansion point vector to yield the desired unknown system parameters contained in $\hat{\mathbf{x}}_{present}$.

When incorporating Kalman filtering techniques in positioning, two equation sets can be formed; a set of prediction equations and a set of update equations [Schwarz, 1987]. Adopting the mathematical notation presented by Krakiwsky [1989], the predicted state vector is

$$\hat{\delta}^{(-)}_{k+1} = \Phi_{k,k+1}\, \hat{\delta}^{(+)}_{k} + \hat{\varepsilon}_{k,k+1} \tag{5.12}$$

with the (-) sign denoting prediction. The following equation (5.13) is developed directly from the secondary model (equation 5.3) which can be expressed as

$$g_{k,k+1}(\hat{x}_k, \hat{x}_{k+1}, \hat{y}_{k,k+1}, t)$$

$$= -\Phi_{k,k+1}\hat{\delta}_k + \hat{\delta}_{k+1} - \hat{\varepsilon}_{k,k+1} = 0. \tag{5.13}$$

Assuming that all system noise is random with a mean of zero (i.e. $E[\hat{\varepsilon}_{k,k+1}] = 0$) it is dropped from the first moment equation 5.13 above leaving

$$\hat{\delta}^{(-)}_{k+1} = \Phi_{k,k+1}\hat{\delta}^{(+)}_k . \tag{5.14}$$

At the second moment level, however, the corresponding covariance matrix of the predicted state is computed by

$$\mathbf{C}^{(-)}_{\hat{\delta}_{k+1}} = \Phi_{k,k+1}\mathbf{C}^{(+)}_{\hat{\delta}_k}\Phi^T_{k,k+1} + \mathbf{C}_{\varepsilon_{k,k+1}}, \tag{5.15}$$

where

$\mathbf{C}_{\hat{\delta}_k}$ is the state vector covariance matrix at time t_k,

$\mathbf{C}_{\varepsilon_{k,k+1}}$ is the system noise residual covariance matrix which is equal to $\mathbf{C}_{y_{k,k+1}}$.

The update equations are applied immediately upon arrival of new measurement information. At this time (i.e. t_{k+1}) the system state estimate is updated using the predicted state vector $\hat{\delta}^{(-)}_{k+1}$ and its covariance $\mathbf{C}^{(-)}_{\hat{\delta}_{k+1}}$ along with the new measurements observed, contained within the misclosure vector w_{k+1}. The updated state vector is computed by

$$\hat{\delta}^{(+)}_{k+1} = \hat{\delta}^{(-)}_{k+1} - \mathbf{K}(\mathbf{w}_{k+1} + \mathbf{A}_{k+1}\,\hat{\delta}^{(-)}_{k+1}), \tag{5.16}$$

where **K** is the well known Kalman gain matrix weighting the misclosure vector contained in the parentheses to appropriately correct the predicted state $\hat{\delta}^{(-)}_{k+1}$ to the new updated state $\hat{\delta}^{(+)}_{k+1}$.

The Kalman gain matrix **K** is solved for using the equation

$$\mathbf{K} = \mathbf{C}^{(-)}_{\hat{\delta}_{k+1}} \mathbf{A}^{T}_{k+1}\,[\mathbf{C}_{\ell_{k+1}} + \mathbf{A}_{k+1}\mathbf{C}^{(-)}_{\hat{\delta}_{k+1}}\mathbf{A}^{T}_{k+1}]^{-1}. \tag{5.17}$$

The covariance matrix of the updated state vector $\hat{\delta}^{(+)}_{k+1}$ is also computed using the gain **K** in the following manner:

$$\mathbf{C}^{(+)}_{\hat{\delta}_{k+1}} = \mathbf{C}^{(-)}_{\hat{\delta}_{k+1}} - \mathbf{K}\mathbf{A}_{k+1}\mathbf{C}^{(-)}_{\hat{\delta}_{k+1}}. \tag{5.18}$$

The above equations (5.1 through 5.18) highlight the general mathematics involved with the implemented filter. Reference to more in depth discussions and derivations on Kalman filtering and estimation theory is directed towards Gelb [1974], Brown [1983], Schwarz [1987], and Krakiwsky [1989]. Presented now are the specific contents of the general matrix equations used for the prototype's filter testing.

Three states are modelled and are contained in the vector

$$\hat{\delta} = \begin{bmatrix} \delta\hat{x} \\ \delta\hat{y} \\ \delta\hat{\alpha} \end{bmatrix}, \tag{5.19}$$

where

$\delta\hat{x}$ is the corrective term of the estimated UTM x coordinate of the vehicle's current position,

$\delta\hat{y}$ is the corrective term of the estimated UTM y coordinate of the vehicle's current position,

$\delta\hat{\alpha}$ is the corrective term of the estimated UTM grid azimuth α of the forward heading of the vehicle.

When computed, $\hat{\delta}$ is added to the dead reckoned (DR) coordinates and azimuth of the vehicle, which are expansion points, to produce the best estimate of these unknown parameters at time t_{k+1}. In equation form this is expressed as

$$\begin{bmatrix} \hat{x}_{k+1}^{(+)} \\ \hat{y}_{k+1}^{(+)} \\ \hat{\alpha}_{k+1}^{(+)} \end{bmatrix} = \begin{bmatrix} \overset{\circ}{x}_{DR} \\ \overset{\circ}{y}_{DR} \\ \overset{\circ}{\alpha}_{DR} \end{bmatrix} + \begin{bmatrix} \delta\hat{x}_{k+1}^{(+)} \\ \delta\hat{y}_{k+1}^{(+)} \\ \delta\hat{\alpha}_{k+1}^{(+)} \end{bmatrix}, \tag{5.20}$$

also expressed as

$$\hat{x}_{k+1}^{(+)} = \hat{x}_{k+1}^{(-)} + \hat{\delta}_{k+1}^{(-)}. \tag{5.21}$$

The vector $\hat{x}_{k+1}^{(-)}$ is computed using the dead reckoning equations (equations 4.4, 4.5, and 4.17) relative to the previous best estimate $\hat{x}_k^{(+)}$ at time t_k.

To begin with

$$\hat{\delta} = \varnothing, \tag{5.22}$$

where $\varnothing$ is a null vector, as no information exists to better estimate the true value of the error in **x** at the first moment level. The covariance matrix of $\hat{\delta}$ is more than likely not equal to zero for AVLN applications initially and is assumed diagonal when first starting out. This matrix is initially represented by

$$\mathbf{C}_{\hat{\delta}} = \begin{bmatrix} \sigma_{\delta\hat{x}}^2 & 0 & 0 \\ 0 & \sigma_{\delta\hat{y}}^2 & 0 \\ 0 & 0 & \sigma_{\delta\hat{\alpha}}^2 \end{bmatrix}. \tag{5.23}$$

From equation 5.11, it is clear that the covariance matrix for $\hat{x}$ ($\mathbf{C}_{\hat{x}}$) is equivalent to $\mathbf{C}_{\hat{\delta}}$ because x_o is a point of expansion in the Taylor series and is treated as constant from a covariance law point of view [Krakiwsky and Gagnon, 1987]. Thus

$$\mathbf{C}_{\hat{x}} = \mathbf{C}_{\hat{\delta}} \quad , \tag{5.24}$$

and thus the solution at any time t of $\mathbf{C}_{\hat{\delta}}$, designated $^{(+)}$ or $^{(-)}$, gives knowledge of the estimated error in vehicle coordinates and heading at time t.

Obviously, from equation 5.25, if $\delta = \varnothing$, then the predicted value $\hat{\delta}^{(-)}_{k+1}$ into the future using equation 5.14 is also zero. At the first moment level

$$\hat{\delta}^{(-)}_{k+1} = \Phi_{k,k+1}\,\hat{\delta} = \varnothing, \tag{5.25}$$

and it is observed no knowledge of the transition matrix $\Phi_{k,k+1}$ is required. At the second moment, however, this is not the case and propagation is dependent on the transition matrix $\Phi_{k,k+1}$ (equation 5.15).

Evaluating equation 5.14 enables one to see that $\Phi_{k,k+1}$ is defined as a transformation operator, transforming system states at epoch t_k to epoch t_{k+1}. The algebraic solution of $\Phi_{k,k+1}$ may be computed by utilizing Laplace transform techniques operating on a set of linear, first order set of differential equations which describe the system [Brown, 1983]. Here, $\Phi_{k,k+1}$ is approximated by expressing the first order partial derivatives of the differential odometer dead reckoning equations (equations 4.4, 4.5, and 4.17) with respect to the unknown parameters x, y, and α. For small intervals $\Delta t_{i,i+1}$, δx, δy, and $\delta\alpha$ can be estimated from one epoch to the next in this fashion.

Often is the case that the interval t_k to t_{k+1} between updates (i.e. GPS or MM) is too large to facilitate such required time intervals due to large, associated linearization error. As a result, a smaller time, in between update interval, t_i to t_{i+1}, is selected and employed for covariance propagation. This requires a new transition matrix to be formed for each i^{th} interval represented by $\phi_{i,i+1}$.

Seen illustrated in Figure 5.2, several intervals of propagation $t_{i=0}$ through to t_n conceivably exist between update epochs t_k and t_{k+1}; the value n being dependent upon the sampling interval $\Delta t_{i,i+1}$ and the frequency of updates dictated by $\Delta t_{k,k+1}$. Over the update interval, $\Phi_{k,k+1}$ can be expressed as a function of $\phi_{i,i+1}$ by the matrix product

$$\Phi_{k,k+1} = \prod_{i=0}^{n} \phi_{i,i+1}. \tag{5.26}$$

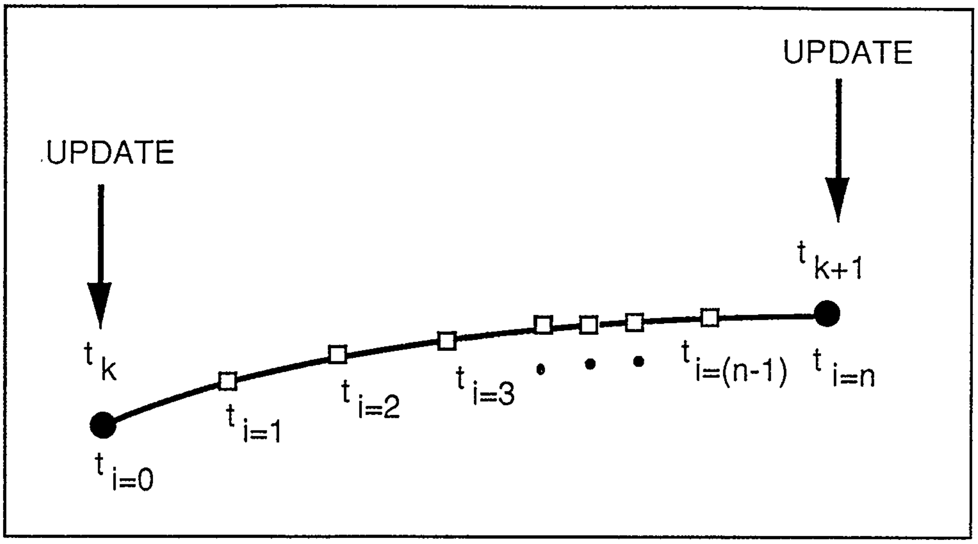


Figure 5.2 - Propagation Interval

In light of equation 5.26, equations 5.14 and 5.15 can be redefined for the interval $\Delta t_{i,i+1}$ by substituting i for k and (+) for (-) (except on the first iteration) resulting in

$$\hat{\delta}^{(-)}_{i+1} = \phi_{i,i+1}\, \hat{\delta}^{(-)}_{i}, \tag{5.27}$$

and

$$\mathbf{C}^{(-)}_{\hat{\delta}_{i+1}} = \phi_{i,i+1}\, \mathbf{C}_{\hat{\delta}_i}\, \phi^{T}_{i,i+1} + \mathbf{C}_{\varepsilon_{i,i+1}}, \tag{5.28}$$

where

$\mathbf{C}_{\hat{\delta}_i}$ is the state vector covariance matrix at time t_i ,

$\mathbf{C}_{\varepsilon_{i,i+1}}$ is the system noise covariance matrix over t_i to t_{i+1}.

When $i+1 = n$, $t_{i+1} = t_n$ which equals t_{k+1} and

$$\mathbf{C}^{(-)}_{\hat{\delta}_{k+1}} = \mathbf{C}^{(-)}_{\hat{\delta}_n} = \mathbf{C}^{(-)}_{\hat{\delta}_{i+1}}, \tag{5.29}$$

is the required covariance matrix at time of update t_{k+1}.

Employing this iterative method to suppress linearization effects allows the filter designer to select a prespecified fixed interval $\Delta t_{i,i+1}$ accommodating varying measurement update intervals (i.e. temporally non-uniform). Now we are in a position to apply the above equations to the case of dead reckoning between updates k and k+1.

Recalling Chapter 4 notation and expanding out fully equations 4.4, 4.5, and 4.17 yields equations for the unknown parameters x_{i+1}, y_{i+1}, and α_{i+1} as functions of the observables $\Delta d^L_{i,i+1}$, $\Delta d^R_{i,i+1}$ and the parameters x_i, y_i, and α_i. These equations are:

$$x_{i+1} = x_i + \left[\frac{\Delta d^L_{i,i+1} + \Delta d^R_{i,i+1}}{2}\right] \sin\left[\alpha_i + \frac{\Delta d^L_{i,i+1} - \Delta d^R_{i,i+1}}{2 \cdot \mathrm{TRACK}}\right], \tag{5.30}$$

$$y_{i+1} = y_i + \left[\frac{\Delta d^L_{i,i+1} + \Delta d^R_{i,i+1}}{2}\right] \cos\left[\alpha_i + \frac{\Delta d^L_{i,i+1} - \Delta d^R_{i,i+1}}{2 \cdot \mathrm{TRACK}}\right], \tag{5.31}$$

$$\alpha_{i+1} = \alpha_i + \frac{\Delta d^{L}_{i,i+1} - \Delta d^{R}_{i,i+1}}{TRACK} . \qquad (5.32)$$

Note all three quantities are functions of the same observables, thus explaining the subsequent correlation (equation 5.43). By taking the partial derivatives of equations 5.30, 5.31, and 5.32 first with respect to the unknowns allows equation 5.27 to be written in a most expanded form explicitly as

$$\begin{bmatrix} \delta\hat{x}^{(-)}_{i+1} \\ \delta\hat{y}^{(-)}_{i+1} \\ \delta\hat{\alpha}^{(-)}_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \bar{\Delta d}\cdot c_n \\ 0 & 1 & -\bar{\Delta d}\cdot s_n \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta\hat{x}^{(-)}_{i} \\ \delta\hat{y}^{(-)}_{i} \\ \delta\hat{\alpha}^{(-)}_{i} \end{bmatrix} , \qquad (5.33)$$

$$\hat{\delta}^{(-)}_{i+1} = \phi_{i,i+1} \quad \hat{\delta}^{(-)}_{i} ,$$

where

$$c_n = \cos\left[\alpha_i + \frac{\Delta\,\alpha_{i,i+1}}{2} \right] , \qquad (5.34)$$

$$s_n = \sin\left[\alpha_i + \frac{\Delta\,\alpha_{i,i+1}}{2} \right] . \qquad (5.35)$$

Recall that the $^{(-)}$ is a $^{(+)}$ on the first iteration in equation 5.36 as pointed out in equation 5.27. In order to predict $\mathbf{C}_{\hat{\delta}_{i+1}}$, all that is required now is $\mathbf{C}_{\varepsilon_{i,i+1}}$.

This matrix is determined by propagating odometer sensor noise into state variance by application of the covariance law [Mikhail, 1976], namely:

$$\mathbf{C}_{\varepsilon_{i,i+1}} = \mathbf{J}\ \mathbf{C}_{odom_{i,i+1}}\ \mathbf{J}^T, \tag{5.36}$$

where

$\mathbf{J}$	is a Jacobian matrix,
$\mathbf{C}_{odom_{i,i+1}}$	is the odometer sensor observation noise.

The Jacobian matrix **J** contains the partial derivatives of the differential odometer dead reckoning equations (5.30, 5.31, and 5.32) with respect to the odometer sensor observables $\Delta d^L_{i,i+1}$ and $\Delta d^R_{i,i+1}$ and not to the unknown parameters. This Jacobian matrix is thus

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial x_{i+1}}{\partial \Delta d^{L}_{i,i+1}} & \dfrac{\partial x_{i+1}}{\partial \Delta d^{R}_{i,i+1}} \\[2ex] \dfrac{\partial y_{i+1}}{\partial \Delta d^{L}_{i,i+1}} & \dfrac{\partial y_{i+1}}{\partial \Delta d^{R}_{i,i+1}} \\[2ex] \dfrac{\partial \alpha_{i+1}}{\partial \Delta d^{L}_{i,i+1}} & \dfrac{\partial \alpha_{i+1}}{\partial \Delta d^{R}_{i,i+1}} \end{bmatrix}, \tag{5.37}$$

which is equal in this case to

$$\mathbf{J} = \begin{bmatrix} \dfrac{s_n}{2} + \dfrac{\overline{\Delta d} \cdot c_n}{2 \cdot \mathrm{TRACK}} & \dfrac{s_n}{2} - \dfrac{\overline{\Delta d} \cdot c_n}{2 \cdot \mathrm{TRACK}} \\[2ex] \dfrac{c_n}{2} - \dfrac{\overline{\Delta d} \cdot s_n}{2 \cdot \mathrm{TRACK}} & \dfrac{c_n}{2} + \dfrac{\overline{\Delta d} \cdot s_n}{2 \cdot \mathrm{TRACK}} \\[2ex] \dfrac{1}{\mathrm{TRACK}} & \dfrac{-1}{\mathrm{TRACK}} \end{bmatrix}. \tag{5.38}$$

The covariance matrix of the odometer sensor observations is

$$\mathbf{C}_{odom_{i,i+1}} = \begin{bmatrix} \sigma^2_{\Delta d^L_{i,i+1}} & 0 \\ 0 & \sigma^2_{\Delta d^R_{i,i+1}} \end{bmatrix} . \tag{5.39}$$

The first diagonal element of $\mathbf{C}_{odom_{i,i+1}}$ is the variance of the left (L) wheel odometer computed by

$$\sigma^2_{\Delta d^L_{i,i+1}} = \left(\Delta d^L_{i,i+1} \cdot 10^{-3} \right)^2 , \tag{5.40}$$

and similarly, the second diagonal element for the right odometer (R)

$$\sigma^2_{\Delta d^R_{i,i+1}} = \left(\Delta d^R_{i,i+1} \cdot 10^{-3} \right)^2 , \tag{5.41}$$

recalling that $1 \cdot 10^{-3}$ is the NU-METRICS odometer manufacturer's quoted relative accuracy of the sensors. No off diagonal elements exist with the assumption that each wheel is uncorrelated.

Substituting equation 5.36 into equation 5.28 results in

$$\mathbf{C}^{(-)}_{\hat{\delta}_{i+1}} = \phi_{i,i+1} \, \mathbf{C}_{\hat{\delta}_i} \, \phi^T_{i,i+1} + \mathbf{J} \, \mathbf{C}_{odom_{i,i+1}} \, \mathbf{J}^T , \tag{5.42}$$

which in hyper-matrix form is

$$C_{\hat{\delta}_{i+1}}^{(-)} = \begin{bmatrix} \phi_{i,i+1} & J \end{bmatrix} \begin{bmatrix} C_{\hat{\delta}_i} & \varnothing \\ \varnothing & C_{odom_{i,i+1}} \end{bmatrix} \begin{bmatrix} \phi_{i,i+1}^T \\ J^T \end{bmatrix} . \quad (5.43)$$

Clearly, correlation exists between x, y, and α, as well as between $\Delta\alpha$ and $\bar{\Delta d}$ which are both functions of $\Delta d_{i,i+1}^L$ and $\Delta d_{i,i+1}^R$. This fact is what makes error propagation in differential odometer dead reckoning different from that in a survey traverse where distances and angles are uncorrelated.

When the vehicle is stationary (i.e. $\bar{\Delta d} = 0$) the transition matrix $\phi_{i,i+1}$ collapses to an identity matrix $\mathbf{I}$, by inspection of equation 5.33. Realizing this, equation 5.43 can be viewed as covariance propagation of a direct model (i.e. unknowns as function of observables) of the unknown parameters, treating the parameters x_i, y_i, and α_i as quasi-observables with observation covariance $C_{\hat{\delta}_i}$. As this is the case, hesitation to use the term 'Kalman' filter arises due to the fact that true error propagation is implicit in the above mathematics with no form of prediction taking place even though it may be conceived in this manner. To better characterize the situation, the more specific term 'kinematic position filter' is used. This notion belongs to a conservative school of thought and others may argue that, since the transition matrix is non-identity, it is in fact a Kalman filter.

The update equations 5.16 and 5.18 are employed for three update scenarios used for testing, namely: map matching azimuth only (MM, α); map matching x,y, and α updates (MM); and, satellite derived x and y (GPS). The mathematics of each scenario is explained here. For each scenario, the arrays $\mathbf{w}_{k+1}$, $\mathbf{A}_{k+1}$, and $\mathbf{C}_{\ell_{k+1}}$ differ. As all other matrices contained in equations 5.16 ($\hat{\delta}^{(+)}_{k+1}$) and 5.18 ($\mathbf{C}\,\hat{\delta}^{(+)}_{k+1}$)have been derived and are not update measurement dependent, only these are presented here; one set for each update type.

Before continuing it should be noted that at the first moment level (i.e. $\hat{\delta}^{(+)}_{k+1}$), if $\hat{\delta}^{(+)}_{k+1}$ is used to correct the unknown parameters at every update epoch and these corrected values (the best estimates) are then used as the new expansion points for dead reckoning reference then $\hat{\delta}^{(-)}_{k+1}$ is always estimated as zero as in equation 5.22. The ramification of this at the update level is the reduction of equation 5.16 to

$$\hat{\delta}^{(+)}_{k+1} = -\mathbf{K}\,\mathbf{w}_{k+1}, \tag{5.44}$$

which holds true for each update type.

The MM,α update involves updating the state of the system whenever the vehicle's heading is matched to a street segment's azimuth. It is noted that this update is not a LOP (Line Of Position) update but strictly an azimuth only. The update equation for this case is the parametric equation

$$\alpha^{MM,\alpha}_{k+1} = \alpha_{k+1}. \tag{5.45}$$

The associated design matrix from equations 5.7 and 5.8 is

$$\mathbf{A}_{k+1}^{MM,\alpha} = [\; 0 \quad 0 \quad 1 \;]\,, \tag{5.46}$$

and the misclosure vector:

$$\mathbf{w}_{k+1}^{MM,\alpha} = \left[\; (\alpha^{DR} - \alpha^{MM}) \;\right]. \tag{5.47}$$

For this case the measurement update covariance matrix is a one dimensional array containing only one element as only one measurement is made (i.e. α_{MM}). This single element matrix is

$$\mathbf{C}_{\ell_{k+1}}^{MM,\alpha} = \left[\; \sigma_{\alpha}^{2\,MM} \;\right] = \left[\; \frac{\sigma_{lkdist}^{2}}{lkdist^{2}} \;\right], \tag{5.48}$$

where

$$\sigma_{lkdist}^{2} = \sigma_{\Delta x}^{2} + \sigma_{\Delta y}^{2} = 25.0\ (\text{metres})^{2}\,, \tag{5.49}$$

σ_{lkdist} is the nominal AMF relative standard error between intersection coordinates (i.e. 5 metres),

$$lkdist = (\Delta x^{2} + \Delta y^{2})^{1/2}\,, \tag{5.50}$$

the distance of the street link whose end point coordinates derived the measured azimuth being used for the update.

The variance $\sigma_{\alpha}^{2}{}^{MM}$ (equation 5.47) is computed by applying the covariance law of error propagation [Mikhail, 1976] to equation 4.1b (azimuth as a function of coordinate differences Δx, Δy).

At time of update an additional variance $\sigma_{\alpha_q}^{2}$ is added to the $\mathbf{C}\,\hat{\delta}_{k+1}^{(-)}$ azimuth element to account for the quantization error [Chapman, 1988] associated with dead reckoned azimuth pickoff and is equal to

$$\sigma_{\alpha_q}^{2} = E(x-\mu)^2 = \int_{-0.5}^{+0.5} (x-\mu)^2 p(x)\,dx = 0.083\ \alpha_q^{2}, \qquad (5.51)$$

where

$p(x)$ is the probability of azimuth step size x (for equal probability $p(x) = 1$),

μ is the mean of the distribution (for interval -0.5 to +0.5; $\mu = 0$),

α_q is the azimuth quantization step size.

Using α_q equal to 9.4 degrees (equation 4.19), the standard deviation σ_{α_q} is equal to 2.7 degrees or 0.047 radians. This azimuth quantization is introduced only at the second moment level at time of update.

The second update type, MM with x,y, and α combined has three update equations of the parametric type, namely:

$$x_{k+1}^{MM} = x_{k+1}, \tag{5.52}$$

$$y_{k+1}^{MM} = y_{k+1}, \tag{5.53}$$

$$\alpha_{k+1}^{MM} = \alpha_{k+1}. \tag{5.54}$$

The design matrix $\mathbf{A}_{k+1}^{MM}$ is an identity matrix

$$\mathbf{A}_{k+1}^{MM} = \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.55}$$

and the corresponding misclosure vector is

$$\mathbf{w}_{k+1}^{MM} = \begin{bmatrix} (x_{k+1}^{DR} - x_{k+1}^{MM}) \\ (y_{k+1}^{DR} - y_{k+1}^{MM}) \\ (\alpha_{k+1}^{DR} - \alpha_{k+1}^{MM}) \end{bmatrix}. \tag{5.56}$$

The measurement covariance matrix for a MM update is

$$\mathbf{C}_{\ell_{k+1}}^{MM} = \begin{bmatrix} \sigma_x^2{}^{MM} & 0 & 0 \\ 0 & \sigma_y^2{}^{MM} & 0 \\ 0 & 0 & \sigma_\alpha^2{}^{MM} \end{bmatrix}, \qquad (5.57)$$

where

$\sigma_x^2{}^{MM}$ is the measured map matched x coordinate variance (20 metres)^2 used for AMF data - nominal absolute coordinate error,

$\sigma_y^2{}^{MM}$ is the measured map matched y coordinate variance (20 metres)^2 - 20 metres nominal absolute coordinate error of AMF data,

$\sigma_\alpha^2{}^{MM}$ is the map matched azimuth variance (same as equation 5.48).

As was performed for the MM,α case above, at time of update additional azimuth variance $\sigma_{\alpha_q}^2$ was added onto the estimated variance of the dead reckoned azimuth.

The third update type is GPS and involves only x and y updating, not azimuth α. The update equations are

$$x_{k+1}^{GPS} = x_{k+1}, \tag{5.58}$$

$$y_{k+1}^{GPS} = y_{k+1}. \tag{5.59}$$

both computed from equations 4.39 through to 4.51.

The GPS update's first design matrix is the unit matrix

$$\mathbf{A}_{k+1}^{GPS} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{5.60}$$

The misclosure is computed by the difference between GPS coordinates and dead reckoned coordinates and is given as

$$\mathbf{w}_{k+1}^{GPS} = \begin{bmatrix} (x_{k+1}^{DR} - x_{k+1}^{GPS}) \\ (y_{k+1}^{DR} - y_{k+1}^{GPS}) \end{bmatrix}. \tag{5.61}$$

The associated measurement covariance matrix is the diagonal array

$$\mathbf{C}_{\ell_{k+1}}^{GPS} = \begin{bmatrix} \sigma_{x_{GPS}}^2 & 0 \\ 0 & \sigma_{y_{GPS}}^2 \end{bmatrix}. \tag{5.62}$$

The diagonal elements $\sigma^2_{x_{GPS}}$ and $\sigma^2_{y_{GPS}}$ are computed using the variance equation 4.53 which is a function of the HDOP. No off-diagonal covariance elements are available from the TANS receiver when ϕ, λ, and h are retrieved directly.

For each of the above three update scenarios, the first design matrix **A** for each case is a unit matrix (i.e. elements of 0 or 1 value). Due to this, when applying these equations to a real-time environment, the equations should be fully expanded and multiplied out to eliminate any unnecessary unit or zero operations.

It is noted for future development if the update equations are set up directly as functions of the pseudo-range observables, the design matrix $\mathbf{A}^{GPS}$ becomes more complex than above design matrix and the correlation between x^{GPS} and y^{GPS} may be carried and entered into the filtered solution in the form of non-diagonal covariance elements of the $\mathbf{C}^{GPS}_{\ell_{k+1}}$ matrix. Computationally this becomes more demanding on the system's CPU load and perhaps should be handled at the receiver end in a multi-processing approach.

5.4 Field Tests and Analyses

The filter equations detailed in the last section were applied to data collected on a test driven route within the test network. The route selected is highlighted bold on the digital map in Figure 5.3 and is just short of 10 kilometres in total length journeyed. Speeds of the vehicle ranged from 0 to 75

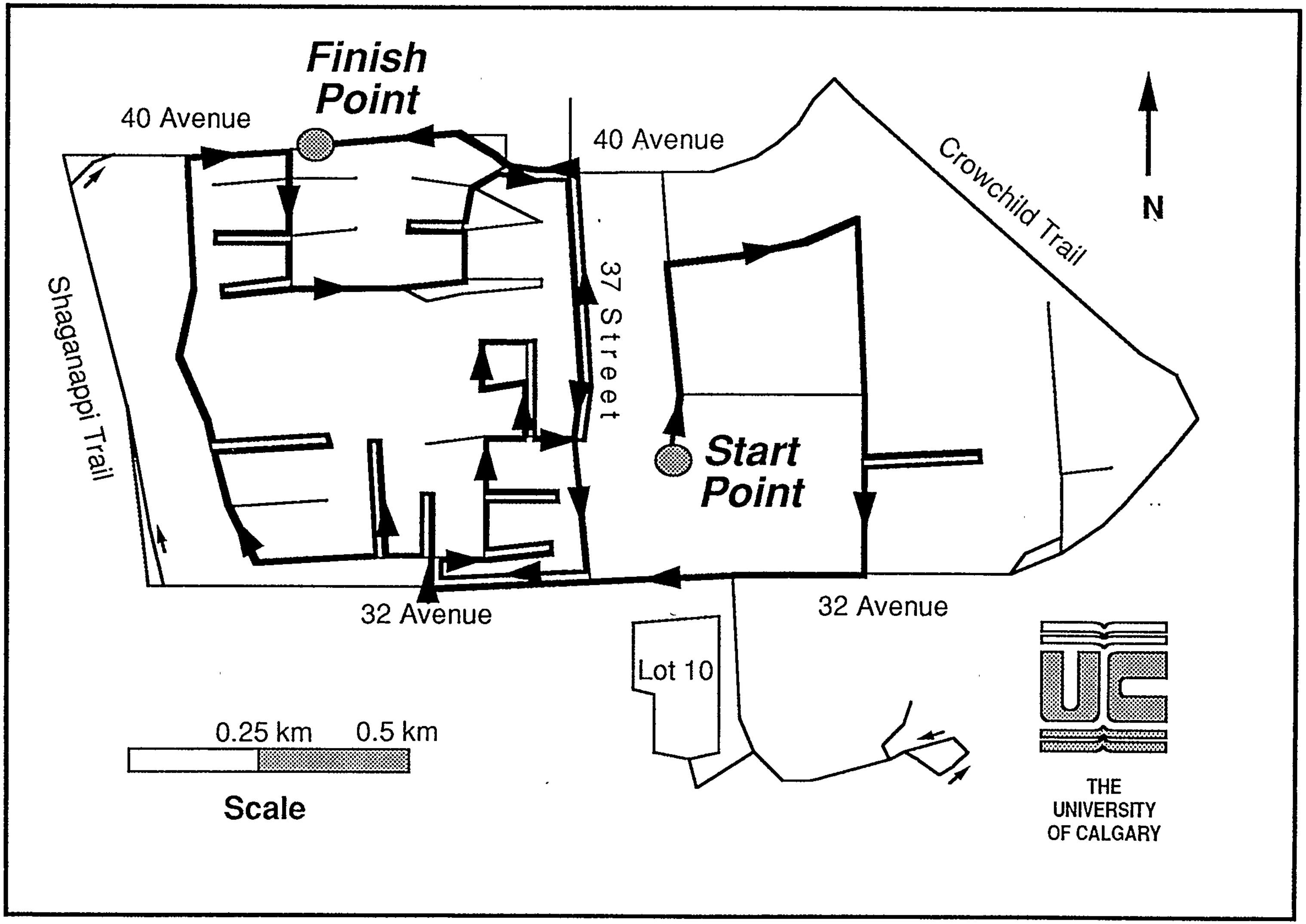


Figure 5.3 - Test Route Driven

kilometres per hour and an effort was made to use a large majority of the streets contained in the network. Both start and finish points are indicated on the map.

Odometer, GPS and MM data was collected and stored in real-time in the system on-board navigation computer. The collection frequency was 10 Hz (i.e. interval t_i to t_{i+1}) for both left and right odometers, 0.1 Hz for GPS fixes, and at every intersection and cul-de-sac turn for MM updates. GPS data was screened allowing only those fixes locked onto four satellites and possessed an HDOP value of 4.5 or less. Due to this screening GPS fixes may have been 20 or 30 second apart. The logged data was transferred to a lab computer (COMTEX IBM compatible containing an Intel 8088 CPU) to be fed through the filter software for testing. The data logging methodology allowed for initial filter development to occur in the laboratory with real sensor data.

Five computational tests are performed on the data collected: differential odometer DR with no updates, DR with map matched azimuth (MM,α) updates only, DR with map matched x, y, and α updates (MM), and combined MM with GPS - x and y updates. Plots of both x and y error (as determined by GPS and MM coordinate updates) of the vehicle and associated covariance error envelopes (3 σ) are displayed for each case.

Figure 5.4a and 5.4b illustrate the open ended DR case (i.e. no updates). The error envelopes are computed by propagating $\mathbf{C}_{\hat{\delta}\,k+1}^{(-)}$ (equations 5.15 and 5.31) from the beginning of the test route right through to the end. Note the y axis ranges from positive 600 metres to negative 600 metres and it is seen that over this run the vehicle's position error in the x coordinate peaks to approximately 80 metres, and in the y coordinate grows to about 120 metres. The total radial error is thus 1% to 2% of the distance run. The error envelopes show generally growing characteristics but undulate due to the segregation into x and y components combined with the geometric configuration (i.e. many turn backs) of the test run. This is also in conjunction with the realization that any one point is dependent on and thus correlated to all other points previously dead reckoned to. Notice in both x and y plots, the error envelopes encapsulate all of the error.

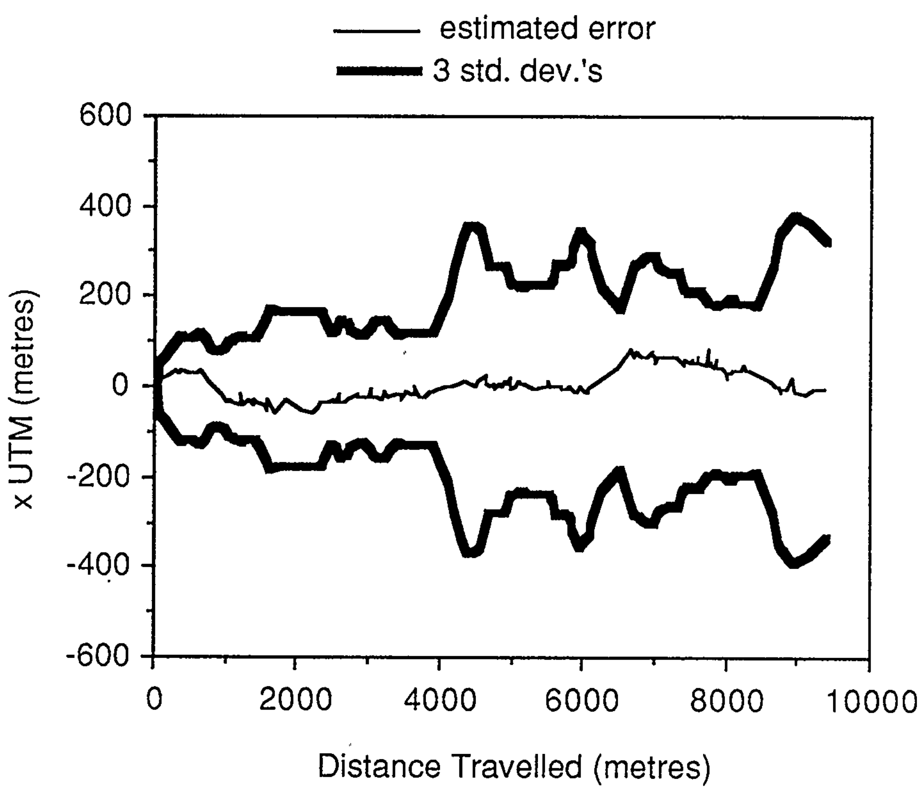


Figure 5.4a - DR Open Ended - x Plot

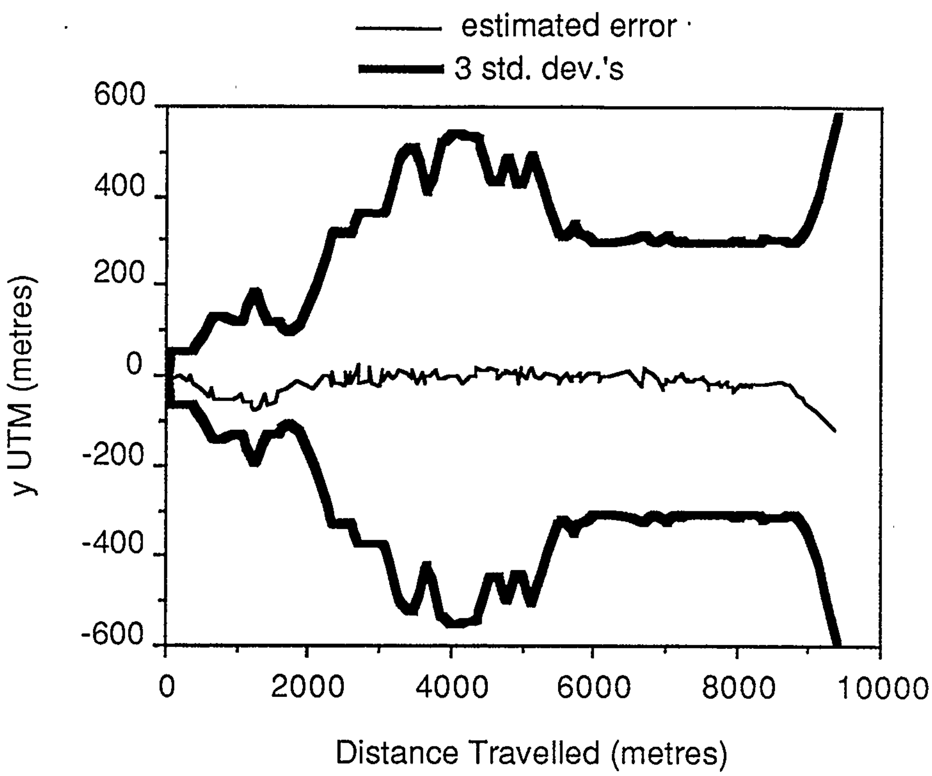


Figure 5.4b - DR Open Ended - y Plot

Figure 5.5a and 5.5b show the filter result for the MM,α case. As one can see, both error envelopes are rising slopes, quite linear in nature. The few spikes in the results reflect the significant correlation existing between x and α, and y and α, at these points along the run. The error in x is roughly 1% of the distance travelled and likewise in y, approximately 1% of the distance travelled. A few errors have escaped the enveloped region of three standard deviations (i.e. 3σ) in Figure 5.5b (y error plot). Three large MM azimuth update misclosures (**w**) exist along the run (data snoop detected) and are the source of these systematic affects. These deviant misclosures are most likely due to:

1) poor quality map data for that update link;
2) vehicle heading is not completely in line with the update link azimuth; or

3) a few missed odometer pulses (9.4 degrees per missed pulse - see equation 4.19),

the latter being least likely in this case by inspection of Figures 5.4a and 5.4b.

One way to assist in accommodating the affects of these large misclosures related to 1) and 2) above is to increase the variance placed on map matched azimuths. Missed odometer pulses can not be treated in this fashion although such blunder detection is feasible if an additional heading sensor is implemented.

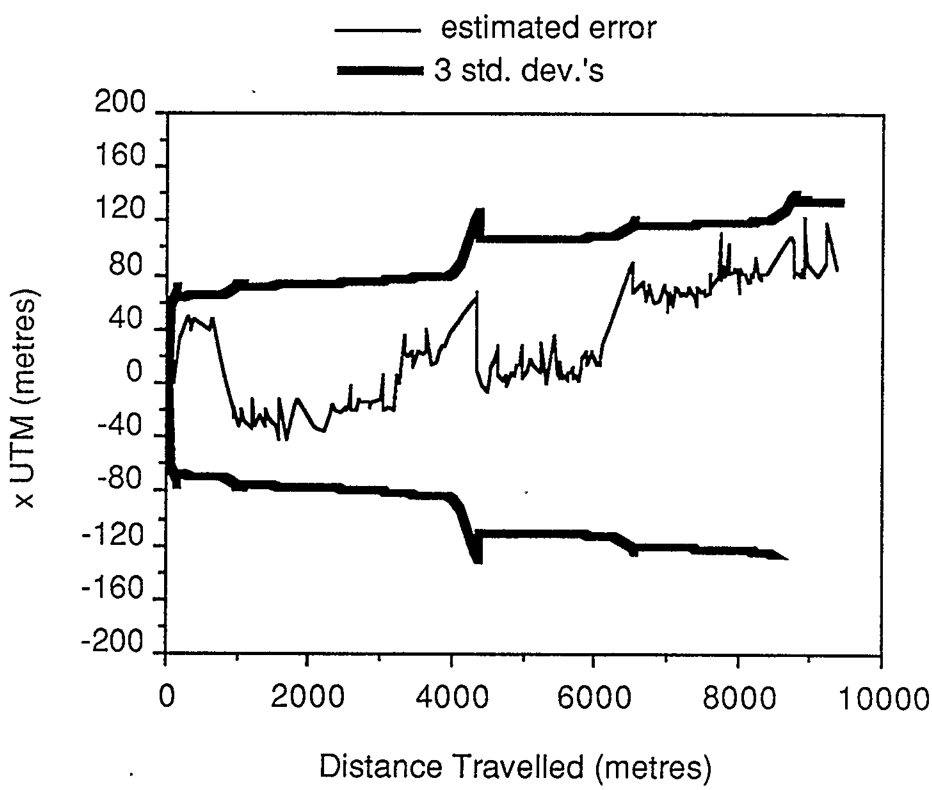


Figure 5.5a - DR and MM,α - x Plot

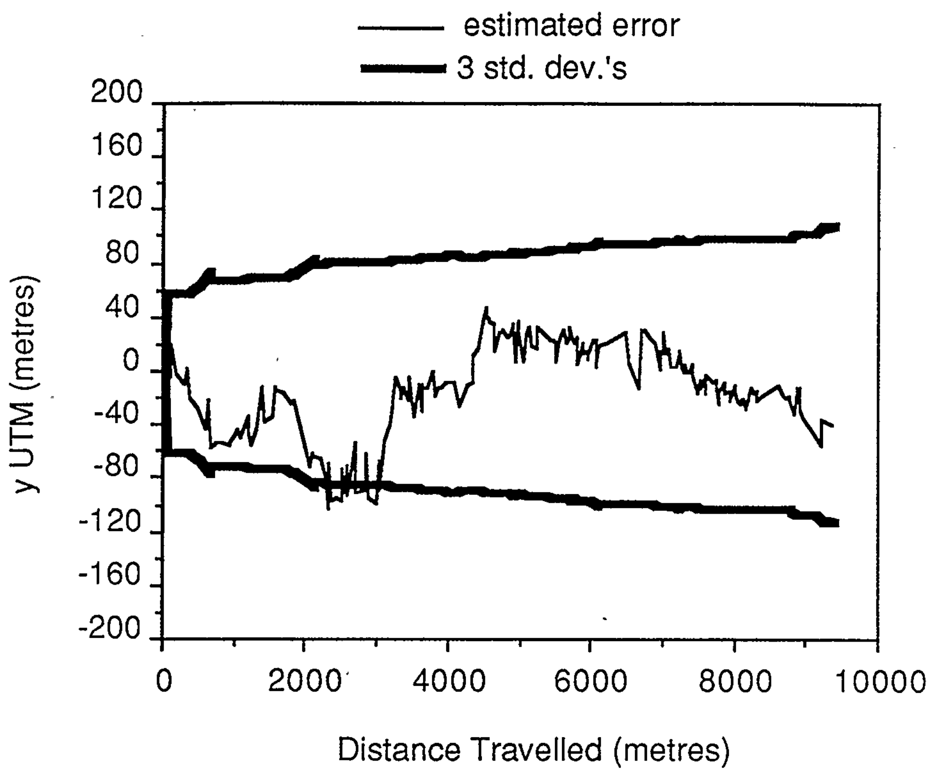


Figure 5.5b - DR and MM,α - y Plot

When DR and MM are combined the results in x and y error (Figures 5.6a and 5.6b) are better controlled in general with improvements up to 80 metres in some cases. Systematic error still seems to creep in causing error to escape the estimated random error envelopes in some instances. Regardless, error is controlled to the 40 metre level in the most part for both x and y throughout the entire run. One must realize that the error curves plotted are evaluated using the GPS pseudo-range solutions and MM updates which both could be in error themselves as much as 30 metres at any one time.

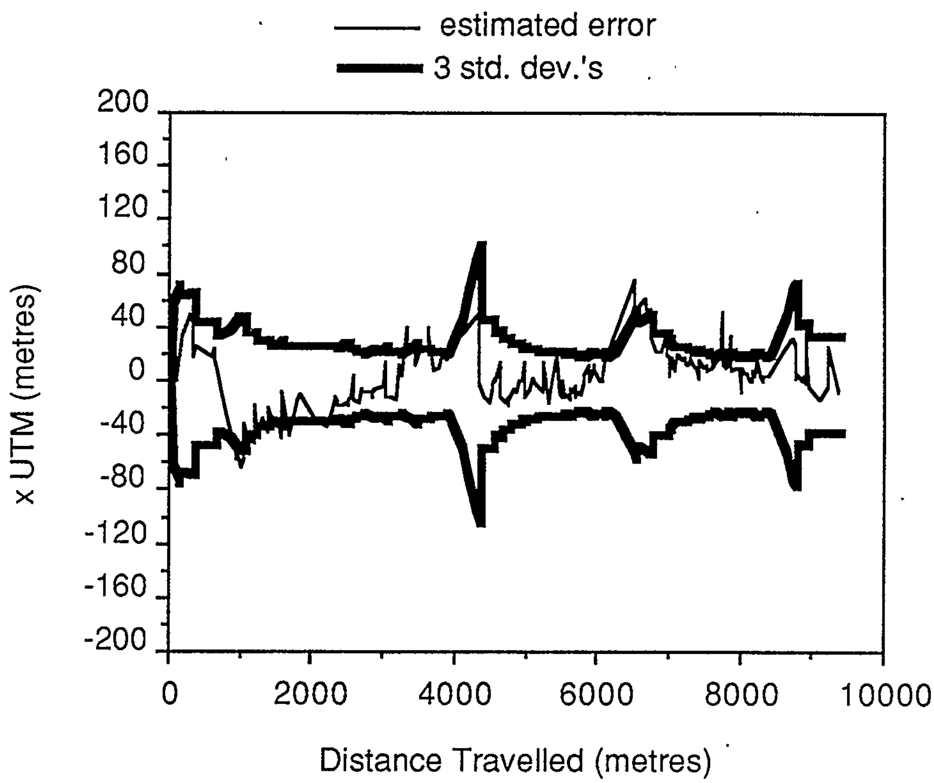


Figure 5.6a - DR and MM - x Plot

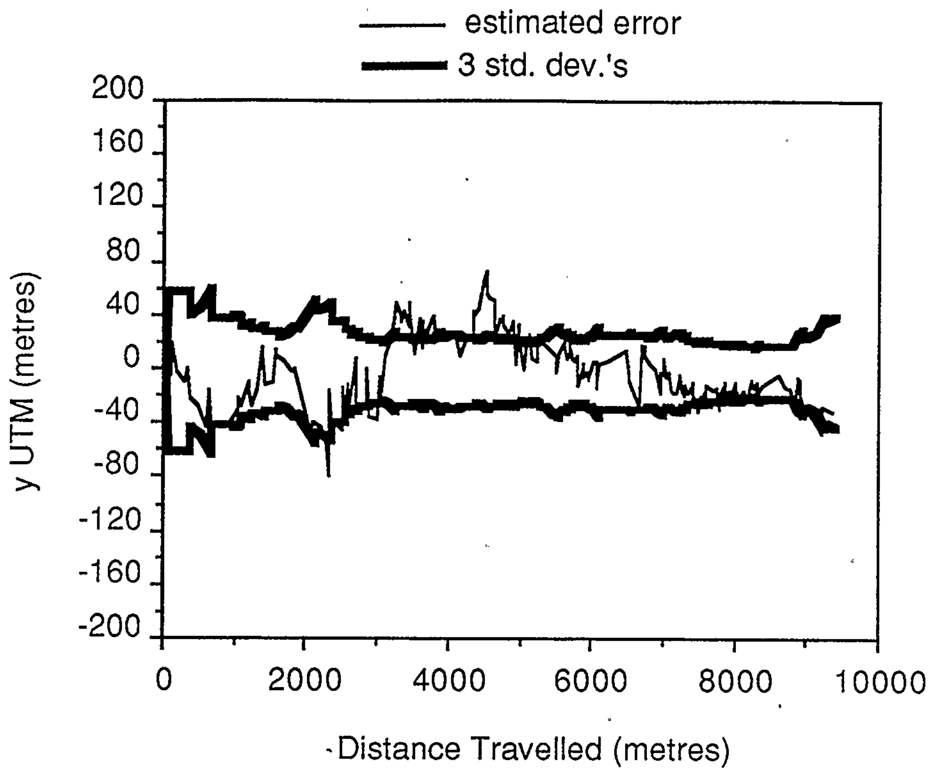


Figure 5.6b - DR and MM - y Plot

Figures 5.7a and 5.7b display the filter data for the case when MM, and GPS updates are both utilized. As expected, the error envelope sizes have narrowed due to the increase in update frequency. Systematic error still prevails due to the large MM azimuth misclosures discussed previously. By step to step inspection of the filter process, it was observed that one GPS or MM update with standard deviation on the order of 20 to 30 metres has less weight than the best estimate at that time. These contribute to but do not dominate the update process at one instance in time. Thus any non-random error existing in the best estimate does not immediately get diminished which makes sense. Over time, however, it is clear the updates control the error to the 30 to 40 metre level in general.

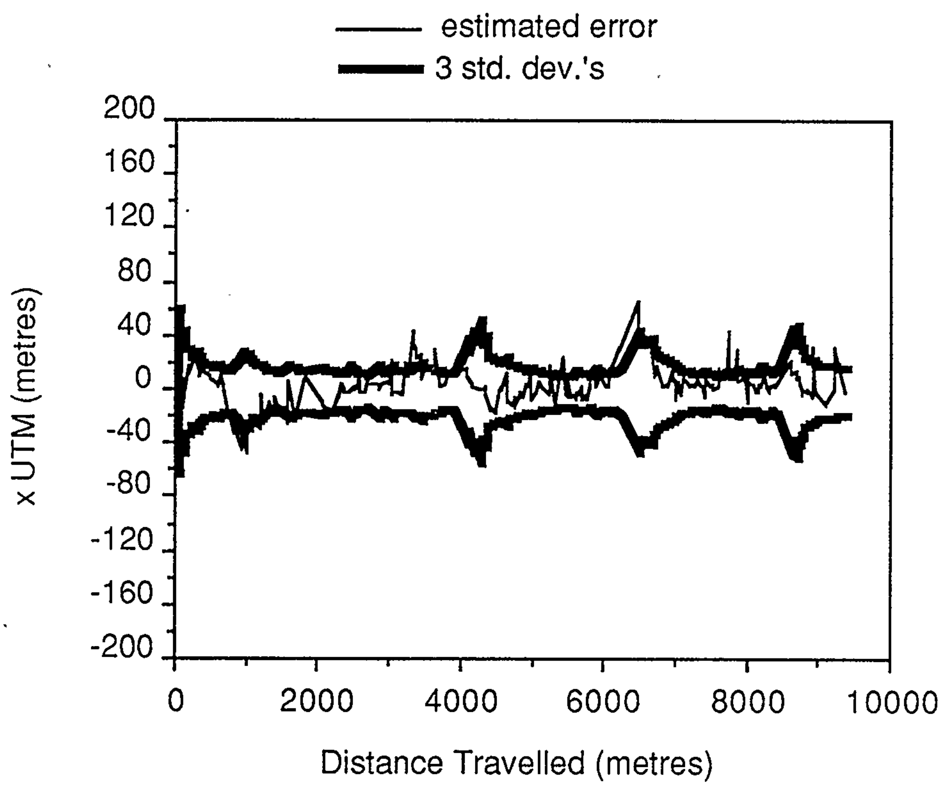


Figure 5.7a - DR, MM, and GPS - x Plot

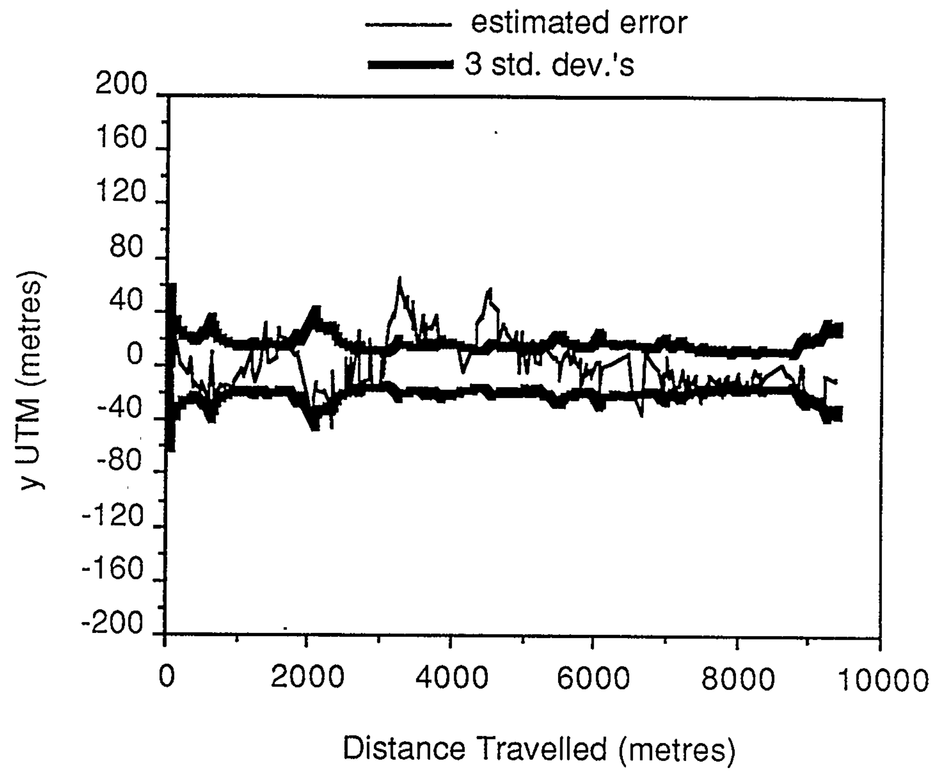


Figure 5.7b - DR, MM, and GPS - y Plot

Figures 5.8a and 5.8b display the kinematic position filter results for the case of GPS updates only. As seen the errors in both x and y plots are controlled to the 40 metre level and are contained within the respective error envelopes. It is noted that without map matched azimuth updates present (i.e. GPS x, y updates only), no systematic affects are present.

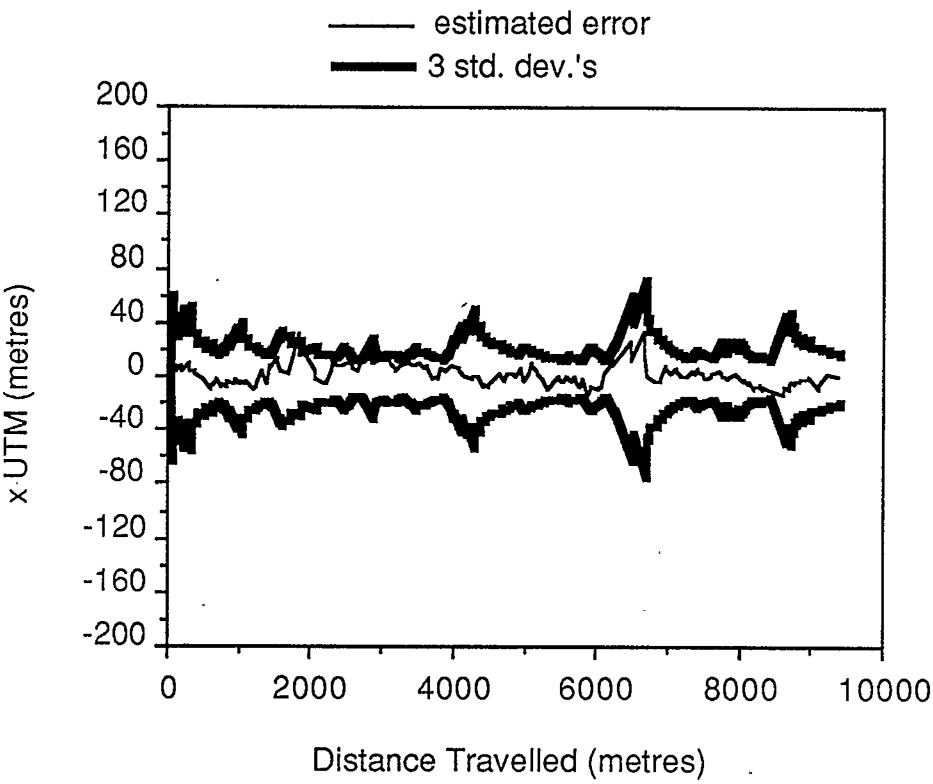


Figure 5.8a - DR and GPS - x Plot

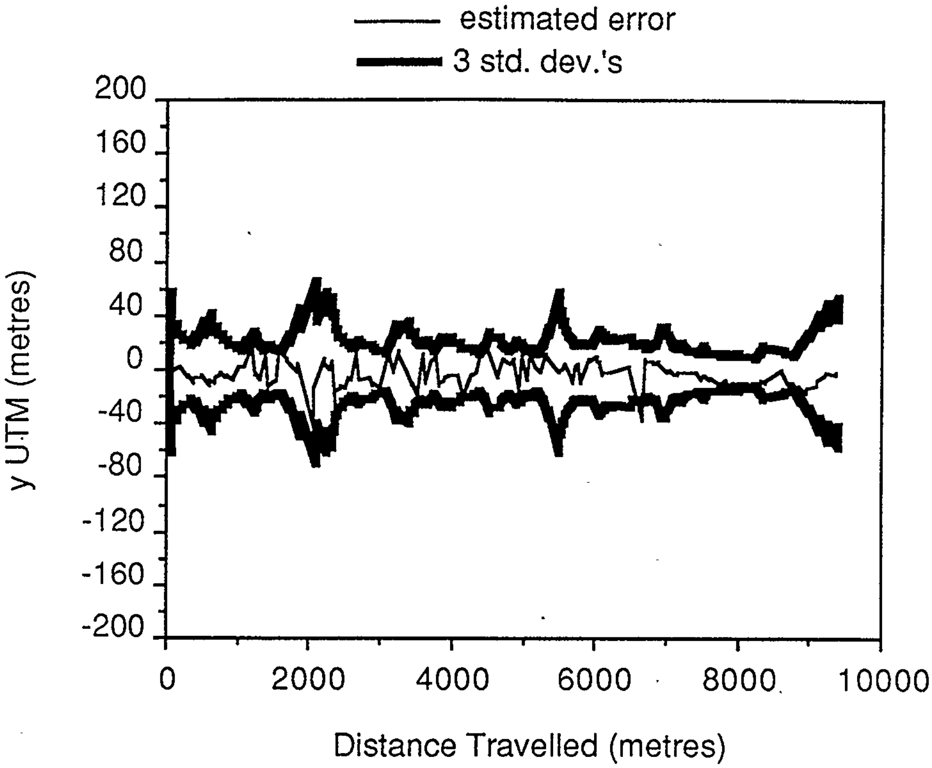


Figure 5.8b - DR and GPS - y Plot

It is suggested that another form of azimuth sensor be introduced to the system in combination with the differential odometry relative azimuth configuration in order to enhance system reliability. An absolute heading sensor such as a magnetic fluxgate compass should be selected. The two systems are likely to compliment each other; the dead reckoned azimuth relatively stable in the short term could spot extreme magnetic compass deviations caused by magnetic anomalies, and the compass could provide a source of stability in the absolute sense, controlling azimuth error accumulation and flagging any odometer blunders which might have gone undetected otherwise.

Improvements in the manufactured odometers should also be considered if one wishes to design a system which can essentially keep track of itself beyond the street segment network and without GPS reception. Firstly, the quantization error in azimuth due to the quantization between distance pulses should be improved. This can be accomplished by reducing spacing between the odometer targets sensing wheel rotation in order to ultimately sense smaller intervals of distance change on both left and right wheels. A one to three degree quantization in azimuth would contribute significantly to reduction of x and y error accumulation due to quantization error and would suppress the impact of a missed pulse substantially. This would require sensors which detect a change in distance of one to two inches. Potential sources of odometer related error which would affect the system's dead reckoning performance are:

1) tire diameter increases due to increased speed (i.e. centrifugal force);
2) tire diameter fluctuations dependent on tire air pressure (e.g. temperature effects);
3) tire slippage including lateral movements;
4) tire diameter reduction, due to long term tire wear;
5) sensor rate frequency error dependent on vehicle velocity (i.e. time quantization);
6) surface to mapping plane scale factor correction (quite small); and,
7) undulating terrain (i.e. no slope reduction).

It has been observed by Lezniak et al. [1977] from odometer tests that errors related to velocity can be as high as 15 metres in a kilometre at 100 Km per hour (78 feet in a mile at 60 MPH) with standard bias ply tires with 5 metres per kilometre at 50 kilometres per hour (20 to 30 feet per mile at 30 MPH). Steel belted radials tires are not so affected demonstrating 2 metres per kilometre at 100 kilometres per hour (12 feet per mile errors at 60 MPH). The test vehicle is equipped with steel belted tires. Methods to correct such errors by applying a linear velocity dependent correction is apparently feasible but items like these must be further addressed when considering systematic noise quantities in the filter and should be considered for future work.

It seems apparent that automatic recalibration of odometers would be an definite asset over the long term for other tire related errors excluding missed pulses, however, with only 30 metre accurate updates, undulating terrain, and irregular driving patterns, changing calibration parameters (at the 10^{-3} level) may be quite difficult to accurately compute and may do more damage than good. Thus scale factors were not modelled into the state vector at this time. It is apparent that with differential odometry, azimuth is subject to drift if one wheel's sensor is poorly calibrated, an additional azimuth error the dead reckoning component can do without. Future research investigating implementation of left and right wheel calibration states into the filter additional to x, y, and α should be carried out, making the filter adaptive in this sense. Caution is advised, however, on variance selection of such parameters which may largely affect the sensitivity of the filter. It would not be surprising to learn that determination of such quantities would be more appropriately done external to the filter itself.

Finally, it is realized that with good map information, additional and new map matching developmental work should be researched. Utilizing the map data cleverly in the dynamic environment will significantly aid in controlling positioning error and further more enhance system integrity. A good example of this would be to use LOP (Line of Position information) of a street upon confident location identification, constraining the vehicle's position within the walls of the road's boundaries (i.e. rectangle with dimensions street segment length by street segment width including map variance). Velocity comparisons between the automobile (derived from odometer distance over time or GPS Doppler measurements) and the map data will contribute tremendously to the confident selection of true road segment location for this update possibility.

5.5 Confidence Ellipse

In Section 4.3: Location Monitoring and Map Matching, the concept of using windows of error to assist in map matching is discussed. Presented is the mathematics involved for computing rigorous confidence regions for the best estimated coordinates $\hat{x}, \hat{y}$ of the vehicle using the upper two by two submatrix of $\mathbf{C}\hat{\delta}^{(+)}_{k+1}$. This portion of $\mathbf{C}\hat{\delta}^{(+)}_{k+1} = \mathbf{C}_{x,y}$ is

$$\mathbf{C}_{x,y} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} . \tag{5.63}$$

Note that for the case of dead reckoning between updates the superscript $^{(+)}$ is a $^{(-)}$ and all of the following ellipse equations are the same.

By determining the eigen values and eigen vector of the above matrix $\mathbf{C}_{x,y}$, $\hat{x}, \hat{y}$ parameters of an error ellipse can be solved for directly. Error ellipses reflect how accurately the point has been positioned [Mepham and Nickerson, 1987]. It is a confidence region. The mathematical equations for the semi-major axis (a), semi-minor axis (b), and the ellipse orientation (θ) (i.e. azimuth of semi-major axis) are as follows [Mikhail, 1976; Mepham, 1983]:

$$a = \left\{ \frac{\sigma_x^2 + \sigma_y^2}{2} + \left[\frac{\sigma_x^2 - \sigma_y^2}{4} + \sigma_{xy}^2 \right]^{1/2} \right\}^{1/2} , \quad (5.64)$$

$$b = \left\{ \frac{\sigma_x^2 + \sigma_y^2}{2} - \left[\frac{\sigma_x^2 - \sigma_y^2}{4} + \sigma_{xy}^2 \right]^{1/2} \right\}^{1/2} , \quad (5.65)$$

$$\theta = \frac{1}{2} \tan^{-1} \left[\frac{-2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2} \right] , \quad (5.66)$$

with a corresponding to σ_{max} and b to σ_{min}.

As these equations are for standard confidence regions they describe a region with a 39 percent probability that the true point lies within the ellipse defined. This is to say that we are 39 percent confident that the estimated position falls within the confinements of the defined elliptical window. A 39 percent level of probability is not suitable for practice, so an expansion factor is multiplied on to the ellipse parameters a and b to increase the probability level of assurance. For two dimensions, the expansion factor is derived from the chi-squared (χ^2) distribution with two degrees of freedom. For 99% confidence regions the expansion factor $\xi\chi^2_{(2)}$ is 3.03 obtained from statistical tables of the χ^2 distribution with two degrees of freedom [Thomson et al. ,1982] and ellipse parameters *a* and *b* are expanded out to

$$a_{99\%} = 3.03\, a\,, \tag{5.67}$$

$$b_{99\%} = 3.03\, b. \tag{5.68}$$

To illustrate the concept, error ellipses pertaining to a few selected points at the beginning of the test run were computed and are superimposed onto the digital map in Figure 5.9 and Figure 5.10. To also show the filter effect upon the error window, Figure 5.9 shows DR only whereas Figure 5.10 shows update ellipses. One can see, with DR only, in certain portions of the network, determination of location by error ellipse will result in ambiguous scenarios. The distance from the centre of the ellipses to their respective MM location gives an indication of true error.

As it is not so practical to work with complex ellipse regions for map matching and location monitoring applications, it is suggested rectangles be used to approximate the regions with dimensions length *a* by width *b* with the rectangle long axis orientated from grid north by the azimuth θ. The ellipse equations above are still used to compute *a* and *b*, the dimensions of the rectangle, but the area of the rectangle approximating the confidence region will be larger than the ellipse itself due to the additional corner extensions. Once computed an additional liner about the window should also be applied, due to map errors, as the ellipses have no regard to the mapped streets underneath them. If not applied, map correlation could conceivably fail.

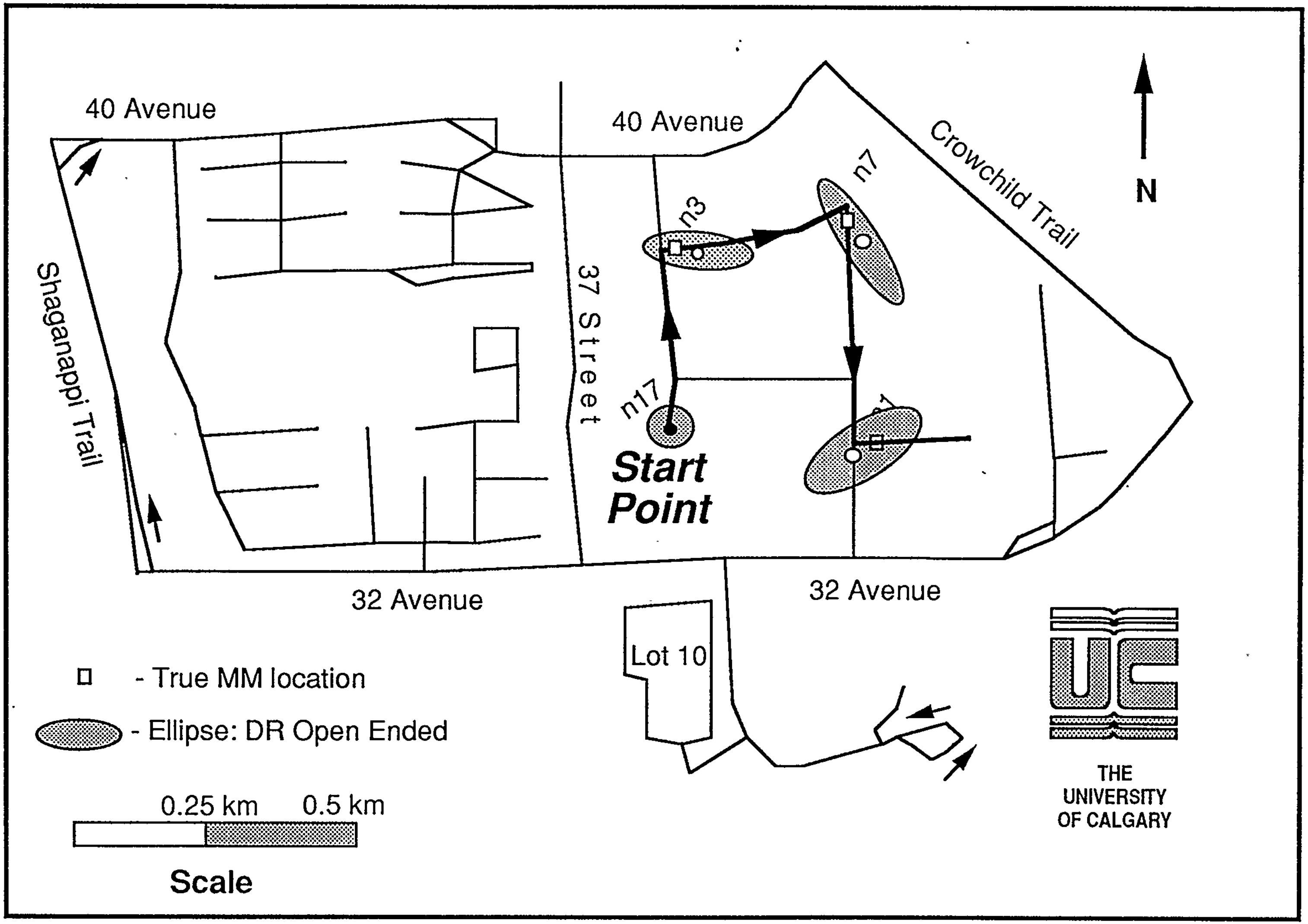


Figure 5.9 - DR Open Ended 99 % Standard Confidence Ellipses

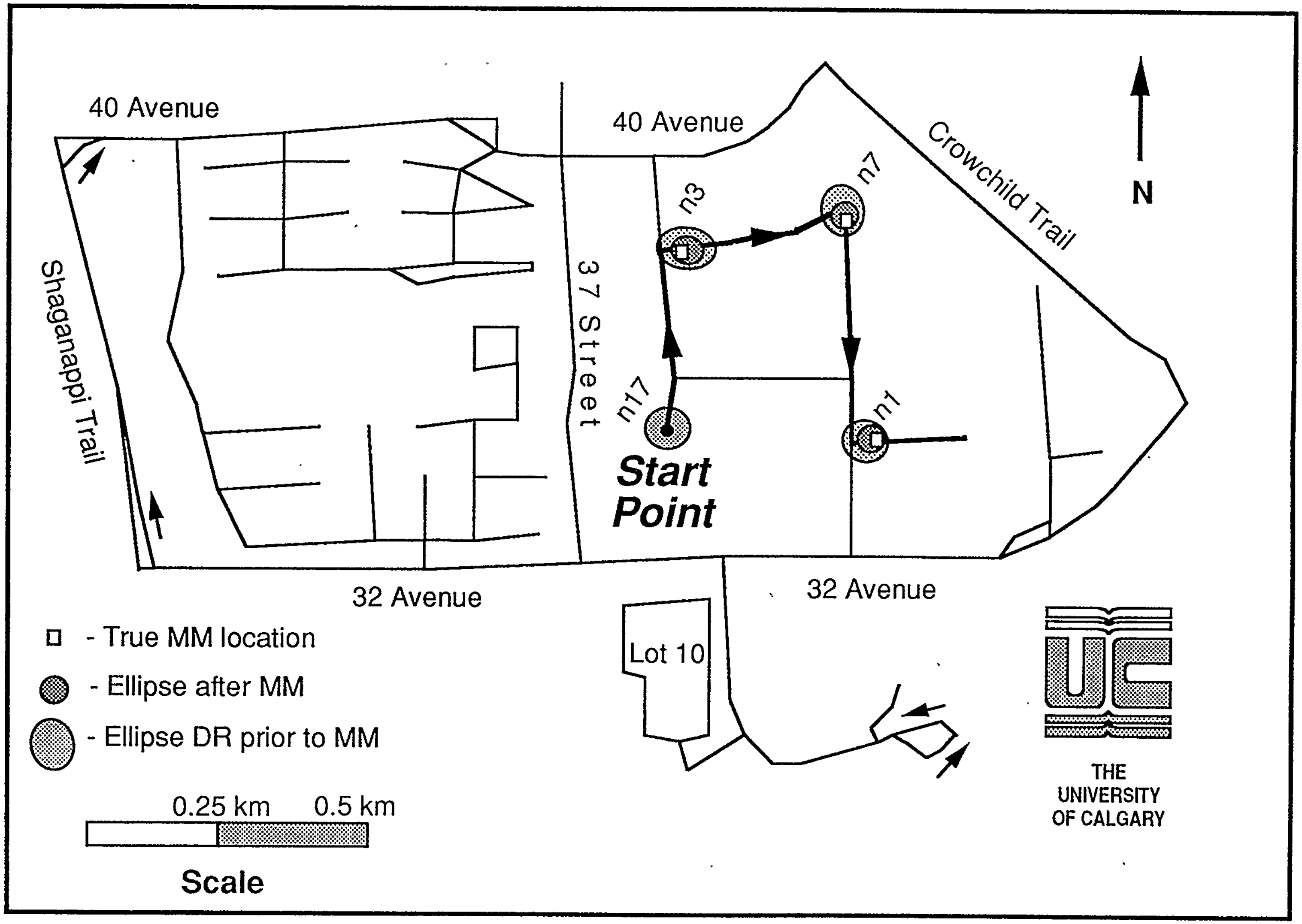


Figure 5.10 - DR plus MM updates 99 % Standard Confidence Ellipses

6. NAVIGATION AND INFORMATION MODULE

6.1 Route Determination

Route determination is that part of the navigation and information module involving optimal route computation between various locations within a street network (e.g. present vehicle location to given destination). This function is comprised only with the underlying mathematics and procedures used to determine the node to node, best route solution through a transportation network.

Many AVLN routing problems can be conceived. For instance, from Dreyfus [1969]:

1) the optimal route between any two prespecified nodes in a network (i.e. source (origin) node *s* to terminating (destination) node *t*) ;
2) the optimal route between *s* and all other nodes in a network;
3) the optimal route between any two prespecified nodes which passes through other predefined nodes (e.g. most scenic touring route);
4) the second, third, fourth, etc., best routes between two nodes; and,
5) the optimal routes between all or many node to node combinations in a network,

to name only a few. Upon review of the list above, it is apparent that many routing problems are conceptually a derivative of the first problem (i.e. one node to one node). It is this problem that the AVLN 2000™ prototype development has focussed upon and considered first.

Optimality involves the solution of an extrema problem (i.e. a maximum or minimum). This brings rise to the question, what is an optimal route? An optimal route can be defined as any route (e.g chain or path) in a given environment (e.g. a network) belonging to a given set of routes which is optimal with respect to a prespecified set of criterion. In the context of AVLN applications, the optimality criterion generally involves a minimization solution, the most obvious example being the shortest travel route by distance between any two points. Other objective functions might include least monetary cost or minimum travel time.

Over the past several decades many algorithms have been developed which solve for the least resistive path according to link weights (e.g. distance, time, cost) through a network. Among these, Dijkstra's algorithm [Dijkstra, 1959] may be the most computationally efficient [Dreyfus, 1969] and [Deo, 1974].

Formally Dijkstra's algorithm has been described by Smith [1982] in the following manner:

"step 0 : Assign a temporary label $\mathbf{L}(i) = \infty$ to all nodes $i \neq s$; set $\mathbf{L}(s) = 0$ and set p = s. Make $\mathbf{L}(s)$ permanent (p is the last node to be given a permanent label.)

step 1 : For each node i with a temporary label, redefine $\mathbf{L}(i)$ to be the smaller of $\mathbf{L}(i)$ and $\mathbf{L}(p) + \mathbf{d}(p,i)$. Find the node i with the smallest temporary label, set p = to this i and make the label $\mathbf{L}(p)$ permanent.

step 2 : If node t has a temporary label, then repeat step 1. Otherwise, t has a permanent label, and this corresponds to the length of the shortest path from s to t through the network. Stop."

The array $\mathbf{d}(p,i)$ is an adjacency matrix element (**AMAT** $[p,i]$) representing the link weight from node p to node i. Extracted from Harris et al. [1987], the following two paragraphs describe the above procedure quite well in the context of shortest path (i.e. minimum travel distance).

"The algorithm works utilizing a permanent and temporary label setting technique as opposed to a label correcting technique. Initially, all nodes (n) in the network (**n**) are assigned a temporary label of ∞ (computationally a large number) except the source node (*s*) and are declared temporary nodes. Each label represents the arc property or weight from *s* to each node. For simplicity we shall refer to these weights as distances but it should be noted that these weights could be representative of other characteristics, such as cost, time and safety, as long as the values are positive.

To begin with, the *s* node is declared permanent and is assigned a label of zero since the minimum distance from *s* to itself is in fact simply that. The algorithm starts at node *s* and by utilizing a travel matrix, compares all distances to nodes adjacent to *s* by one link with their current temporary label. If the distance is less then the corresponding temporary label, it replaces the label. When all these adjacent nodes have been checked the algorithm then chooses the node with the least distance in the temporary label set and assigns that node to be a member of the permanent set indicating that the shortest distance from s to the newly defined permanent node has been determined. The algorithm then repeats itself in the same fashion starting at the newly assigned permanent node. The algorithm roots through the network each iteration continually converting temporary nodes into permanent nodes and is terminated when the *t* node has been labelled permanent."

Dijkstra's algorithm computes the value of least resistance between nodes (e.g. shortest distance) but excludes to retain the node to node route to follow. To obtain this, each permanently labelled node's parent node must be stored during each iteration. This is quite important. Once the algorithm ends a back stepping procedure from *t* to *s* using this parent node information must be applied to capture the node to node list identifying the sequenced optimal path.

One way streets and turn restrictions pose no operational problem to the algorithm as long as the topology database reflects this information accurately. In section 3.4, the network topological data file example displays how one way restrictions are stored at no additional cost to storage. Turn restrictions require remodelling of the intersection or insertion of additional flags indicating certain successor link restrictions of only a portion but not all of the adjacent node set (i.e. conditional one ways).

During algorithm operation, a one way link is assigned an ∞ weight (i.e computationally a very large number relative to the weights in the network) corresponding to the directional restriction. Large enough such that the least resistive path tree could not possibly flow against it.

It seems apparent that the user of an in-car, stand alone AVLN system will more often than not desire best route computations beginning at their vehicle's present location. To increase attractive automatic quality of the system, start location can be provided by the location monitor. If a car is situated on a link between the link's two defining nodes, which represents the majority of the cases, then *s* is taken as the node that the automobile is heading towards and will subsequently drive to first. This demonstrates the importance of retaining not only vehicle location but vehicle heading as well.

Vehicle heading, in light of destination approach, is also important to consider. If a street address is keyed in as a destination input and happen to be a mid-block address, then the two nodes defining the corresponding link must both be considered as destination nodes *t* with the best one selected based on least resistance and approach (i.e. What side of the street is the destination on?).

Computationally, Dijkstra's algorithm in the worst case, requires $O(n^2)$ operations. This quantity is directly related to algorithm run time, a major consideration for an AVLN system, especially from a user's perspective. For the prototype test network (n = 118) Dijkstra's algorithm by itself performs quite well providing node to node routes between any two points in well under a second. For larger networks (e.g. municipal, provincial, federal), however, the number of operations are increased tremendously and thus enhancements to the algorithm are most definitely required to satisfy the AVLN users who must wait for a route.

One way of decreasing computations to certain degree is to utilize an adjacency array list (**ADJ_LIST**) instead of an adjacency matrix (**AMAT**) in the algorithm (incidently another advantage of incorporating adjacency lists into the database). This only applies to sparse networks, which, fortunately includes road transportation networks. As the topological data file intrinsically contains an adjacency list, this method was incorporated into the prototype system's navigation software. Even with this enhancement, Dijkstra's algorithm for very large networks still will not provide route information quickly enough to satisfy the user. Two other enhancement techniques which will aid in reducing run time include a divide and conquer strategy entitled the patch-quilt concept and the application of heuristics.

The patch-quilt concept was developed at The University of Calgary and is presented by Harris et al. [1987]. By dividing a large network by means of geographic polygonization into several smaller, more manageably sized subnetworks, the problem may be conquered by bridging between subnetworks containing source *s* and destination *t* nodes across a higher level hypernetwork. The methodology comprises of: computing best routes using Dijkstra's algorithm between *s* and all gateway (i.e. entrance-exit) nodes of the polygon subnetwork containing *s*; computing best routes using Dijkstra's algorithm between *t* and all gateway nodes of the polygon subnetwork containing *t*; computing best routes using Dijkstra's algorithm between all *s* and *t* polygon gateway nodes through a hypernetwork consisting of only polygon gateway nodes; and, selecting out of all possible combinations one best route.

Deceivingly, the primary advantage of this technique is running Dijkstra's three times as the networks it is operating on are much smaller compared to the original overall network. Published patch-quilt tests [Harris et al., 1987] performed on the Alberta, highway, provincial, transportation network (n = 1500) indicate good potential for AVLN implementation. It was concluded from these studies that assigning major arterials lighter weights than secondary and tertiary highways based on a some function of speed should be investigated and that further developments are required to overcome some shortcomings of the method. Investigations into applying heuristic techniques to enhance the patchquilt method was also recommended.

The application of heuristics implies using additional knowledge which when applied will aid in solving a problem. The goal in many instances is to speed up the time it takes to arrive to a solution compared to strict algorithmic approaches. This of course is of paramount importance considering the patience levels of most potential users.

Often times, heuristics are conceived out of imagining how humans might attempt to solve a particular problem and are usually based on common sense. A good example related to route determination is utilizing the additional information of how the crow flies (i.e. Euclidean distance) between *s* and *t*. Generally, when a human is given the task to identify the best route between two points on a road map, one of the first steps performed is to look at roads which lie between *s* and *t*, neglecting all roads which seem to take you further away from the destination goal.

Dijkstra's algorithm, as it is described above, creates a tree which branches out radially, iteration by iteration, growing each time, branch by branch, extending to the next closest node to *s* and then the next closest and so on. As described above it is strictly algorithmic. In the context of graph search methodology it is considered as a breadth first searching procedure. In a directive sense, if one could narrow the search down to a particular corridor of possibilities which has a high likelihood of containing the optimal route, then the solution could be arrived at much more swiftly. In this context, the general term for this procedure is depth first searching.

In order to enhance Dijkstra's algorithm towards a more depth first rather than breadth first search, the following weight equation may be incorporated [Nilsson, 1980]:

$$f(n) = g(n) + h(n), \tag{6.1}$$

where

$g(n)$ = cost of optimal path from s to n,

$h(n)$ = additional cost from n to goal t based on heuristic.

The evaluation function $f(n)$ represents the least resistive value from node s to node n and is composed of two parts. The first component $g(n)$ is the cost of the path in the search tree (i.e. the tree branches are all the optimal routes from s to all other nodes) from s to n. The second component is a heuristic function, adding additional weight to the temporary label of n allowing for the search to be biased in some fashion (i.e depth first). If $h(n)$ is equal to zero, the algorithm collapses to Dijkstra's algorithm. If $h(n)$ is equal to the Euclidean distance from node n to the goal node t (i.e. 'as the crow flies path'), then the search would be drawn towards the goal node in the network leaning towards a depth first procedure. This is due directly to the fact that Dijkstra's algorithm seeks least resistance and those nodes further away in a Euclidean sense from *t* will have a larger weight relative to those nodes that are closer. Interestingly, by human nature (i.e. heuristics), most of us given the task to find the best route between two locations on a paper road map would first search route possibilities that lie in a general corridor of a crow's flight path to arrive at a quick and near optimal or optimal solution .

The type of application using heuristic functions expressed above is generally known as the A* algorithm (pronounced 'A star') [Nilsson, 1980]. It should be realized that many heuristic weight schemas can be used within the heuristic function h(*n*) according to the application type. For example, links extending out from the last permanently labeled node that have a left turn orientation (obtained from the topology data file according to node approach) may be given a larger weight at that particular moment to aid in finding a route where the number of left hand turns are minimized. One should note as well, two ended searching (i.e. branching out from both the start node *s* and the destination node *t*), depending on the heuristics employed, may yield significantly faster solutions than just branching out from one end. Investigations and developments of the A* algorithm for AVLN path planning are discussed by [Guzolek and Koch, 1989].

Heuristics is a powerful route determination tool, however, in any given application no guarantee of optimality exists when it is applied. In an effort to speed up process time, there becomes a chance that maybe a second or third best route is all that will be computed. Nevertheless, for road transportation networks, the chance of finding the most optimum route is quite good and to many, even a good route will be more than satisfactory.

Many interesting developments can be made from employment of Dijkstra's algorithm and Dijkstra enhanced algorithms (e.g. perhaps A* algorithm combined with the patch-quilt concept). Traffic conditions, for one, are presently being broadcasted digitally via radio links to in-car navigation systems in Europe. The system's ability to know the dynamics of the network such as traffic flow congestion can be transformed into parameters by adding more weight to static weights already contained in the database. This would allow the algorithms to avoid mathematically, ahead of time, particularly unfavorable routes due to temporal, real-time delays, unseen by and unknown to the driver.

Successful AVLN route determination packages utilizing Dijkstra related approaches will require enhancements, outside of those related to run time optimization discussed above, to overcome intrinsic limitations. One, not so obvious limitation characteristic to AVLN operation is that fact that these algorithms compute simple path solutions only. Thus, if circling around the block was required in order to change heading and get onto the the computed optimal path, the algorithm's could not handle this in their present states as their solutions do not include cycles. Numerous and well thought out logic steps are required to produce an appealing and pragmatic optimal route determination product for society.

6.2 Route Guidance

Route guidance is that part of the navigation and information module which provides instructions on how to navigate from the vehicle's present location through the streets and avenues to a specified destination. It requires a vector containing the node to node optimal route as computed by the the route determination function described in the last section.

Development of the AVLN 2000™ prototype's route guidance function involved proof of concept laboratory testing. The software developed simulates vehicle location along a Dijkstra determined optimal route and displays in real-time route instructions providing intersection negotiation advice along with the route which is highlighted bold on the computer map. The underlying concepts are presented below.

Two nodes are selected as start and end nodes (see Figure 6.1) and an optimal route is computed between these using Dijkstra's shortest path algorithm from the route determination function. Once the route is determined, vehicle location is simulated along the chosen path with a velocity of roughly 40 kilometres per hour. As the vehicle proceeds forward , the next intersection en route is identified and a local successor tree of opportunity is formed. This tree is built just as it is in the map matching and location monitor function of the position and location monitor (Chapter 4) which making this methodology a very attractive feature in view of the system's subsystem compatibility. Approach orientation is rectified by knowing the previous node and the approaching node with respect to the vehicle's present location. The next node after the approaching intersection in sequence along the best route node to node series is compared to all successor nodes of the successor tree of possible advancement just. When a positive match is made, a left turn, right turn, or continue forward manoeuvre identification is made and an appropriate navigation message displayed. The street name (retrieved from the auxiliary data file) corresponding to the road which the vehicle should proceed onto is also included in the message. Instructions of the next manoeuvre are provided just after a correct manoeuvre has been completed to allow for ample time for

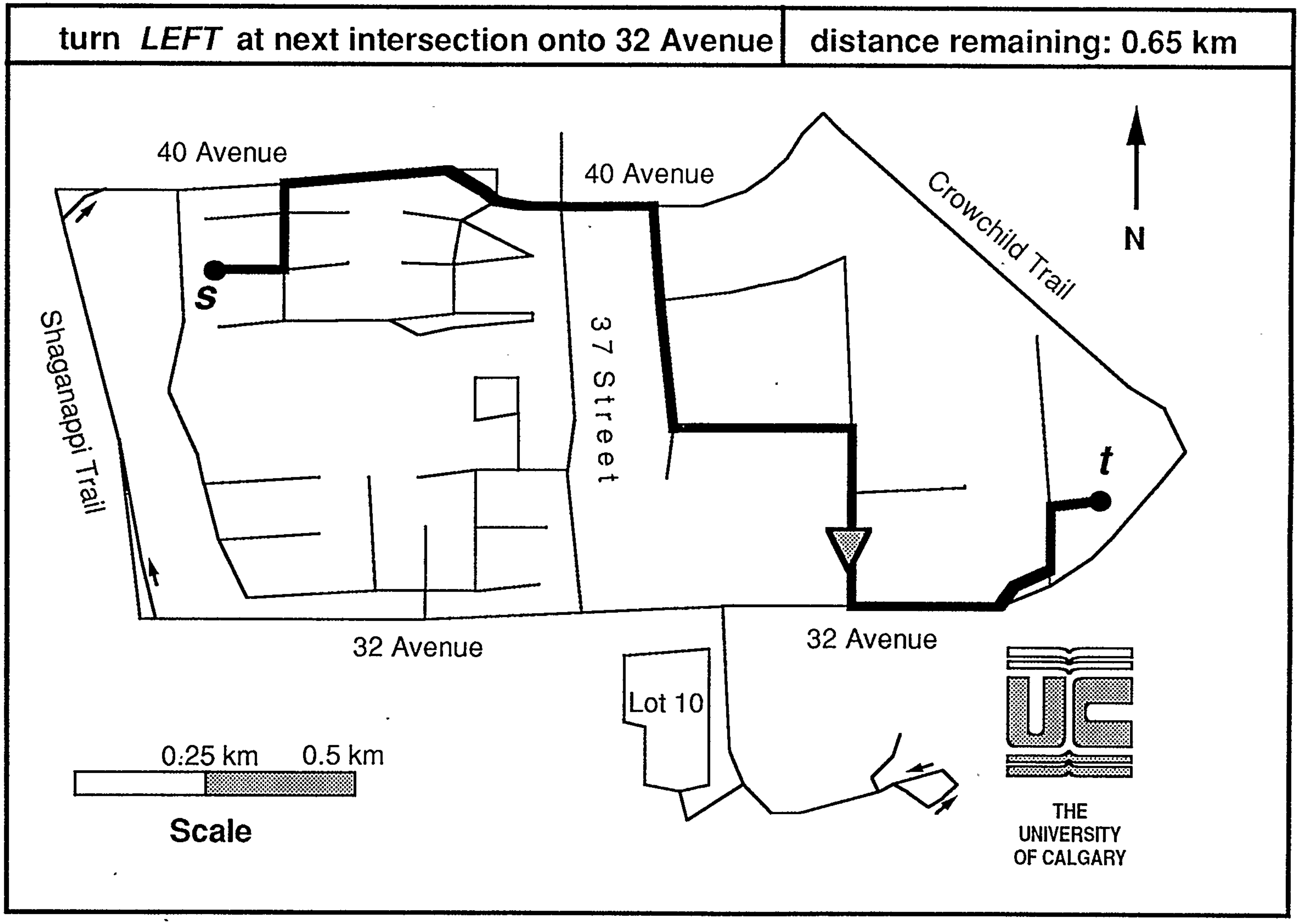


Figure 6.1 - Navigation Display Freeze Frame

required lane changes, etc. Inclusion of street names gives the driver additional information other than just simple left, right and forward instructions. If the auxiliary database contained landmark information, this could also be used to assist in navigation in the same manner that street names aid in intersection recognition .

Illustrated in Figure 6.1 is a freeze frame of the AVLN prototype's navigation feature. The entire optimal route computed by Dijkstra's algorithm with shortest distance as the criterion is highlighted in bold. The arrow icon represents the vehicle's location and heading. The navigation message is also shown at the top corresponding to the next suggested manoeuvre accompanied by the distance remaining until destination arrival is displayed. This is computed by subtracting the distance journeyed up to the present time from the entire length of the shortest path. Note that if speed limits were contained in the data files, ETA (Expected Time of Arrival) could be estimated by dividing distances of appropriate street segments along the route by the corresponding permissible maximum speed limits and presented to the user in real-time.

One issue regarding output presentation of route guidance facilities in vehicles concerns safety. While driving, it is quite important that visual distractions be minimized in order to allow the driver to concentrate on the road and driving responsibilities. If a AVLN is required to read the computer monitor for instructions all the time, accidents will inevitably occur. Not only will this lead to injury, but undoubtedly litigation. To enhance safety but maintain navigation, voice synthesization [Mark and McGranaghan,1986; and, Davis and Schmandt, 1989] can possibly replace or better yet compliment visual displays used for navigation. Great work still remains concerning the underlying and often overlooked MMI (Man Machine Interface) issues of AVLN systems. No matter how one wishes to present the information to the user wether it be by LCD character display, digital map, voice synthesization, or even H.U.D.s (Head Up Displays) , the methodology presented above may be used regardless of the output medium to supply real-time, intersection negotiation advice.

6.3 Auxiliary Information Provision

The auxiliary information provision function was designed into the system architecture to account for all the intrinsic, potential, and beneficial uses of an AVLN system's database apart from the system's regular functions. The list is conceivably endless and additional features will continuously be developed in response to user needs, once first generation systems have entered the market place. Truly auxiliary information, from the user's perspective is invaluable and the more there is the greater value AVLN systems will be to society.

The most obvious feature which user's will benefit from is the replacement of in-car paper maps with dashboard displayed, digital atlases. Information of hotels, campgrounds, restaurants, communities, shopping malls, gas stations, recreational centres, museums, you name it, in the local vicinity or even in outside regions at various scales may be displayed with the touch of a few buttons. Anything the standard automobile paper map is used for, will essentially be upgraded to a computer based map system hosting a multitude of events.

Years of GIS and ergonomic development will have to take place before such technological features will enter the automobile markets. Observing the present push however by major automobile suppliers and manufacturers around the world in AVLN developments, the number of years required may be less than anticipated.

7. CONCLUSIONS AND RECOMMENDATIONS

The engineering project was successful in that the objectives laid out were met. A stand alone, Automatic Vehicle Location and Navigation (AVLN) system for land-based, on-road applications was designed and a working prototype system was developed. The system was built for proof of concept, demonstration, and testing purposes only.

The system's hardware component consisted of off the shelf packaged existing components which were assembled together. It is realized that more electronic and hardware enhancements are required before a production level prototype system can commence. A modular approach in the design strategy was taken and this should be carried forward into all future generation designs.

A working digital map database was constructed and it was determined that adjacency lists were the most appropriate means of digital network representation for AVLN applications. Topology of a digital road map is fundamental to any AVLN system and supports map matching, location monitoring, route determination, and route guidance. Unfortunately, topological information does not exist in most existing digital map GIS road databases but is of paramount importance.

The domain of the test region digital map database involved a community size network. It is recommended that future work should address the foreseen problems associated with development of city wide and further nation wide road transportation networks. Also AVLN map attribute selection should be seriously addressed to guide the digital mapping industry to make correct decisions on what types of information should be included in developing efforts as mapping the nation is no small task and turning back to redigitize valuable, missing information would undoubtedly be an extremely costly venture . Industry standards in mapping must also be decided upon shortly in order to speed up the process of AVLN implementation across the nation and reduce redundant, non-conforming mapping efforts.

The underlying principles of positioning a vehicle by differential odometer dead reckoning and also by GPS were presented and found to be feasible. Map matching and location monitoring on the other hand was not as straight forward. A method was conceived and implemented, however, more logic intensive functions must be devised. The importance of the location monitor function is not well recognized in general but deserves a tremendous amount of attention for the following reasons:

1) map matched updates are essential to control dead reckoned error accumulation when GPS is not available due to blockage;
2) location and not position is the bridge between the vehicle and the digital map database;
3) users readily identify with location not position; and,
4) route determination and guidance functions depend upon continuous maintenance of vehicle location.

The success of future systems and their acceptance in the market place will depend on the ability and level of sophistication of internal location monitor facilities. Further research and development emphasis should be directed to this end, namely increasing the level of sophistication of the location monitor's facilities (i.e. more logic intensive).

The kinematic position filter consists of two components, a dead reckoning component and an absolute point update component (i.e. position and azimuth). The dead reckoning component is a special case of the prediction model of a Kalman filter. It is distinguished from a standard text book survey traverse model in that correlation prevails between the change in distance $\overline{\Delta d}$ and the change in azimuth $\Delta\alpha$ due to the fact they are both functions of Δd^L and Δd^R.

This filter can be easily and quickly modified to accommodate any new update measurement (e.g. signpost, LORAN-C) outside of map matching and GPS positioning as updates are introduced at the coordinate and azimuth level. GPS pseudo-range observations are not directly used in the filter's mathematics and no need for this is presently apparent as all GPS computations (including future differential GPS possibilities) can be performed externally from the filter. The use of pseudo-range models directly embedded into the filter instead of at the map coordinate level should be evaluated further to identify potential system benefits such as statistically testing for blunders in pseudo-ranges observations.

Analysis of test results revealed that in the open ended dead reckoning with differential odometry accuracy on the order 1 to 2 % of the distance travelled can be achieved. With the introduction of digital map azimuth updates, marginal reduction of the error was observed over the ten kilometre test run. Systematic affects resulting from a few large map matched azimuth misclosures prevails, causing to this marginal difference. The affects of introducing MM and GPS coordinate updates was quite good, maintaining the accuracy of the system's position to the 40 metre level. This accuracy estimate corresponds to the average size of 99 % confidence regions in two dimensions. This value is consistent with the bandwidth within which the estimated errors of position lie.

It is recommended that an absolute azimuth sensor be introduced as an addition to the system, to enhance system reliability and control azimuth error growth over longer runs. Also the differential odometer configuration used should be replaced with one having much higher resolution to reduce quantization error especially in $\Delta\alpha$ (i.e. change in azimuth). More extensive odometer investigations are required to better understand sources of odometer error.

Reliability measures to supplement the accuracy estimate given in covariance computations, should also be researched. This could enhance quantifying the integrity of the system.

Route determination and guidance functions were developed in light of AVLN systems requirements. The development of these were tied directly to the system's digital map database which accommodates many other digital map dependent functions housed within the system.

Route determination functions are developing for AVLN applications and further investigations and developments concerning run time reduction by network partitioning and heuristic methodologies is still required. It is recommended that the navigation and information module's functions be enhanced to include many vehicle related network scenarios that must be addressed to satisfy user needs and requirements. Also this area seems to be a natural place to apply and explore modern MMI (Man Machine Interface) techniques, an essential and important component of such systems. The possibility of introducing knowledge based approaches where rules and facts developed for system specific operations should also be investigated at the implementation stage of the next generation.

REFERENCES

Ahn, J., and Freeman, H. [1984]. A Program for Automatic Name Placement. Cartigraphica. Vol. 21, No. 2 and 3, Summer and Autumn, pp. 101 to 109.

Appleyard, S.F., Linford, R.S., and Yarwood, P.J. [1980]. Marine Electronic Navigation. Routeledge and Kegan Paul, New York, New York, U.S.A.

Bedini, S.A. [1988]. Collector's Corner. Professional Surveyor. American Surveyors Publishing Company Inc., Falls Church, Virginia, U.S.A., May/June, pp. 36 and 60.

Bedini, S.A., and de Solla Price, R.J. [1967]. Technology in Western Civilization. (editors: Kranzberg, M. and Pursell, C.W. Jr.), Vol. 1, Chapter 11: Instrumentation, Oxford University Press, London, England.

Brown, R.G. [1983]. Introduction to Random Signal Analysis and Kalman Filtering. John Wiley & Sons, New York, New York, U.S.A.

Case, E.R. [1986]. Overview of Automobile Navigation Systems and Potential Applications. Proceedings of District 6 and 7 Joint Annual Meeting, Institute of Transportation Engineers (ITE). Vancouver, British Columbia, Canada, July 27-31, pp. 7-16 to 7-25.

Cannon, M.E. [1987]. Kinematic Positioning Using GPS Pseudorange and Carrier Phase Observations. M. Sc. Thesis, Department of Surveying Engineering, The University of Calgary, Calgary, Alberta, Canada.

Chapman, M.A. [1988]. Le traitement des modeles numerique de l'elevation d'une forte densite du GPM (Processing GPM Dense Digital Elevation Models), Ph.D. dissertation Universitè Laval, 171 p.

CONTEC IBM-PC/XT/AT Add-On Board CNT16-4M(PC) 16-Bit 4-Channel UP/DOWN Counter User's Manual. [1986]. CONTEC Microelectronic U.S.A., 3000 Scott Boulevard, Suite 211, Santa Clara, California, 95054, U.S.A., Factory Lab Automation Series.

Cooke, D.F. [1985]. Vehicle Navigation Appliances. Proceedings on Digital Representation of Spatial Knowledge, AUTO - CARTO 7. Washington, D.C., U.S.A., March 11 - 14.

Covil, J.L., Cooney, T.A., and Martin, P.C. [1987]. New Uses of Automatic Vehicle Identification Systems in Highway Transportation. Institute of Transportation Engineers (ITE), 57th Annual Meeting. U.S.A., pp. 209 to 213.

Davis, J.R., and Schmandt, C.M. [1989]. The Back Seat Driver: Real Time Spoken Driving Instructions. Conference Record of Papers Presented at the First Vehicle Navigation and Information Systems Conference (VNIS '89). Toronto, Ontario, Canada, September 11 - 13, pp. 146 to 150.

Deo, Narsingh [1974]. Graph Theory with Applications to Engineering and Computer Science. Prentice Hall, Inc., New Jersey, U.S.A.

Dial, R., Glover, F., Karney, D., and Kingman, D. [1979]. A Computational Analysis of Alternative Algorithms and Labeling Techniques for Finding Shortest Path Trees. Networks. Vol. 9, John Wiley and Sons, Inc.

Dijkstra, E.W. [1959]. A Note On Two Problems on Connexion with Graphs. Numerische Mathematik 1. p. 269 - 271.

Dork, R.A. [1986]. Satellite Navigation Systems for Land Vehicles. I.E.E.E. Position Location And Navigation Symposium (PLANS '86). Las Vegas, Nevada, U.S.A., November 4 - 7, pp. 414 to 418.

Dreyfus, Stuart E. [1969]. An Appraisal on Some Shortest-Path Algorithms. Operations Research and Bulletin. Vol. 17, Number 3, pp. 395 to 412.

Feuchtwanger, M., and Lodwick, G.D. [1986]. Position-Based Land Information Systems. Division of Surveying Engineering, The University of Calgary, Calgary, Alberta, Canada. (January).

French, R.L. [1986a]. Historical Overview of Automobile Navigation Technology. Proceedings of the 36th I.E.E.E. Vehicular Technology Conference. Dallas, Texas, U.S.A., May 20 - 22, pp. 1 to 9.

French, R.L. [1986b]. Automobile Navigation: Where is it Going? I.E.E.E. Position Location and Navigation Symposium (PLANS' 86). Las Vegas, Nevada, U.S.A., November 4 - 7, pp. 406 to 413.

French, R.L. [1987a]. Automobile Navigation in the Past, Present, and Future. Eighth International Symposium on Computed-Assisted Cartography, AUTO-CARTO 8. Baltimore, Maryland, U.S.A., March 29 - April 3, pp. 542 to 551.

French, R.L. [1987b]. The Evolving Role of Vehicular Navigation. Navigation. Journal of the Institute of Navigation, Vol. 34, No. 3, Fall, U.S.A., pp. 212 to 228.

French, R.L. [1989]. Map Matching Origins, Approaches and Applications. Second International Symposium on Land Vehicle Navigation. Munster, Federal Republic of Germany, July 4 - 7, pp. 91 to 116.

Fuentes, LT., and Adeste F. [1987]. LORAN-C in the 21st Century. I.E.E.E. AES (Automotive Engineering Society) Magazine. December, pp. 8 to 10.

Gelb, A., editor, [1974]. Applied Optimal Estimation. written by the technical staff, The Analytic Sciences Corporation, M.I.T. Press, Cambridge, Mass., U.S.A.

Glick, B.J., and Johnson, T.R. [1984]. Cartographic Database Requirements for Land Vehicle Navigation. I.E.E.E. Position Location and Navigation Symposium (PLANS'84). San Diego, California, U.S.A., November, pp. 25-31.

GRiDCase EXP Owner's Guide [1987]. GRiDSystems Corporation 47211 Lakeview Blvd. P.O. Box 5003, Fremont, California, 94537-5003, U.S.A., October.

GRiDCase Technical Reference Manual. [1985]. GRidSystem Corporation, 2535 Garcia Avenue, Mountain View, California, 94043, U.S.A., Order number 29500-50, May.

Guzolek, J., and Koch, E. [1989]. Real-Time Route Planning in Road Networks. Conference Record of Papers Presented at the First Vehicle Navigation and Information Systems Conference (VNIS '89). Toronto, Ontario, Canada, September 11 - 13, pp. 165 to 169.

Harris, C.B., Goss, R., Krakiwsky, E.J., and Karimi, H.A. [1987]. Optimal Route Information System - Automatic Vehicle Location (AVL) Route Determination. Proceedings of the Urban Regional Information Systems Association (URISA). Calgary, Alberta, Canada, November 30 - December 1, pp. 29 to 41.

Harris, C.B., Klesh, L.A., Krakiwsky, E.J., Karimi, H.A., and Lee, N.S.T. [1988]. Digital Map Dependent Functions of Automatic Vehicle Location Systems. I.E.E.E. Position Location And Navigation Symposium (PLANS '88). Orlando, Florida, U.S.A., November 29 - December 2, pp. 79 to 87.

Hart, I.B. [1925]. The Mechanical Investigations of Leonardo DaVinci. Chapman and Hall Ltd., London, England.

Hayes, Capt. C. William, NOAA Ret. [1986]. Digital Cartographic Data, A Case for Electronic Delivery. I.E.E.E. Position Location and Navigation Symposium (PLANS'86). Las Vegas, Nevada, U.S.A., November 4-7, pp. 316-320.

Haywood, P. [1986]. The Differing Requirements of Producers and Users and the Design of Topographic/Cartographic Databases. The Cartographic Journal. Vol. 23, December. pp. 131-134.

Heiskanen, W.A., and Moritz, H. [1987]. Physical Geodesy. Institute of Physical Geodesy Technical University, Graz, Austria. (Corrected reprint of original published by W.H. Freeman and Company, San Francisco, 1967).

Hillier, V.A.W., and Pittuck, F.W. [1975]. Fundamentals of Motor Vehicle Technology. Hutchinson Educational, London, England.

Honey, S.K., White, M.S., and Zavoli, W.B. [1986]. Extending Low Cost Land Navigation into Systems Information Distribution and Control. I.E.E.E. Position Location And Navigation Symposium (PLANS '86). Las Vegas, Nevada, U.S.A., November 4-7, pp. 439 to 444.

Honey, S.K., and Zavoli, W.B. [1985]. A Novel Approach to Automotive Navigation and Map Display. RIN Conference Proceedings - Land Navigation and Location for Mobile Applications. York, England.

Honey, S.K., Zavoli, W.B., Milnes, K.A., Phillips, A.C., White, M.S. Jr., and Loughmiller, G.E. Jr. [1989]. Vehicle Navigation System and Method. United States Patent, Patent No. 4,796,191, January 3.

Ikeda, M., Fukaya, H., and Tabe, T. [1987]. New Navigation System, MAPIX-III, Proceedings. 16th International Symposium on Automobile Technology and Automation. Vol. 2, Florence, Italy, May 11 - 15, pp. 61 to 80.

Kalman, R.E. [1960]. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering. Transaction of the ASME, 82D. s. 35 to 45.

Karimi, H.A., and Krakiwsky, E.J. [1988a]. Intelligent Kinematic Route Guidance Systems. Canadian Conference on Electrical and Computer Engineering. Vancouver, British Columbia, Canada, Nov. 3 - 4, pp. 405 to 408.

Karimi, H.A., and Krakiwsky, E. J. [1988b]. Design Concepts for Knowledge-Based Route Guidance Systems. I.E.E.E. Position Location And Navigation Symposium (PLANS '88). Orlando, Florida, U.S.A., November 29 - December 2, pp. 95 to 103.

Karimi, H.A. [in prep.]. Design of Intelligent Route Guidance Systems for Land Based Applications. Ph.D. Thesis, Department of Surveying Engineering, The University of Calgary, Calgary, Alberta, Canada.

Klesh, Laurie A. [1989]. Design and Implementation of Automatic Dispatch Vehicle Information and Location Systems. M.Sc. Thesis, Department of Surveying Engineering, The University of Calgary, Calgary, Alberta, Canada.

Krakiwsky, E.J., and Wells, D.E. [1971]. Coordinate Systems in Geodesy. The Department of Surveying Engineering, The University of New Brunswick, Fredricton, New Brunswick, May. (Reprinted with additional corrections August 1973).

Krakiwsky, E.J. [1973]. Conformal Map Projections in Geodesy. Lecture Notes No. 37, Department of Surveying Engineering; University of New Brunswick, Fredericton, New Brunswick, Canada, September.

Krakiwsky, E.J., Thomson, D.B., and Steeves, R.R. [1977]. A Manual for Geodetic Coordinate Transformations in the Maritimes. Department of Surveying Engineering, University of New Brunswick, Fredericton, New Brunswick, Canada.

Krakiwsky, E.J., and Gagnon, P. [1987]. Least Squares Adjustment. Papers for the CISM Adjustment and Analysis Seminars. The Canadian Institute of Surveying and Mapping (CISM), Second Edition, January, pp. 108 to 149.

Krakiwsky, E.J., Karimi, H.A., Harris, C., and George, J. [1987a]. Research into Electronic Maps and Automatic Vehicle Location. Eighth International Symposium on Computer-Assisted Cartography, AUTO-CARTO 8. Baltimore, Maryland, U.S.A., March 29 - April 3.

Krakiwsky, E.J., George, J., Karimi, H.A., and Harris, C. [1987b]. An Analysis of Land Based Automatic Vehicle Location (AVL) Systems. Invited Paper to the Engineering Institute of Canada Centennial Conference. Montreal, Quebec, Canada, May 22.

Krakiwsky, E.J., Harris, C.B., and Wong, R.V.C. [1988]. A Kalman Filter for Integration of Dead Reckoning, Map Matching, and GPS Positioning. Proceedings of I.E.E.E. Position Location and Navigation Symposium (PLANS '88). Orlando, Florida, U.S.A., November 29 - December 2, pp. 39 to 46.

Krakiwsky, E.J. [1989]. The Method of Least Squares: A Synthesis of Advances. Department of Surveying Engineering, The University of Calgary, Calgary, Alberta, Canada. Retyped - reprinted version of May 1985 original.

Lachapelle, G., Falkenberg, W., and Casey, M. [1986]. Use of Phase Data for Accurate Differential GPS Kinematic Positioning. IEEE Position Location And Navigation Symposium (PLANS '86). Las Vegas, Nevada, U.S.A., November 4 - 7, pp. 393 to 398.

Lezniak, T.W., Lewis, R.W., and McMillen, R.A., [1977]. A Dead Reckoning/Map Correlation System for Automatic Vehicle Tracking. IEEE Transactions on Vehicular Technology. February, VT-26, No. 1, pp. 47 to 60.

Lee, N.S.T., Karimi, H. A., and Krakiwsky, E.J. [1989]. Road Information Systems: Impact of Geographic Information Systems Technology to Automatic Vehicle Navigation and Guidance. Conference Record of Papers Presented at the First Vehicle Navigation and Information Systems Conference (VNIS '89). Toronto, Ontario, Canada, September 11 - 13, pp. 347 to 352.

Li, L., Deecker, G., Yurach, K., and Seguin, J. [1989]. Updating Urban Street Network Files with High Resolution Satellite Imagery. Ninth International Symposium an Computer-Assisted Cartography, AUTO-CARTO 9. Baltimore, Maryland, U.S.A., April 2 - 7, pp. 21 to 30.

Mark, D.M., and McGranaghan, M. [1986]. Effective Provision of Navigation Assistance to Drivers: A Cognitive Science Approach. Proceedings AUTO-CARTO London. England, pp. 399 to 408.

Mepham, M.P. [1983]. Computer Aided Survey Network Design. M.Sc. Thesis, Division of Surveying Engineering, The University of Calgary, Calgary, Alberta, Canada.

Mepham, M.P., and Nickerson, B.G. [1987]. Preanalysis. Papers for the CISM Adjustment and Analysis Seminars. The Canadian Institute of Surveying and Mapping (CISM), Second Edition, January, pp. 150 to 181.

Mikhail, E.M. [1976]. Observations and Least Squares. University Press of America. Lanham, Maryland, U.S.A. (with contributions by F. Ackerman).

Monmonier, Mark S. [1982]. Computer Assisted Cartography: Principles and Prospects. Englewood Cliffs, New Jersey, Prentice Hall Inc., pp. 214.

Needham, J. [1965]. Science and Civilisation in China. Volume 4: Physics and Physical Technology, Part II: Mechanical Engineering, Cambridge at the University Press, London, England.

Neukirchner, E.P., and Zechnall, W. [1986]. Digital Map Databases for Autonomous Navigation Systems. I.E.E.E. Position Location and Navigation Symposium (PLANS' 86). Las Vegas, Nevada, U.S.A., November 4 - 7, pp. 320 to 324.

Newton, K., Steeds, W., and Garrett, T.K. [1983]. The Motor Vehicle. Butterworths, England.

Nilsson, N.J. [1980]. Principals of Artificial Intelligence. Palo Alto, California, Tioga Publishing Co.

NU-METRICS Roadstar Distance Measuring Instruments Operating Manual. [1987]. NU-METRICS Inc., 947 Longdale Avenue, Longwood, Florida 32750, U.S.A.

Petrov, D. [1976]. Study of Land Navigation in the North in Both Vehicular and Manpack Applications. (technical editor: Lise Vaillancourt), Phase II - Navigation Systems Study, Vol.1, Chapter 1.2.1: Dead Reckoning Systems, Bendix Aviation Electric Ltd., AETR 36587, June 30.

Price, D.J. [1957]. A History of Technology. editors: Singer, C., Holmyard, E.J., Hall, A.R., and Williams, T.I.) Vol. III: From the Renaissance to the Industrial Revolution, Chapter 23: The Manufacture of Scientific Instruments From c. 1500 to c. 1700, Oxford University Press, London, England.

Schwarz, K.P. [1986]. ENSU 561 Lecture Notes (Inertial Positioning). Department of Surveying Engineering, The University of Calgary, Calgary, Alberta, Canada.

Schwarz, K.P. [1987]. Kalman Filtering and Optimal Smoothing. Papers for the CISM Adjustment and Analysis Seminars. The Canadian Institute of Surveying and Mapping (CISM), Second Edition, January, pp. 230 to 264.

Sleeswyk, A.W. [1981]. Vitruvius' Odometer. Scientific American. October, pp. 188-200, Scientific American Inc., New York, New York, U.S.A.

Skomal, E.N. [1981]. Automatic Vehicle Location Systems. Litton Educational Publishing Inc., U.S.A.

Smith, David K. [1982]. Network Optimisation Practice: A Computational Guide. Ellis Horwood Ltd., Halsted Press.

Sonneburg, G.J. [1978]. Radar and Electronic Navigation. Butterworth Scientific, Toronto, Ontario, Canada.

Strandh, S. [1979]. A History of the Machine. A & W Publishers, Inc., New York, New York, U.S.A.

Sugie, M., Menzilcioglu, O., and Kung, H.T. [1984]. CARGuide - On Board Computer for Automobile Route Guidance. AFIPS Conference Proceedings. July.

TANS (Trimble Advanced Navigation Sensor) GPS Receiver Specification and User's Manual. [1988]. Trimble Navigation Ltd., 585 North Mary Avenue, Sunnyvale, California, 94086, U.S.A., March.

TANS (Trimble Advanced Navigation Sensor) System Definition and Program, User's Guide [1988]. Trimble Navigation Ltd. 585 North Mary Avenue, Sunnyvale, California, 94086, U.S.A., March.

Thomas, D.B. [1952]. Conformal Projections in Geodesy and Cartography. U.S. Geodetic Survey Special Publication No. 251.

Thomson, D.B., Krakiwsky, E.J., and Nickerson, B.G. [1982]. A Manual for the Establishment and Assessment of Horizontal Survey Networks. Division of Surveying Engineering, The University of Calgary, Calgary, Alberta, Canada. August, Publication 10005.

Thoone, M.L.G., and Bruekers, R.M. [1987]. A Car Information and Navigation System. Phillips Technical Review. Vol. 43, No. 11/12, pp. 317 - 329.

Van Dierendonck, A.J., Russell, S.S., Kopitzke, E.R., and Birnbaum, M. [1980]. "The GPS Navigation Message". Global Positioning System, Papers Published in *Navigation*,. Vol. I, The Institute of Navigation, Washington, D.C., U.S.A., pp. 55 - 73.

Vanicek, P. and Krakiwsky, E.J. [1986]. Geodesy: The Concepts. 2nd edition, Elsevier Science Publishing Company Inc., New York, U.S.A.

Vehicular Navigation and Location Systems Technology (VNLST): Course 221. [1986]. Navigation Technology Seminars Inc., presented at The Capital Hilton Hotel, Washington D.C. (Navigation Technology Seminars, Inc., 6134 Beachway Drive, Falls Church, Virginia, 22041, U.S.A.), November 17 - 18.

Walker, W.J., and Usher, W.D. [1988]. The Impact of the Redefinition and Readjustment of the North American Datum on Land-Related Information Systems in Alberta. The NAD '83 Redefinition in Canada and its Impact on the Industry. Calgary, Alberta, Canada, April 27, 1988.

Wells, D., Beck, N., Delikaraoglou, D., Kleusberg, A., Krakiwsky, E.J., Lachapelle, G., Langely, R.B., Nakiboglu, M., Schwarz, K.P., Tranquilla, J.M., and Vanicek, P. [1987]. Guide to GPS Positioning. (Canadian GPS Associates) University of New Brunswick Graphic Services, Fredricton, New Brunswick, Canada.

White, M. [1987]. Digital Map Requirements of Vehicle Navigation. Eighth International Symposium on Computer-Assisted Cartography, AUTO-CARTO 8. Baltimore, Maryland, U.S.A., March 29 - April 3, pp. 552 to 561.