TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Pavel Davidson

Algorithms for Autonomous Personal Navigation Systems

Julkaisu 1171 • Publication 1171

Tampere 2013

Tampereen teknillinen yliopisto. Julkaisu 1171
Tampere University of Technology. Publication 1171

Pavel Davidson

Algorithms for Autonomous Personal Navigation Systems

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 15th of November 2013, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2013

ISBN 978-952-15-3174-3 (printed)
ISBN 978-952-15-3215-3 (PDF)
ISSN 1459-2045

ABSTRACT

Personal positioning is a challenging topic in the area of navigation mainly because of the cost, size and power consumption constraints imposed on the hardware. Satellite based positioning techniques can meet the requirements for many applications, but cover well only outdoor environment. Problems like weak satellite signals make the positioning impossible indoors. Urban canyons are also difficult areas for GNSS based navigation because of large multipath errors and satellite signal outages. Many applications require seamless positioning in all environments. However, there is no overall solution for navigation in GNSS denied environment, which is reliable, accurate, cost effective and quickly installed. Recently developed systems for indoor positioning often require pre-installed infrastructure.

Another approach is to use fully autonomous navigation systems based on self-contained sensors and street or indoor maps. This thesis is concerned with autonomous personal navigation devices, which do not rely on the reception of external information, like satellite or terrestrial signals. The three proposed algorithms can be integrated into personal navigation systems.

The first algorithm computes positioning for a map aided navigation system designed for land vehicles traveling on road network. The novelty is in application of particle filtering to vehicle navigation using road network database. The second algorithm is aimed at map aided vehicle navigation indoors. The novelty is in the method for correction of position and heading. The third algorithm computes solution for pedestrian navigation system, which is based on body mounted inertial measurement unit and models of human gait.

PREFACE

The work for this dissertation has been carried out at the Department of Computer Systems, Tampere University of Technology. I would like to thank my supervisor Prof. Jarmo Takala for professional guidance and support during these years.

I would also like to acknowledge the unconditional support from Prof. Robert Piché during my thesis work. He was always there for a good technical discussion whenever I had a question or problem.

My colleagues at TUT, especially Jussi Collin, Helena Leppäkoski, Jussi Parviainen, Martti Kirkko-Jaakkola, Olli Pekkalin and Simo Ali-Löytty are gratefully acknowledged for their support and friendship.

This work has been partially done in projects funded by Finnish Technology Agency TEKES and its financial support is gracefully acknowledged. I'd like to express my gratitude to Murata Electronics Oy and u-Blox (former Fastrax Oy) for providing components and equipment for the field tests reported in this thesis.

I especially recognize the invaluable efforts of my pre-examiners Prof. Oleg Stepanov and Dr. Susanna Pirttikangas for providing constructive comments.

My thanks are not complete without acknowledging the love and support of my family. Sergey, Kirill, Nastya, Dima, Yura, Yaroslav and Tanya I thank you all. My warmest thanks go to my dear Svetlana for her love, help and support during these years. Her love has been a source of inspiration to me.

This dissertation is dedicated to my parents. Their unconditional love, support, and understanding over the years is appreciated more than words can express. Above all others, they are most responsible for my success.

Tampere, September 2013

Pavel Davidson

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

2D	Two-dimensional
3D	Three-dimensional
ABS	Anti-lock Braking System
ARCS	Automatic Route Control System
AutoCAD	Software application for Computer-Aided Design
CAD	Computer-Aided Design
CHAIN	Cardinal Heading Aided Inertial Navigation
COG	Course Over Ground
DARPA	Defence Advanced Research Projects Agency
DGPS	Differential GPS
DOF	Degree of Freedom
DR	Dead Reckoning
DRMS	Distance Root Mean Squared
ECEF	Earth Centre Earth Fixed coordinate frame
EKF	Extended Kalman Filter
ENU	Coordinate system axes convention where x is east, y is north, and z is up -axis
FIS	Fuzzy Inference System

GIS	Geographical Information System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
Gyro	Gyroscope, an inertial sensor that measures angular rotation with respect to inertial space about its input axis.
HSGPS	High Sensitivity GPS
HDOP	Horizontal Dilution Of Precision
HDR	Heuristic Drift Reduction
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ITS	Intelligent Trasportation System
LiDAR	Light Detection and Ranging
MEMS	Micro Electro Mechanical System
MMSE	Minimum Mean Square Error
LBS	Location-Based Services
NHC	Non-Holonomic Constraint
OBD	On-Board Dignostics
PDF	Probability Density Function
PDR	Pedestrian Dead Reckoning
PND	Personal Navigation Device
RF	Radio Frequency
RSS	Root Sum of Squares
SHS	Step and Heading System

SLAM	Simultaneous Localization and Mapping
TIMMS	Trimble Indoor Mapping Solution
UWB	Ultra-Wideband
WGS	World Geodetic System
WiFi	Popular technology that allows an electronic device to exchange data wirelessly over computer network
WLAN	Wireless Local Area Network
ZARU	Zero Angular Rate Update
ZUPT	Zero Velocity Update

SYMBOLS

Δ_{seg}^{j}	Mahalanobis distance
σ	Standard deviation
σ_n	Standard deviation of random variable n
x_k	Vector at epoch k
λ	Eigenvalue
d	Distance
$\hat{n}$	Estimate of n
H^T	Transpose of matrix H
Φ	State transition matrix
$\mathcal{N}(x_k, a_k, \Sigma_k)$	Normal distribution with expectation value a_k and standard deviation Σ_k
k	subscript corresponds to the t_k time instant
t_k	time instant
ψ_k	vehicle's heading (ground track)
χ_k	Position measurement error described by the Markov first-order processes
$\widetilde{N}_k$	Measurement of the vehicle's North position
$\widetilde{E}_k$	Measurement of the vehicle's East position
δN_k	North position measurement error

δE_k	East position measurement error
$\rho^h(x)$	Implicit nonlinear function describing the road
R^h	Road h
$R_{i,i+1}$	Road segment between the nodes ξ_i, ξ_{i+1}
ξ_i	i th node
$p(x_k\|x_{k-1})$	Transitional probability density (motion model)
$p(y_k\|x_k)$	Likelihood (measurement model)
$p(x_{0\ldots k}\|y_{1\ldots k})$	Posterior probability density function
$w^{(i)}$	i th particle weight
x_k^{meas}	Measured position at time t_k
ψ_k^{meas}	Measured heading at time t_k
σ_{pos}^2	Position measurement variance
N_{eff}	Effective number of particles
N	Total amount of particles
E_{map}	East positions from the map matching solution
N_{map}	North positions from the map matching solution
E_{DR}	East positions of the dead-reckoning solution
N_{DR}	North positions of the dead-reckoning solution
δS	Odometer scale factor error
τ_g	Correlation time of the gyroscope bias
$\hat{E}$	Estimated vehicle position (easting)
$\hat{N}$	Estimated vehicle position (northing)
σ_{map}^2	Variance of position errors of map matching algorithm

σ^2_{DR}	Variance of position errors of dead reckoning algorithm
$\delta\hat{\psi}$	Estimate of the dead reckoning system's heading errors computed by the Kalman filter
$\delta\hat{N}$	Estimate of the dead reckoning system's north position errors computed by the Kalman filter
$\delta\hat{E}$	Estimate of the dead reckoning system's east position errors computed by the Kalman filter
δV_N	North component of velocity error
δV_E	East component of velocity error
ϕ_N	Horizontal tilt error around North axis
ϕ_E	Horizontal tilt error around East axis
B_N	Projection of North accelerometer biases on the local-level frame
B_E	Projection of East accelerometer biases on the local-level frame
ε_N	Projection of North gyro drift on the local-level frame
ε_E	Projection of East gyro drift on the local-level frame
δV_N^{INS}	North component of INS velocity error
δV_E^{INS}	East component of INS velocity error
$\overline{V}_N^{INS}$	Average North velocity component calculated from the INS indicated velocity
$\overline{V}_E^{INS}$	Average East velocity component calculated from the INS indicated velocity
$\overline{V}_N^{step}$	Average North velocity component over one step calculated from the kinetic model
$\overline{V}_E^{step}$	Average East velocity component over one step calculated from the kinetic model

w_N	North velocity errors of the kinetic model based velocity estimator
w_E	East velocity errors of the kinetic model based velocity estimator
n_a	Wideband measurement noise in accelerometer output
n_g	Wideband measurement noise in gyro output
P	error covariance
Φ	state transition matrix
Q	process noise covariance
H	A matrix that relates measurements to a state vector
R	Measurement noise covariance matrix
R_e	Distance to the Earth center
$\widehat{x}\left(t_k^-\right)$	Predicted state vector in Kalman filter
$\widehat{x}\left(t_k^+\right)$	Updated state vector in Kalman filter
$P\left(t_k^-\right)$	Predicted covariance matrix in Kalman filter
$P\left(t_k^+\right)$	Updated covariance matrix in Kalman filter

1. INTRODUCTION

A Personal Navigation Device (PND) is a portable electronic product which combines a positioning capability and navigation functions. PNDs can be designed for different applications and come in different types and forms. The typical applications will be car and pedestrian navigation, sport and fitness applications, location based services (LBS) and assets tracking. The examples of PNDs include vehicle navigation systems, smartphones, and other navigation enabled mobile devices.

The latest generation of PNDs has sophisticated navigation functions and features a variety of user interfaces including maps, turn-by-turn guidance and voice instructions. Most of the currently available PNDs are based on global navigation satellite system (GNSS), which can provide required position accuracy under open sky when many GNSS satellites are available. However, GNSS receiver performance may be lower than expected. For example, in urban areas, GNSS positioning suffers from degraded satellite availability or multipath error arising from signals reflected by the buildings. Moreover indoors the GNSS signals can be very weak and not suitable for navigation computations.

In many applications, it is desirable to have navigation everywhere including indoors. Therefore, significant efforts have been invested into improvements of navigation in difficult environments. First and foremost some approaches seek to enhance the reception of GNSS receiver by a number of means: high-sensitivity GNSS receivers that use integration across the 50 Hz data bit, anti-jam antennas, assisted GNSS.

Unfortunately, improvements of GNSS receiver performance in difficult environments are limited. Ultimately there will be times when all enhancements to GNSS signal reception fail. In these situations, a number of approaches to GNSS denied navigation have been proposed. These approaches include: use of radio frequency (RF) signals, either those already present or those generated by new and dedicated infrastructure such as wireless local area networks (WLAN), ultra-wideband (UWB),

etc; making measurements with respect to local landmarks or features, which can be optical, acoustic, or RF; self-contained navigation systems based on inertial or speed sensors; map matching using street or indoor maps.

1.1 Scope of the Thesis

This thesis is concerned with improvement of positioning capabilities of PND in GNSS denied environments. The primary objective is to develop a methodology for autonomous personal navigation, which assures reliable and accurate positioning in GNSS denied environment, in particular indoor. In this thesis, the required navigation performance for indoor navigation is a room level, for street navigation, it is desired to identify the correct road segment and location of a vehicle on this segment within 10-20 m. The examples of the applications where such accuracy is sufficient are discovering the nearest banking cash machine or the whereabouts of a friend or employee. The navigation system cannot rely on external infrastructure and can be complementary to GNSS. It may include self-contained sensors and maps or building floor plans. A self-contained sensor block might consist of inertial sensors (gyroscopes and accelerometers), barometric altimeter, and speed sensor (odometer, Doppler radar etc). While GNSS is available the system has all the excellent attributes of the GNSS and autonomous sensors combination. When GNSS is denied the system devolves into a relative navigation (or an ”instrumented dead reckoning (DR)”). The thesis is focused on the following two problems:

- How indoor and street maps can be combined with autonomous sensors to obtain reliable and accurate navigation in GNSS denied environments.
- How low-cost inertial sensors can be used in autonomous pedestrian navigation system.

The autonomous sensor data can be processed to obtain position, velocity, and attitude. A major problem with this approach is accumulation of error: small errors in velocity (acceleration) and angular rate result in large errors in position. Therefore, position and heading update as well as calibration of autonomous sensors are required. Street or indoor maps can possibly be the source of position and heading information to an autonomous navigation system. Thus the navigation solution

calculated based on measurements from autonomous sensors can be corrected and improved by imposing additional constraints on possible vehicle trajectories.

A map can be another major component of an autonomous navigation system. The process of improving navigation through more accurate positioning with the help of a map is called map matching. In this approach the user trajectory, which is computed based on sensor data is compared with the elements of the map in order to improve position and heading estimation. Then the map matching corrections are applied to the trajectory calculated by the dead reckoning system. The map-aided dead reckoning system can potentially provide accurate vehicle positioning for long periods of time without using GNSS data.

State of the art autonomous pedestrian navigation systems use foot-mounted inertial measurement unit (IMU) and apply zero velocity update (ZUPT) to reduce error accumulation. These navigation systems can show good performance. Although foot-mounted sensors are impractical in many applications and alternative methods are needed.

1.2 Thesis Contributions

The thesis's contribution is in improvement of positioning capabilities for autonomous personal navigation systems for "on-foot" and "in-vehicle" applications, which include the following tasks: combination of road network database and car navigation system; combination of building plans with navigation system; improvement of pedestrian dead reckoning system performance. The main contributions of the thesis can be stated as follows:

- A novel method for performing map matching based on information contained in road network database. The proposed mathematical framework for solving the map matching problem is based on recursive implementation of Monte-Carlo based statistical signal processing, also known as particle filtering. The algorithm is robust and has ability to correct position and heading of the dead reckoning system.

- Algorithm that combines navigation data with building floor plan for autonomous navigation systems operating indoor. The proposed method prevents un-

bounded error growth by correcting position and heading errors of dead reckoning system.

- A novel approach to autonomous pedestrian navigation, which combines the data from body-mounted IMU with human gait models. The algorithm has all the advantages and high performance of the foot-mounted IMU approach, but it overcomes its major drawback of impractical location.

1.3 Author's Contribution

The work reported in this thesis has been partially published in publications: Davidson et al. (2008, 2009b,a, 2010a, 2011a, 2010c,b, 2011b); Davidson and Takala (2013); Oshman and Davidson (1999). In all these publications, the author has played a significant role in providing novel ideas and implementing them. The author acted as the first author in 9 publications, in which he provided the ideas, created simulation programs in Matlab, verified performance of the algorithms and wrote manuscripts.

The following algorithms were proposed and developed by the author: data fusion from the accelerometers, gyroscope and GPS for car navigation system (Davidson et al. (2008)), fusion of data from inertial sensors and road map (Davidson et al. (2009b)), adding stop detection module to the car navigation system based on low-cost inertial sensors (Davidson et al. (2009a)), map matching algorithm for vehicle navigation using road network database (Davidson et al. (2010a, 2011b)), map matching algorithm for indoor navigation using building floor plan based on simulated measurements and maps (Davidson et al. (2010c)), indoor map matching algorithm with real-word data (Davidson et al. (2011a)), algorithm for pedestrian navigation combining data from body-mounted IMU and gait models (Davidson and Takala (2013)).

Nevertheless, the publications would not have been possible without the contributions of my co-authors Jussi Collin, Jani Hautamäki, Jarmo Takala, John Raquet, Yaakov Oshman, Manuel Vázquez Lopez and Robert Piché who provided invaluable help with the design of hardware and real-time software, making measurement setup and carrying field tests, validating algorithms and methodology, and finalizing the text.

1.4 Thesis Outline

This thesis is organized as follows: Chapter 2 describes the state of the art technologies applicable to personal navigation devices. Chapter 3 presents a novel approach for map matching algorithm applicable to vehicle navigation using dead reckoning sensors and street map. In Chapter 4 the map aided navigation system for indoor aplication is described including a novel algorithm for position and heading error correction. Finally, Chapter 5 presents a novel approach for pedestrian navigation with IMU strapped to upper body. In Chapter 6, concluding remarks and a summary will be presented. Chapter 6 also presents some recommendations for future research.

2. POSITIONING CAPABILITY OF PERSONAL NAVIGATION DEVICES

PNDs have become ubiquitous over the last 15 years, mainly because of the development of low cost GNSS chipsets. Most modern consumer market PNDs are GNSS driven. However, some applications such as first responders, firefighters, soldiers, and industrial workers require navigation in GNSS denied environment. All PNDs can be generally categorized into two classes according to their positioning capabilities: the devices that require external infrastructure (signals), and devices, which in addition to GNSS include self-contained sensors, and can perform autonomous navigation during GNSS outages. We will call these groups a standard mass market PND and an advanced PND. This thesis is focused on improvement of positioning capabilities for advanced PNDs.

2.1 Modern Mass Market Personal Navigation Devices

Most of the modern commercially available PNDs are small handheld devices, which have been developed primarily to provide positioning and route-guidance information to the user. They can be incorporated into mobile phones or dedicated navigation devices and used for both pedestrian and vehicle navigation. Positioning is based on GNSS or assisted GNSS and, in some cases, on WLAN. This position is compared with the digital map and the most likely position of the vehicle on the road is estimated.

The same device can be used for pedestrian and vehicle navigation. These devices work only when GNSS or WiFi positioning is available. In this case they provide positioning accuracy of about 10 m under open skies, which satisfies the requirements of many consumer applications such as vehicle and pedestrian navigation and LBS. These systems are designed in a way that they fit the computed GNSS positions into

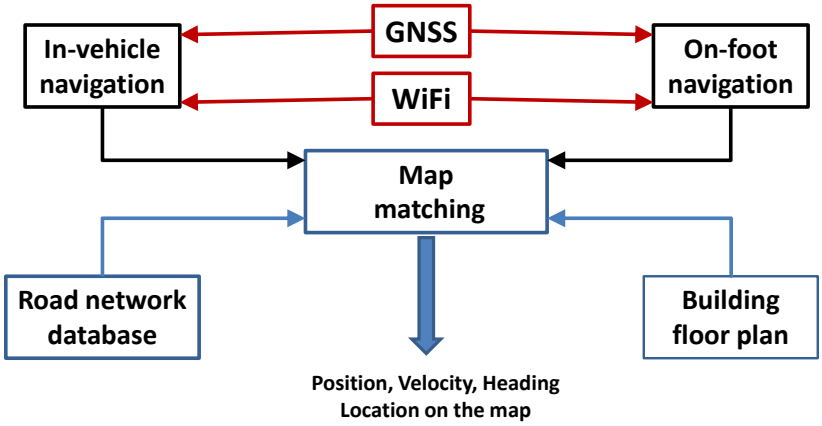

Fig. 2.1. *Typical architecture of a modern mass-market PND.*

a digital map for user interpretation. However, map is used only for display purposes. Fig. 2.1 shows major components and data flow in a standard PND.

These systems are useful as long as the GNSS receiver has a direct line of sight to four or more satellites. In many urban navigation scenarios the GNSS signal availability is limited due to a variety of reasons. For example, a tunnel will completely stop the GNSS based navigation while a typical downtown area with tall buildings will significantly limit the visibility of the number of satellites. In some cases GNSS signals are not completely blocked, but seriously degraded because of multipath. In indoor navigation scenarios GNSS signals are usually not available and WLAN can be the only option.

2.2 Advanced Personal Navigation Devices

Some intelligent transportation system (ITS) and car telematics applications require accurate and reliable positioning with 100% availability. The example of such applications can be lane keeping, and collision avoidance, which require higher position accuracy and update rates, than a commercial GNSS receiver can provide. In addition to this it is also desired to have good integrity for fast detection of sensor and subsystem failures (Skog and Händel (2009)). Positioning technologies based on a single GNSS receiver are vulnerable and cannot meet the positioning requirement for all ITS applications and services, especially, in dense urban areas. Therefore, GNSS

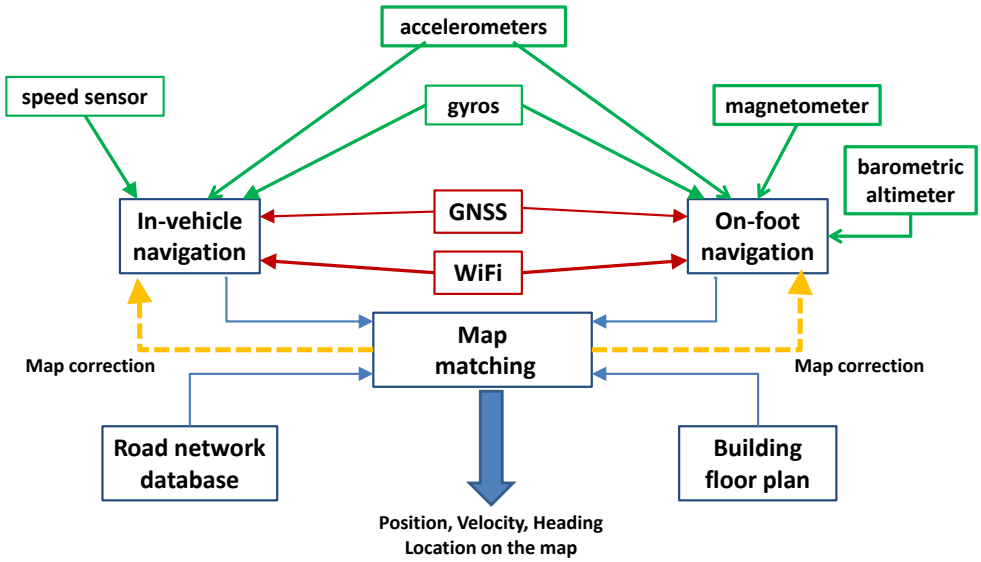

Fig. 2.2. *Typical architecture of an advanced PND.*

has to be supported by additional sensors.

The architecture of a typical advanced PND is shown in Fig. 2.2. The navigation sensors, which require preinstalled infrastructure such as GNSS and WiFi are shown by the red blocks. Autonomous sensors are shown by the green blocks. The blocks "In-vehicle navigation", "On-foot navigation", "Map matching" represent sensor data processing and navigation computations.

Not necessarily all of these sensors have to be present in the PND. A device designed for pedestrian navigation usually contains accelerometers, gyros, magnetometer and, barometric altimeter as autonomous sensors. For vehicle navigation the optimal combination of dead reckoning sensors includes odometer and gyro. The speed data and, in some cases, the gyro data can be taken from a vehicle's on-board diagnostics (OBD) system and wirelessly transmitted to a PND. Thanks to the self-contained sensors advanced PNDs can work during GNSS outages and indoors by switching to autonomous mode. High performance pedestrian and vehicle navigation systems are able to operate autonomously and provide reliable navigation for long period of time. In addition to standard navigation functions and route guidance, advanced in-vehicle PNDs contain navigation systems that are designed to provide highly reliable input for ITS applications with higher accuracy, update rates.

The first step of navigation algorithm is processing data from the self-contained sensor to obtain position, velocity and attitude. Depending on the number of sensors and their kind, quality and application this algorithm can be implemented as a standard INS, a 2D dead reckoning navigator, pedestrian dead reckoning system, etc. We will generally call all these alsgorithms a dead reckoning solution. The main drawback of dead reckoning is unlimited error growth with time. No matter how accurate the sensors are, the residual errors will accumulate and eventually cause large position errors. The following methods can be implemented to improve positioning capabilities of PND for GNSS denied environments:

- Fusion of GNSS and autonomous sensors (Farrell and Barth (1999); Salychev (2004)).
- Fusion with maps including corrections from the map (Dmitriev et al. (1999); Gustafsson et al. (2002)).
- Implementation of navigation algorithms considering constraints specific to pedestrians or vehicles (El-Sheimy (2008); Groves et al. (2007); Foxlin (2005)).
- Vision-aided INS, which employs cameras to extract motion information from images of the surroundings and provide corrections to the inertial estimates (Keßler et al. (2012)).

For consumer market PNDs the challenge is to develop high-performance navigation system solutions using low-cost sensor technology. Advanced car navigation systems usually apply the augmentation of GNSS with one gyro and odometer (Hollenstein et al. (2008); Somieski et al. (2010)). If the odometer data is unavailable the accelerometers can be used instead to estimate vehicle's speed (Chowdhary et al. (2007)). Although this approach is less accurate and reliable (Davidson et al. (2009a)). Examples of advanced commercially available PNDs include TomTom 940 in which a GPS receiver is augmented with a gyro and 2D accelerometer, and u-blox NEO-6V and LEA-6R GPS modules with dead-reckoning based on gyro and odometer. These positioning systems provide continuous positioning everywhere even in tunnels with the output rate of 1 Hz. They also include automatic sensor calibration and temperature compensation.

2.3 Fusion of GNSS and Autonomous Sensors

The goal of autonomous sensor fusion with GNSS is to obtain accurate position estimates with high integrity, full availability and continuity of service. Fig. 2.3 shows the strategy employed for blending the self-contained sensor data with GNSS or other absolute positioning systems (Brown and Hwang (1996)). This process is used to correct the dead reckoning position, velocity and attitude estimates as well as the sensor errors. The autonomous subsystem of PND can be comprised of some of the following sensors: gyroscopes, accelerometers, speed sensor (odometer, Doppler radar etc), barometric altimeter, and magnetometer. It also includes the navigation processor that calculates the vehicle position, velocity and attitude. In 2D case, the attitude contains only heading. The fusion is often based on loosely coupled algorithm in which the autonomous subsystem and absolute positioning system (GNSS, WiFi etc.) can operate as stand alone systems.

In feedforward implementation, which is shown in Fig. 2.3a, the Kalman filter compares output from the dead reckoning navigator with external independent measurements of some of the states and estimates errors in the dead reckoning solution. Then, these estimated errors are subtracted from the dead reckoning solution, thus improving the accuracy. The dead reckoning system operates as if there was no aiding: it is unaware of the existence of the filter or the external data. Corrections to the dead reckoning system output are utilized externally. Acceptable Kalman filter performance is subject to the adequacy of a linear dynamics model, which requires the errors in the dead reckoning solution to remain of small magnitude (Brown and Hwang (1996)).

In feedback implementation, which is shown in Fig. 2.3b, the Kalman filter generates estimates of the errors in the dead reckoning system, but they are fed back into the INS to correct it. In this way, the errors are not allowed to grow unchecked, and the adequacy of a linear model is enhanced. Since the dead reckoning solution now depends on the corrections from the Kalman filter it is important to detect the external aid or filter failures. This failure detection is possible because of the slow dead reckoning solution error dynamics. If such failures are detected the corrections can be removed before significant performance deterioration is caused (Brown and Hwang (1996)).

An odometer provides information on the traveled distance of a vehicle by measuring

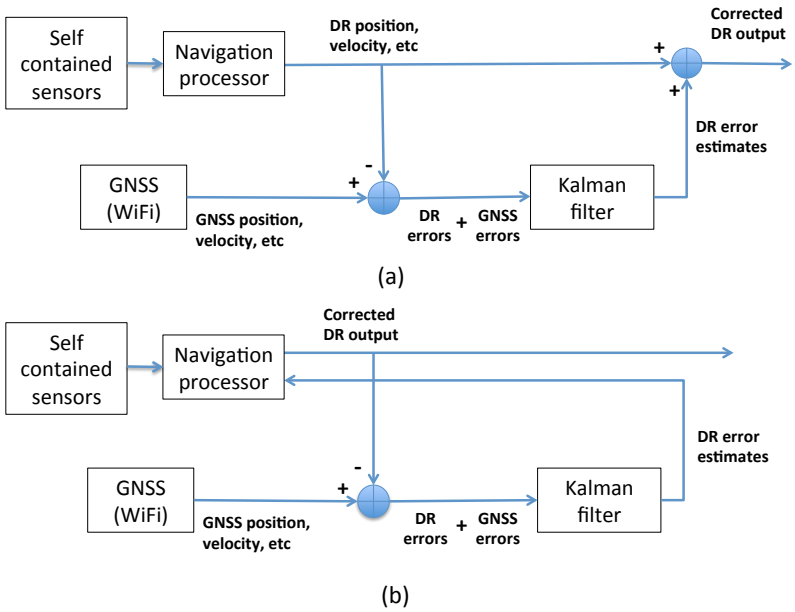

Fig. 2.3. *Sensor fusion algorithm: (a) feedforward implementation and (b) feedback implementation.*

the number of full and fractional rotations of the vehicle's wheels. This is done by an encoder that outputs an integer number of pulses for each revolution of the wheel, which are converted to the traveled distance through multiplication with a scale factor depending on the wheel diameter. If separate encoders are used for the left and right wheel an estimate of the heading change of the vehicle can be found through the difference in encoders output. Information on the speed of the different wheels is often available through the sensors used in the antilock breaking system (ABS).

Accelerometers and gyroscopes measure motion parameters with respect to the inertial space and can be used for both vehicle and pedestrian navigation. Accelerometers sense linear inertial displacement, and gyroscopes measure rotational movement, which is usually represented by angular rate. The displacement, velocity and angles are computed by integrating the output of accelerometers and gyroscopes respectively. Therefore the measurement errors will always accumulate. This is where the GNSS part of the fusion algorithm is required. The GNSS receiver provides vehicle position, velocity and heading at regular intervals. Some GNSS receivers may also output a measure of how good the receiver thinks its solution is.

Barometer measures the altitude above a fixed level. It is more reliable, and often more accurate, than GNSS in measuring altitude. Because barometric pressure changes with the weather, it must be periodically recalibrated at the locations, in which the altitude is known. Barometric altimeters are also sensitive to an operating air-conditioning system when it is used indoors, and, therefore, these limitations have to be taken into account.

Magnetometers measure the absolute azimuth with respect to the magnetic north. The main drawback of the magnetic compass is unpredictable perturbations of the magnetic field caused by the disturbances, which are usually high indoors because of electric fields and steel structures. Magnetometers can be used in pedestrian navigation outdoors, in the places where magnetic disturbances are small.

The first step in the blending process is to create an error signal which is the difference between the GNSS variables and the dead reckoning variables. In the ideal case this difference would be zero because the dead reckoning solution would perfectly track the GNSS solution. However, there are may reasons why the error is non-zero and, in fact, it will always diverge over time. Also, it is worth noting that the sources of error in both systems display quite different properties. GNSS errors are absolute and are less than 10 m for 95% of the time under open skies. In contrast, dead reckoning errors are cumulative and increase without bound at a rate determined by the quality of the sensors and the signal processing algorithms. However, when low-cost sensors are applied their measurements are often corrupted by large errors. Thus implying the need for digital signal processing as an enabler for post-processing of the raw sensor data, including integrity monitoring.

The error signal (which is the GNSS and dead reckoning errors combined) is passed into the navigation filter, which is usually a Kalman filter (Parviainen et al. (2011)). The job of the navigation filter is to estimate the value of the dead reckoning error variables from the combined error input signal. The resulting dead reckoning error estimate is then subtracted from the inertial solution to produce the corrected position solution.

This thesis is focused on navigation systems which require no external infrastructure and can be complementary to GNSS. GNSS and WiFi, which are used in PND have similar positioning accuracy and, therefore, fusion of WiFi and autonomous sensors is quite similar, besides the fact that the heading is not measured when WiFi is used.

While GNSS is available the system has all the excellent attributes of the GNSS/INS combination. When GNSS is denied the system devolves into a relative navigation (or an instrumented dead reckoning).

2.4 Integration of Navigation System with Map

In order to increase the performance of a positioning system, map matching can be added. The idea of map matching is to compare the estimated trajectory of a vehicle with roads or building plans stored in a map database, and the best match is chosen as the position of the vehicle. The implementation of map matching algorithm is different for vehicles on the street and pedestrians indoor and on street.

If vehicle is traveling on the road, its location and trajectory is restricted by the road network. Therefore, a digital map of the road network can be used to impose constraints on the vehicle navigation system solution. Vehicle navigation indoors is quite different from car navigation on the street because their movements are less constrained. In this thesis, it is assumed that the movements indoors are restricted only by walls of a building and non-holonomic constraint.

Implementation of map matching on PND can reduce the accumulation of DR errors. Assuming sufficient map quality, the results of the map matching can be fed back into the system to correct sensor errors and enhance system accuracy. Indeed, in Dmitriev et al. (1999); Gustafsson et al. (2002), for example, a vehicle navigation system is solely based on wheel speed sensors and a probabilistic map matching. A digital road map is used to constrain the possible positions, where a dead reckoning of wheel speeds is the main external input to the algorithm. Gustafsson et al. (2002) acknowledged that by matching the driven path to a road map, a vague initial position (order of kilometers) can be improved to a meter accuracy. This principle can be used for improvement, or even replacement of GNSS.

Standard map matching algorithm is a unidirectional process, where the position and trajectory estimated by the GNSS receiver and/or dead reckoning system, is used to display a vehicle's location on the map. Advanced map matching algorithms have the possibility for bidirectional information (Gustafsson et al. (2002); Skog and Händel (2009)), viewing map information as additional observation in the sensor fusion algorithm.

2.5 Difference in Pedestrian and Vehicle Implementations

The standard GNSS based PNDs can be used for both vehicle and pedestrian navigation without significant modifications of navigation data processing. However, if additional self-contained sensors are used, the navigation system can take advantage of different implementation for pedestrian and vehicle navigation. Vehicle navigation algorithm usually incorporate the non-holonomic constraint (NHC) and odometer, if 6-DOF IMU is used (El-Sheimy (2008)). Pedestrian dead reckoning systems (PDR) usually apply motion constraints based on models of human gait (Groves et al. (2007)) and ZUPT for foot mounted IMU (Foxlin (2005)). Inertial sensors in a PDR can be used for step detection and step segmentation. The advanced PND can also detect the mode of transit and switch the "in-vehicle" and "on-foot" implementations of the navigation algorithm.

The approaches for pedestrian navigation will be discussed in Chapter 5. A high performance pedestrian navigation system usually contains three gyroscopes and three accelerometers and can also use ZUPT and models of human gait. It is a common practice to distinguish between Inertial Navigation Systems (INS) and Step-and-Heading Systems (SHS) (Harle (2013)). An INS is a system that tracks position by estimating the full 3D trajectory of the sensor at any given moment. In the case of pedestrian navigation, an external constraint in a form of ZUPT can be applied at every stride (Foxlin (2005)). Another form of constraint is imposed by kinetic model of human gait (Groves et al. (2007)). In SHS, accelerometers are used to detect steps and sometimes step length, which can be assumed constant in simple systems. Magnetometer and gyro can be used to determine heading. SHS computes position by accruing steps.

Methods for vehicle navigation can also include odometer as an additional speed sensor. If standard INS is used, the NHC can be applied to improve the navigation performance (El-Sheimy (2008)). NHC refers to the fact that the velocity of the vehicle in the plane perpendicular to the forward direction is almost zero. This constraint can be regarded as virtual velocity measurement for cross-track and vertical velocity components.

3. MAP AIDED VEHICLE NAVIGATION USING ROAD NETWORK DATABASE

A map database is a source of valuable information that can be used to improve the position accuracy given by a vehicle navigation system. In the case of street navigation, a map is represented by the road network and only two dimensional planar movement is considered. It is also common to assume that the vehicle is travelling on the road, which is part of this road network. Therefore, the real position of the vehicle is on the network at any moment. Map matching refers to the procedure of comparing the user's location with the underlying map and verifying the location of a vehicle on a road.

When both the user's location and the road network database are very accurate, the map matching algorithm is straightforward. The estimation of vehicle location on the map can be obtained by snapping the position fix to the nearest road segment in the network. Most of the existing algorithms have been developed under assumption that the vehicle position and the map are known with a high degree of accuracy. However, there are many situations in which this is unlikely to be the case. Hence, this research considers map matching algorithms that can be used to reconcile inaccurate position data with an inaccurate road network.

In addition to more accurate position estimation and displaying vehicle location on the map, the computed position can be used to correct the output of vehicle navigation system. This is important in the case of autonomous navigation systems and GNSS receivers in high multipath areas. This thesis shows how the accumulated position and heading errors of the dead reckoning system can be compensated via an interaction with the map database and association of the measured position with the street network.

3.1 State of the Art Methods

One of the first map-aided navigation systems was proposed by French and Lang (1973). It was called the Automatic Route Control System (ARCS). The goal was to direct the operation of a conventional motor vehicle over predetermined routes and control activities (such as the delivery or pickup of items) performed along the route. The approach used in ARCS to determine the vehicle's position along the route and to detect deviations is based on the fact that any route driven from a given starting point has a unique direction-distance signature.

A few years later Lezniak et al. (1977) developed a dead reckoning system combined with a map based on correlation, for automatic vehicle tracking. In this approach, the measured vehicular heading is monitored all the time to determine when the vehicle is turning from its assigned street segment. At those times the aiding from the map stops and tracking switches to the open-loop mode based only on dead reckoning sensors. After a good match between the vehicle trajectory computed by the dead reckoning system and the map is established, the algorithm searches the library of street segments and the vehicle is assigned to the proper new street segment. Tracking then switches back to the closed-loop mode with correction from the map. The map correlation provides an accurate means of correcting the cumulative increasing errors usually characteristic of a dead reckoning system. It also provides a display of vehicle location in a format readily usable by the dispatcher.

The large amount of map matching algorithms was developed in the last twenty years. According to Zhao (1997) and Quddus (2006) the existing map matching algorithms can generally be classified as (a) semi-deterministic approaches including geometric and topological algorithms, (b) probabilistic algorithms, (c) fuzzy-logic algorithms, and (d) pattern recognition algorithms. The type and complexity of the map matching algorithms depends on the application and the available navigation data. Simple algorithms perform well when the user travels on a fixed network similar to those described in French and Lang (1973). The example of such applications can be a bus travelling on known bus route. In this application, the algorithm assumes that the user follows the suggested route and matching is performed to that route. However, if the user deviates from the suggested route, the system detects a large discrepancy between the location computed by the navigation system and the matched location, and algorithm can fail. More sophisticated algorithms do not assume any knowledge

about the route or expected location of the user.

The semi-deterministic algorithms require that the initial vehicle location and a direction of travel will be provided. Another requirement of this algorithm is that vehicle is moving along a predefined, known route. Various conditional tests described in French (1996) can be performed to determine whether the vehicle is travelling on the known road network. Semi-deterministic algorithms work well in situations in which there is fairly good navigation data such as with GPS receivers under open skies or in environments with low multipath. However, if sensors less accurate than GPS are to be used (such as low-cost gyroscopes, differential odometry, etc.), then improvements in existing map matching algorithms may be necessary.

Many modern personal navigation systems use **geometric and topological approaches** to perform map matching. If only geometric information is used, the algorithm relies only on the shape of the arcs and not on the way they are connected (White et al. (2000)). When the topological information is used in addition to geometrical information, the connectivity, proximity, and contiguity of the arcs are also considered. Thus the match is done in context and in relationship to the previous established matches. That makes the topological solution more likely to be correct.

One of the commonly used geometric approaches is **point-to-curve matching** (Bernstein and Kornhauser (1996); White et al. (2000)). In this approach, the position fix obtained from the navigation system is matched onto the closest road segment in the network as shown in Fig. 3.1. The true vehicle path is shown by the thick green line, position fixes and results of map matching are shown by the circles and triangles respectively. In practice point-to-curve matching can give very unstable results in dense road network because the closest link may not always be the correct link (White et al. (2000)). This approach may fail if the user is travelling on a nearby parallel street and near intersections as shown in Fig. 3.1 by the hollow triangles.

Improvement of the point-to-curve algorithm can be achieved by taking into account not only the current position fix but also the previous fixes. This leads **curve-to-curve matching** as was proposed in Bernstein and Kornhauser (1996) and White et al. (2000). In this approach, the vehicle's trajectory is compared against known road network. For given candidate node, it constructs piecewise linear curves comprised from road segments and originating from that node. Then the distance between measured trajectory and candidate curve, usually in terms of weak Fréchet distance,

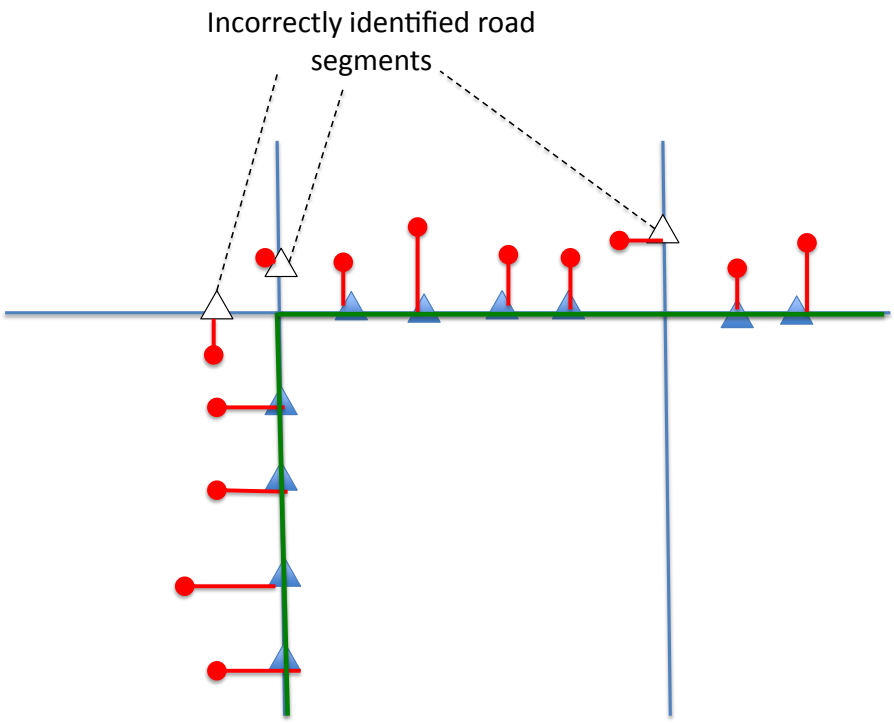

Fig. 3.1. *Point-to-curve map matching algorithm. Measurements (circles), map matching (triangles), true path is shown by the thick green line.*

is computed. The closest curve is chosen as the one on which the vehicle is apparently travelling. This approach is quite sensitive to outliers and depends on point-to-point matching, sometimes giving unexpected results (Quddus (2006)).

An enhancement of the point-to-curve and curve-to-curve map matching algorithms is generalized in the **weighted topological algorithms** which was first proposed by Greenfeld (2002). This algorithm is based on assessing the similarity between the characteristics of the street network and the positioning pattern of the user. The weighting scheme is based on the perpendicular distance of the position fix from the link (proximity), the degree of parallelism between the user's track as it is computed by GPS and the link (orientation), and the intersecting angle (intersection). It is based on topological analysis and it uses only coordinate information for the observed position of the user. It assumes no knowledge of the expected travelling route and it does not use any GPS determined heading and/or speed information. A weighted score for several candidate road segments is computed and the match is determined by se-

lecting the highest score or the most likely candidate for the correct match. Some measures to remove outliers due to GPS errors and to handle skipping of unmatched arcs have been implemented as well. However, the algorithm solves only the problem of identifying the correct link and the location of the traveller on the correct link is not computed.

Fouque et al. (2008) used the Mahalanobis distance computed between the candidate segment and the estimated vehicle position and heading. It can be computed as a weighted sum of vehicle-to-segment distance and difference in orientation between vehicle and segment headings as

$$\Delta_{seg}^{j} = \frac{d^2}{\sigma_d^2 + \lambda_{max}^2} + \frac{(\theta_s - \theta_u)^2}{\sigma_{\theta_s}^2 + \sigma_{\theta_u}^2}. \tag{3.1}$$

where d is the distance from the vehicle to the chosen segment, θ_s is the segment's heading. σ_d, σ_{θ_s} are the standard deviations associated with these measurements, σ_{θ_u} is the standard deviation of vehicle heading measurement, λ_{max} is the maximum eigenvalue of the position covariance. The road segment with the lowest Δ_{seg}^{j} is chosen as the one where the vehicle is travelling.

The weights are computed based on variances associated to these measurements. Different versions of weighted topological algorithm were proposed also by Srinivasan et al. (2003); Quddus (2006). In their algorithms, they not only checked the distance between the position fix and the candidate road segments but also heading difference between the instantaneous vehicle heading and the bearing of the candidate road segment. They state that the enhanced point-to-curve algorithm improves the correct link identification. According to Quddus (2006), the correct road detection rate of topological algorithms based on GNSS positioning data can reach in urban areas 92%. **The probabilistic approach** gives the most reliable solution to map matching problem compared to other methods. It also overcomes the disadvantage of semi-deterministic methods: assumption that vehicle is moving on predefined route. The conventional probabilistic map matching algorithm considers all links that fall within an error ellipse around a position fix as candidate links. The dimensions of the error ellipse are chosen based on the error variance-covariance matrix associated with position error of vehicle navigation system. The size of the error ellipse normally depends on the probability (95% or 99%) that the ellipse contains a true link (Quddus (2006)).

The probabilistic algorithm calculates probabilities of vehicle traveling on different road segments to select a correct road segment and then estimates vehicle position on the selected road link. This approach differs from the semi-deterministic approach in that it does not perform any explicit map matching step, and has advantage in both robustness and flexibility. Different versions of this algorithm were proposed in Scott (1994); Dmitriev et al. (1999); Hall (2001); Gustafsson et al. (2002). They acknowledged that correct road segment identification is a key component of any map-aided estimator, because the performance derived from the map matching algorithm can be misleading if the vehicle location is projected to an incorrect road.

Dmitriev et al. (1999) proposed a mathematical framework for solving the map matching problem based on the recursive Bayesian estimation and non-linear filtering theory. In this work it was mentioned that during the turn a posteriori distribution of the vehicle position on the road is non-Gaussian and non-linear filtering methods are required to solve this problem.

Hall (2001); Gustafsson et al. (2002) implemented particle filtering for map aided car navigation. The performance of the algorithm was evaluated on a simple imaginary map using simulated measurement data. Hall (2001) acknowledged that the particle filter showed disappointing performance: "The frequency of filter divergence was about 20%. Even after convergence the filter could suddenly loose the track of the vehicle, resulting in divergence." He also mentioned a phenomenon, which was referred to as *particle clustering*, i.e. when the initial distribution does not spread the particles on the road well. To improve the particle filter reliability Gustafsson et al. (2002) proposed to split up the measurements to a filterbank, which includes several independent filters. Voting can be used to restart each filter when necessary.

Fuzzy-logic based map matching is an example of the use of a qualitative decision making process to identify the correct road segments among the candidate segments. In fuzzy logic, linguistic terms with vague concepts can be expressed mathematically by making use of fuzzy sets. Fuzzy sets represent expert knowledge and experience to draw inferences through an approximate reasoning process. The basic characteristics of this approach to map matching is to build various knowledge-based IF-THEN rules comprising the speed of the vehicle, the heading, the historical trajectory of the vehicle, the connectivity and the orientation of road links. Examples can be found in Syed and Cannon (2004); Kim and Kim (2001).

Fu et al. (2004) proposed a map matching algorithm that uses a fuzzy logic model to identify the correct link among the candidate links. There are two inputs to the Fuzzy Inference System (FIS): (1) the minimum distance between the position fix and the link, and (2) the difference between the vehicle direction and the link direction. The single output of their fuzzy inference system is the possibility of matching the position fix to a link. Quddus (2006) showed that this simple fuzzy logic model is sensitive to measurement noise. Moreover, the vehicle heading obtained from GPS is inaccurate at low speed, as speed has not been taken into account. As the algorithm selects a link for each position fix with no reference to the historical trajectory, there is a high possibility of selecting incorrect link, especially at junctions.

Quddus et al. (2003) developed another fuzzy logic-based map matching algorithm for land vehicle navigation. In this algorithm, the factors considered to build various knowledge-based IF-THEN rules were the speed, heading, and historical trajectory of the vehicle, the connectivity and the orientation of the links and the satellite geometric contribution to the positioning error (HDOP). A Sugeno-type fuzzy inference system was used to develop the algorithm and the membership functions were trained and modified using a given input/output dataset obtained from GPS carrier phase observations. Quddus claimed that their map matching algorithm outperforms the other algorithms including those algorithms that were also based on fuzzy logic methods. This improvement is primarily due to the use of additional information, such as speed, error sources associated with navigation sensors and map data, and more sophisticated fuzzy rules.

The map matching algorithm proposed in this thesis overcomes various shortcomings of existing algorithms. Most of the existing algorithms cannot reduce the along-track position error because they apply only perpendicular projection from the measured position on the road link. The rate of incorrect road link identification for many algorithms is also high even when user position is known with high degree of accuracy. This is mainly due to the fact that the history of position and heading data is not taken into account. The proposed algorithm is formulated by taking into account the historical trajectory of the vehicle and topological information of the road network, e.g., connectivity and orientation of links, to precisely identify the correct link on which the vehicle is travelling. The algorithm avoids explicit map matching and decision making.

3.2 A Novel Probabilistic Approach to Map Matching

In this section, a probabilistic, numerical approach to the map matching problem is proposed. The algorithm is based on recursive implementation of Monte-Carlo based statistical signal processing known also as particle filtering. The basic principle is to use random samples (also referred to as particles) to represent the posterior density of the car position in a dynamic state estimation framework where road map information is used. Since particle filters have no restrictions on the type of models and noise distribution, the velocity and heading measurement errors can be modeled accurately.

The major advantage of a particle filter for this particular application is that it provides a natural way for road map information to be incorporated into vehicle position estimation by applying the direct constraint on the state vector (which affects each particle). Another advantage is its ability to capture multi-modal distributions which tend to occur when there is uncertainty in which road the user is on. By considering multiple candidate roads, the particle filter is able to quickly adapt if an initial guess at the proper road is found to be incorrect.

This thesis presents both simulation results and results from real-world data collection in a city environment. In field tests, both GNSS and non-GNSS position measurements were collected and simulated. These measurements were combined with OpenStreetMap (a freely-available map database) to calculate the position of the vehicle on the road as it drove through a city. A precision DGPS position solution was used as the reference trajectory in order to evaluate the accuracy of the particle-filter based solution. The results shown in this chapter demonstrate that the proposed particle filter approach is reliable and accurate. It is able to correct large (about 200 m) errors in dead reckoning position by applying the map constraints. It is also demonstrated that the particle filter based map matching algorithm is robust to errors in the predetermined map.

3.3 Digital Maps

Digital map data for map matching algorithms is usually based on a single-line road network representing the centerlines of the roads. Multiple lanes are usually shown as separate road segments. Some databases, for example, OpenStreetMap also include

additional road attributes such as roadway classification (one-way or two-way) and type of the road. Road attributes such as width, turn restrictions at junctions may not exist in the map data. Nevertheless, the accuracy and uncertainty of digital road network data can be a critical issue if the data is used for high accuracy land vehicle navigation.

A digital map is created by converting a paper map into a vector-encoded structure. Road network can be represented by its features expressed as vectors using Cartesian geometry. A feature is denoted as an existing item in the real world. The digitized road network typically represents the road data using line segments whose endpoints (nodes) and shapes are defined in terms of latitude and longitude. According to Zhao (1997), nodes, segments, and shape points can be defined as follows:

- A node is a cross point or an end point of a street and is used to represent an intersection or a dead-end of a road.
- A segment is a piece of roadway between two nodes and is used to represent fragments of roadways and other features.
- Shape points are ordered collections of points, which map the curved portion of a given segment to a series of consecutive straight-line pieces. Road of any curvature can be approximated by a sequence of straight lines (called polylines).

Topology is the arrangement of nodes and segments in a network defining their location, direction, and connectivity. Topological features on the road network include both nodes and segments. Curved roads are normally represented as polylines and straight roads are represented as lines in a digital map. In other words, arcs (roads) without shape points are referred to as lines and arcs with shape points are referred to as polylines. Each polyline consists of a series of lines depending on the number of shape points within the arc. Each arc is assumed to be piecewise linear. For simplicity, each shape point is assumed to be a node. Connectivity information among segments at a junction can be derived from the road network database and can then be used as an important input to the map matching algorithm.

The map of Tampere (OpenStreetMap) is shown in 3.2. The digital road network corresponding to the same area is shown in Fig. 3.3. This map covers the area of

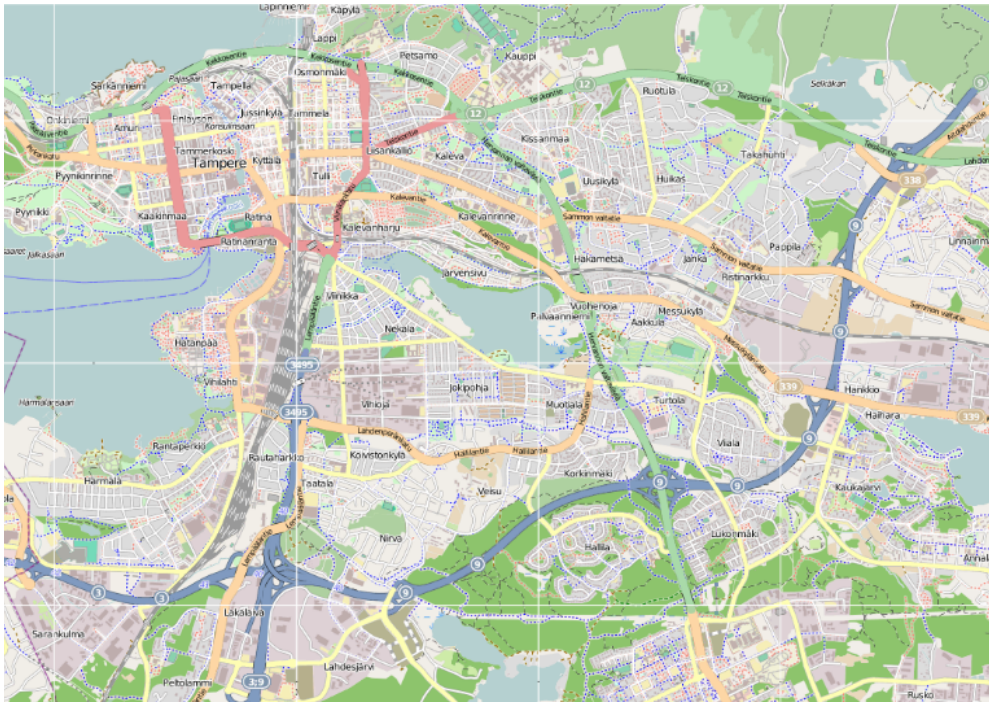

Fig. 3.2. *Street map of Tampere.*

approximately 100 square kilometers. There are about 1300 roads and streets in this area. Some streets are represented by several segments. The total number of segments in this area is approximately 8000. In addition to road links a digital map may include the following parameters about each road: geometry ('Point', 'Multipoint', 'PolyLine', or 'Polygon'), ID within the map database, and attribute fields such as name, type and driving restrictions (for example, oneway).

In the work described in this thesis, a public domain database OpenStreetMap was used. The geographical information stored in navigable road maps (e.g., maps from OpenStreetMap) is usually expressed in geographic coordinates. As proposed in Fouque et al. (2008), only a limited area (called "road cache") around the estimated location need be considered. After the road cache extraction, the points of the polylines that describe the roads are converted into a local tangent East-North-Up (ENU) frame. Then, by choosing a reference point, the transformation between ECEF (WGS84 Earth-Centered, Earth-Fixed) and ENU is computed. Since the elevation is usually not available in a navigable map, we convert the map points in the working frame by supposing that they are all located at the ellipsoidal height. As the

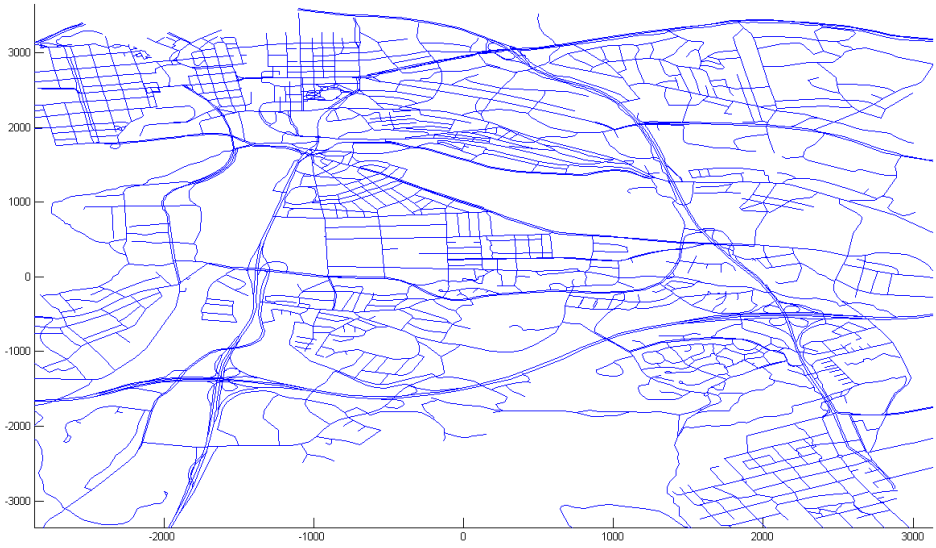

Fig. 3.3. *Digital road network of Tampere.*

ENU frame is attached to a road cache, it should be noted that the working frame is temporary and valid only for small regions.

There is imprecision associated with the GIS based digital road map due to errors in map creation and digitization. As a result of such inaccuracies in the positioning systems and a flawed GIS digital base map, actual vehicle positions do not match with the spatial road map although the vehicle is known to be restricted on the road network. This phenomenon is known as spatial mismatch. The spatial mismatch is often more severe at junctions, roundabouts, complicated flyovers, and built-up urban areas with complex route structure environments.

3.4 Applying Road Network Constraints

A vehicle is restricted to move within the boundaries of streets and parking spaces under normal circumstances and this is especially true for urban areas. Assume that the vehicle is located at certain point on the road. The position fix computed by the vehicle navigation system deviates from the road centerline due to error in the map and the navigation data. The map matching algorithm can apply constraint assuming that the vehicle is on the road network by snapping vehicle's position to a road

centerline of identified road segment. But this only eliminates the cross-track error of the position fix. The along-track error can be reduced when vehicle is turning or moving on a curved road. When the vehicle executes a turn, the distance or along-track error accumulated up to that time is reduced to zero if the turn was correctly identified. The correction of along-track errors during turns can be performed when the algorithm assigns the vehicle to a new street segment.

3.5 Proposed Algorithm

It is assumed that the vehicle is moving on the roads, which are known from the digital database. The objective of map matching algorithm is to estimate recursively the position of the vehicle from a set of measurements. The state vector consists of a vehicle's northern and eastern coordinates, and heading: $x_k = [N_k\ E_k\ \psi_k]^T$. Here, the k subscript corresponds to the t_k time instant. The evolution of the state in time is described with the aid of a constant velocity model in form of the following dead-reckoning equations (Zhao (1997)) and is subject to the road network:

$$\begin{bmatrix} N_{k+1} \\ E_{k+1} \\ \psi_{k+1} \end{bmatrix} = \begin{bmatrix} N_k \\ E_k \\ \psi_k \end{bmatrix} + \begin{bmatrix} L_k cos\psi_k \\ L_k sin\psi_k \\ \Delta\psi_k \end{bmatrix} \tag{3.2}$$

where L_k is the distance traveled from time instance t_k to t_{k+1}, $\Delta\psi_k$ is the change in vehicle's heading during this time. It is assumed that the distance traveled is estimated based on sensor measurements. The heading is assumed to be the same as road segment's heading when a vehicle is moving along straight parts of the road. When a vehicle is turning or moving along curved parts of the road the heading is measured by the sensors. This will be explained in details in Section 3.5.1. The measurements of the vehicle's position and heading are available at time instants t_k from the on-board DR navigation systems:

$$y_k = \begin{bmatrix} \widetilde{N}_k \\ \widetilde{E}_k \\ \widetilde{\psi}_k \end{bmatrix} = \begin{bmatrix} N_k + \delta N_k \\ E_k + \delta E_k \\ \psi_k + \delta\psi_k \end{bmatrix} = x_k + \chi_k + v_k \tag{3.3}$$

where $\delta N_k, \delta E_k, \delta\psi_k$ are the measurement errors that can be assumed a combination of Markov first-order processes χ_k and zero-mean white Gaussian noise v_k (χ_k and v_k are three-dimensional vectors).

As proposed in Dmitriev et al. (1999), roads can be described by an implicit nonlinear function $\rho^h(x)$ in the form of

$$R^h = \left\{ x : \rho^h(x) = 0 \right\}, h = 1, \ldots, M \tag{3.4}$$

where M is the number of roads. For the purpose of map-aided estimation, each road R^h in the road network can be approximated by a set of road segments $R_{i,i+1}$, each of which is a straight line between the nodes ξ_i, ξ_{i+1} that satisfy Eq. 3.4.

It is assumed that the state can be described by partially observable discrete-time Markov chains. Furthermore, the state x_k depends on the previous state x_{k-1} according to the probabilistic law $p(x_k|x_{k-1})$. This problem can be stated as the estimation of the sequence of states $x_{0\ldots k} = \{x_0, \ldots, x_k\}$ given the series of observations $y_{1\ldots k} = \{y_1, \ldots, y_k\}$ subject to the motion model $p(x_k|x_{k-1})$, measurement model $p(y_k|x_k)$, and constraints on the state vector given in the form of a road network. The prior probability at t_0, $p(x_0)$ is assumed to be known. The goal is to find the best trajectory in terms of the minimum mean square error (MMSE) criterion.

This problem can be solved within the framework of the Bayesian estimation theory (Dmitriev et al. (1999)). According to the Bayesian view, the posterior probability density function (pdf) $p(x_{0\ldots k}|y_{1\ldots k})$ contains all the statistical information available about the x_k state vector, based on the information in the $y_{1\ldots k}$ measurements. The algorithm is derived from the recursive decomposition of $p(x_{0\ldots k}|y_{1\ldots k})$ based on Bayes rule and the law of total probability as follows (Arulampalam et al. (2002)):

$$\begin{aligned} p(x_{0\ldots k}|y_{1\ldots k}) &= \frac{p(y_k|x_{0\ldots k}, y_{1\ldots k-1}) p(x_{0\ldots k}|y_{1\ldots k-1})}{p(y_k|y_{1\ldots k-1})} \\ &= \frac{p(y_k|x_{0\ldots k}, y_{1\ldots k-1}) p(x_k|x_{0\ldots k-1}, y_{1\ldots k-1}) p(x_{0\ldots k-1}|y_{1\ldots k-1})}{p(y_k|y_{1\ldots k-1})}. \end{aligned} \tag{3.5}$$

If the probabilistic model of the transitional density is described by a Markov process of the first order, such as $p(x_k|x_{0\ldots k-1}, y_{1\ldots k-1}) = p(x_k|x_{k-1})$ then the calculation of Eq. 3.5 can be simplified. It is calculated recursively as

$$\begin{aligned} p(x_{0\ldots k}|y_{1\ldots k}) &= \frac{p(y_k|x_{0\ldots k}, y_{1\ldots k-1}) p(x_{0\ldots k}|y_{1\ldots k-1})}{p(y_k|y_{1\ldots k-1})} \\ &\propto p(y_k|x_k) p(x_k|x_{k-1}) p(x_{0\ldots k-1}|y_{1\ldots k-1}). \end{aligned} \tag{3.6}$$

Since the distribution in (3.6) cannot be solved analytically in general nonlinear and non-Gaussian case, a particle filter will approximate it by using a cloud of *particles*

and their associated weights. The ith particle is a candidate state vector $x^{(i)}$ and has a weight $w^{(i)} \in [0,1]$, and each particle is propagated in a procedure analogous to Kalman-type filters:

1. Prediction step: The particles are projected to the next time step by sampling from a proposal distribution which we choose to be the transitional model $p(x_k|x_{k-1}^{(i)})$.

2. Update step: The weights are updated according to

$$w_k^{(i)} \propto w_{k-1}^{(i)} p\left(y_k|x_k^{(i)}\right). \tag{3.7}$$

   If multiple types of measurements y are available, a separate update step can be taken for each type.

Eq. 3.7 is expressed as a proportion instead of an equality because we normalize the weights to sum to unity; this way, it is straightforward to estimate, e.g., the mean and covariance of the posterior distribution.

The resulting set of weighted trajectories $\left\{x_{0\ldots k}^{(i)}, w_k^{(i)}\right\}$, $i = 1,\ldots,N$ with normalized weights provides an approximation to the distribution $p(x_{0\ldots k}|y_{1\ldots k})$. Based on the discrete approximation of the posterior pdf, an estimate of the best trajectory at step $k+1$ can be obtained. The weighted mean of the particles represents a Monte Carlo approximation of the posterior pdf expectation, which gives the best trajectory in terms of the MMSE.

3.5.1 Particle Filter Implementation

The proposed system model has two operational modes: (1) a vehicle is moving along straight parts of the road and (2) a vehicle is turning or moving along curved parts of the road. Switching between these two modes is performed based on an analysis of the vehicle's heading rate data from the sensors. During the first operational mode, the particles are propagated using only the speed information from the onboard speed sensors (odometer, GNSS, etc.). The ith particle's heading is assumed to be the same as the heading of the road segment where this particle is located and which is known from the map:

$$\psi_k^{(i)} = \psi_{seg}^{(i)}. \tag{3.8}$$

In this case the propagation of i th particle can be described by the following equation

$$\begin{bmatrix} N_{k+1}^{(i)} \\ E_{k+1}^{(i)} \end{bmatrix} = \begin{bmatrix} N_k^{(i)} \\ E_k^{(i)} \end{bmatrix} + (L_k + \Delta L_k^{(i)}) \begin{bmatrix} cos\psi_k^{(i)} \\ sin\psi_k^{(i)} \end{bmatrix} \tag{3.9}$$

where L_k is the distance traveled from time instance t_k to t_{k+1} as it measured by the odometer. For each particle the heading is given by (3.8), and distance traveled by this particle is perturbed using some model for distance measurement error $\Delta L_k^{(i)}$. This distribution does not have to be Gaussian. However, in our case the particles are drawn from Gaussian distribution with zero mean and constant variance. The variance of this distribution is one of the design parameters, which can influence diversity of the particles. This propagation model can guarantee that the particles will always stay on the road. However, different particles can move on different road segments. The road segment with the highest probability (with more particles on it) is selected as the most likely road segment where vehicle is located. If the particles are moving on the correct road segment then estimated position cross-track error can be reduced substantially by applying a simple perpendicular projection of the position fixes onto the selected link. The estimated vehicle position can also be calculated as the weighted average of all the particle coordinates from this segment.

During the second operation mode when the vehicle is turning, its heading and speed are required; the propagation model can be described by the dead-reckoning equations

$$\begin{bmatrix} N_{k+1}^{(i)} \\ E_{k+1}^{(i)} \\ \psi_{k+1}^{(i)} \end{bmatrix} = \begin{bmatrix} N_k^{(i)} \\ E_k^{(i)} \\ \psi_k^{(i)} \end{bmatrix} + \begin{bmatrix} L_k cos\psi_k^{(i)} \\ L_k sin\psi_k^{(i)} \\ \Delta\psi_k^{(i)} \end{bmatrix} \tag{3.10}$$

where the change in heading is measured by the vehicle's navigation system. The road segment identification is not performed at this step. During the turn the vehicle and the particles are moving along some trajectories as illustrated in Fig. 3.4. There are some important features of these trajectories that help to reduce the along-track error of vehicle position estimation after the turn:

- The vehicle and the particles start turning at the same time since the turn is sensed by some heading-rate measuring device, e.g., gyroscope.
- The vehicle and the particles stop turning at the same time.

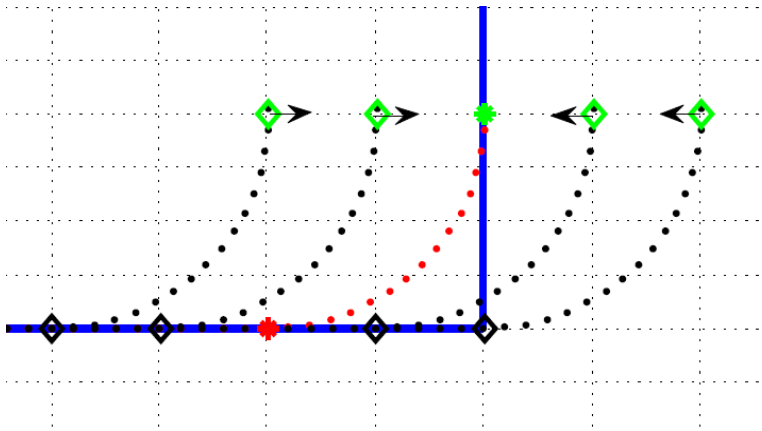

Fig. 3.4. *Propagation of particles during the turn. True vehicle path is shown by the red dotted line; Particles trajectories are shown by the black dotted lines; The asterisks designate the true vehicle location before and after the turn; the diamonds designate the particles locations before and after the turn.*

If gyro and odometer are used as dead-reckoning sensors the accumulation of position errors during the turn is small. Therefore, all these trajectories are parallel and can be obtained by parallel translation of the actual vehicle trajectory along the horizontal road link.

In ideal case, when propagation of particles during the turn is error free, the particles at the end of the turn will be on the same line parallel to the road link where they started the turn. Applying perpendicular projection of the particles position fixes onto the selected link will eliminate the along-track error of the estimated vehicle position accumulated before the turn.

Note that this approach of eliminating along-track error works for turns at any kind of road junction (not only 90 degrees) and also on curved roads. However, in reality, because of position errors accumulated during the turn, there will be some residual along-track error of the estimated vehicle position after the turn. The magnitude of this error depends on the quality of dead-reckoning sensors and curvature of the turn. For road links with small curvature, the reduction of along-track error is negligible.

The particle weights are updated using the recurrent formula given in Eq. (3.7) where

$p(y_k|x_k^{(i)})$ is the likelihood calculated for each particle based on the proximity between the position fix and the particle and the difference between the measured vehicle heading and the heading associated with this particle. The likelihood is calculated according to

$$p(y_k|x_k^{(i)}) \propto \left\{ -\frac{\left(\psi_k^{(i)} - \psi_k^{meas}\right)^2}{2\sigma_{hdg}^2} - \frac{(N_k^{(i)} - N_k^{meas})^2}{2\sigma_{pos}^2} - \frac{(E_k^{(i)} - E_k^{meas})^2}{2\sigma_{pos}^2} \right\} \quad (3.11)$$

where $\psi_k^{(i)}$ is the ith particle's heading, ψ_k^{meas} is the vehicle's measured heading, σ_{hdg}^2 is the heading measurement variance, $x_k^{(i)}$ is the ith particle's coordinates and heading, N_k^{meas}, E_k^{meas} are the measured vehicle North and East position coordinates, and σ_{pos}^2 is the position measurement variance.

When estimating the posterior distribution in the way described above, there will eventually be only few, or even zero, samples that have a nonzero weight. This both wastes computational resources and causes the filter to fail if all samples have zero weight. This problem is called *degeneracy* and can be avoided by *resampling* (Arulampalam et al. (2002)). Resampling procedure eliminates particles with small weights and multiply particles with large weights. During the resampling step a new set of particles is constructed by drawing them N times from the discrete distribution defined by the old set of particles and their respective weights. Finally, all weights are reset to $1/N$. This new set of particles represents the same distribution as the previous set but makes use of all N particles. Various methods for drawing the new particles have been developed. The examples of systematic, multinomial, residual and stratified resampling algorithms are given in Gordon et al. (1993); Kitagawa (1996); Liu and Chen (1998); Douc and Cappé (2005); in this study, the approach of multinomial resampling was used.

A common strategy is to resample whenever the effective number of particles defined as (Ristic et al. (2004))

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} \left(w_k^{(i)}\right)^2} \quad (3.12)$$

is lower than a predefined threshold. Here N is the total amount of particles. The threshold lies between $1 \leq N_{thr} \leq N$ and can be chosen from the field tests to obtain desired performance. In our tests it was set to $N/2$. When N_{eff} is small then only small number of particles have substantial weight which indicates a case of severe

degeneracy. Note that if no other updates than the map update in Eq. (3.11) are being applied, N_{eff} equals the number of particles with nonzero weight if the initial particles are uniformly weighted.

If resampling is applied at each update step, the relationship in (3.7) can be reduced to

$$w_k^{(i)} \propto p\left(y_k | x_{k-1}^{(i)}\right). \tag{3.13}$$

3.5.2 Correcting the Dead Reckoning Solution

The current algorithm requires position and heading measurements as described in Eq. (3.11). If any means to obtain position and heading are available, e.g., GPS, then it is used for these measurements. However, if only dead-reckoning sensors are available (such as a gyro and odometers), then some sort of position and heading measurement must be generated in order to use the algorithm proposed in Section 3.5.

The approach taken here is to use the map-based solution (based upon the weighted mean of the particles) to calculate the error in the dead-reckoning solution on an occasional basis, e.g., every 1000 m and immediately after each turn. Then, at every epoch, a corrected dead-reckoning solution (i.e., the raw dead-reckoning solution, after correcting for the error) is used as a pseudo-measurement for the update described in Eq. (3.11).

When a map is available it can provide such correction to dead reckoning solution. The accuracy of map matching is limited mainly by digital map accuracy which is about $2-5$ m in Europe. If the road is identified correctly, then the cross-track error is always about the same as the accuracy of the map. The along track error is usually smaller after vehicle turned. Therefore, it is better to perform the correction of the dead reckoning solution after the turn. The error in the dead-reckoning solution is calculated on an occasional basis as follows. First, the estimation of vehicle position is determined as a weighted average of dead reckoning and map matching solutions:

$$\begin{aligned} \hat{E} &= \left(\frac{\sigma_{DR}^2}{\sigma_{map}^2+\sigma_{DR}^2}\right) E_{map} + \left(\frac{\sigma_{map}^2}{\sigma_{map}^2+\sigma_{DR}^2}\right) E_{DR} \\ \hat{N} &= \left(\frac{\sigma_{DR}^2}{\sigma_{map}^2+\sigma_{DR}^2}\right) N_{map} + \left(\frac{\sigma_{map}^2}{\sigma_{map}^2+\sigma_{DR}^2}\right) N_{DR} \end{aligned} \tag{3.14}$$

where E_{map} and N_{map} are the east and north positions based on the weighted mean of the particles, E_{DR} and N_{DR} are the east and north positions of the raw dead-reckoning

solution, $\hat{E}$ is the estimated vehicle position (easting), $\hat{N}$ is the estimated vehicle position (northing), σ^2_{map} is the variance of position errors of map matching algorithm, and σ^2_{DR} is the variance of position errors of dead reckoning algorithm. Note that if σ^2_{map} is small (such as after a turn), then $\hat{E}$ and $\hat{N}$ will be very close to E_{map} and N_{map}.

Next, the error in the dead-reckoning solution is calculated as

$$\begin{aligned} \delta E &= \hat{E} - E_{DR} \\ \delta N &= \hat{N} - N_{DR}. \end{aligned} \tag{3.15}$$

These error terms are then held constant and are used to correct the dead-reckoning solution to generate position updates until the next time they are recalculated (typically at a km of travel or after a turn, whichever comes first). An alternative approach for dead reckoning position and heading errors correction will be shown in the next chapter.

3.6 Simulations and Field Tests

This section presents both simulation results and results from real-world data collection in a city environment. We start from the simulation results. To demonstrate the performance of the proposed algorithm the road network, vehicle trajectory, and vehicle position measurements were simulated (Fig. 3.5). It was assumed that the vehicle is traveling along the road (which is typically the case) and its heading matches the heading of the current road segment when the vehicle is travelling along straight stretches of road. We also assume that the terrain is flat and, therefore, the altitude will not be estimated. The road network consists of a set of parallel lines. The triangles and circles denote the true vehicle position and the estimated position, respectively. The stars indicate the position measurements. For illustration purposes only, the corresponding measurement is connected with true position via a dotted line and with corresponding estimated position via a solid line (Fig. 3.5).

In this example it was assumed that the speed over ground and heading measurements are available, but they are corrupted by measurement noise with standard deviation of 2 m/sec and 3 deg, respectively. The position measurement errors include combination of constant offset and random noise with the total distance root mean square (DRMS) of horizontal position errors approximately 40 m. Such large position errors

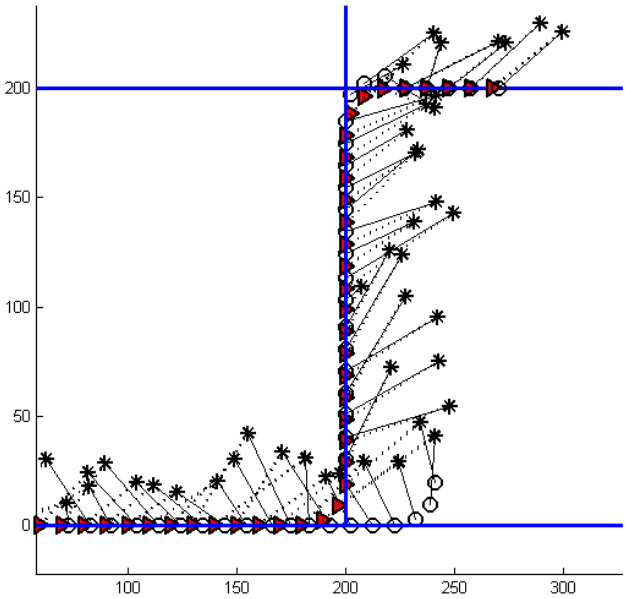

Fig. 3.5. *Map matching results for simulated vehicle path. Triangles show the true vehicle position. Circles show the estimated position. Stars show the position measurements. For each time instant, the corresponding star, triangle, and circle are connected.*

may correspond, for example, to performance of a GPS receiver in high-multipath urban environment.

The performance of the particle filter was evaluated when the vehicle was moving along the trajectory that included several intersections with left and right turns. The simulation results are based on 200 particles. A part of this trajectory and the results of map matching are shown in Fig. 3.5. These results show that cross-track error is always reduced to a level of digital map error. The along-track error is reduced after vehicle turned on intersections. For example before the second intersection the along-track error was approximately 10 m. After the turn the along-track error was reduced to approximately one meter. The distribution of particles along the road segment before and after the turn is shown in Fig. 3.6. Before the turn, the deviation of estimated position from the true position is about 10 m and the standard deviation is about 4 m. In the histogram the horizontal axis represents the deviation of the

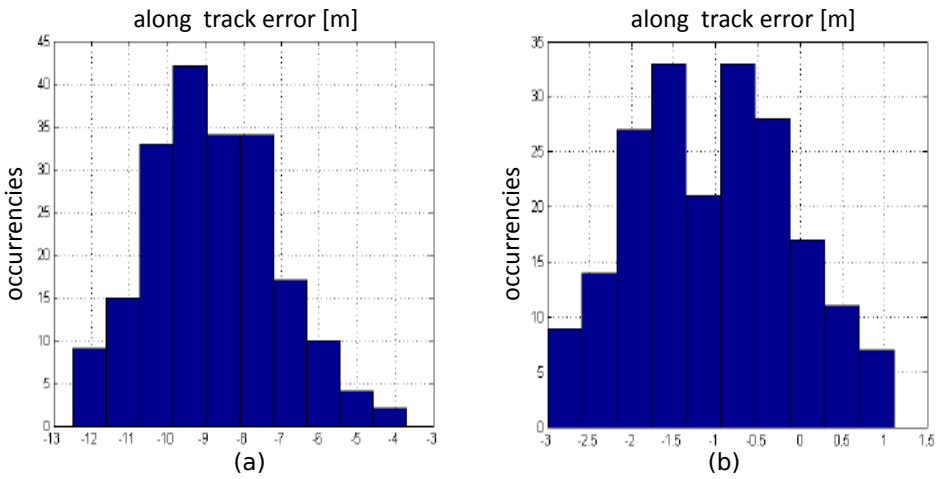

Fig. 3.6. *Distribution of particles along the road segment before (a) and after (b) the turn.*

particles from the true position in meters. After the turn, the deviation from the true position is less than 1 m and the standard deviation is about 2 m.

This example shows that the map matching based on a particle filter can improve substantially the positioning by reducing both along-track and cross-track errors. The cross-track error can be eliminated when the vehicle is not turning. In this case, if the road segment is correctly identified, the vehicle position calculated by GNSS or another navigation system can be corrected by projecting it onto a chosen road link since we know that the vehicle is located on this segment. But this does not eliminate the along-tack error, which can be removed only when the vehicle is turning at intersections or moving on curved roads. In this case, the particle filter is switched to the second operational mode in which the vehicle heading is measured by an on-board sensor. After the turn the vehicle position estimate was substantially improved by reducing the along-track error to sub-meter level.

3.6.1 Field Test

The proposed map matching algorithm was also tested with actual digital maps and real-world heading rate and ground speed data. The heading rate was measured by the Murata SCR-1100-D04 low-cost MEMS gyro. The speed data was collected from the standard speed sensor installed in the car using the standardized digital communication port and on-board diagnostics (OBD) interface. The algorithm position accur-

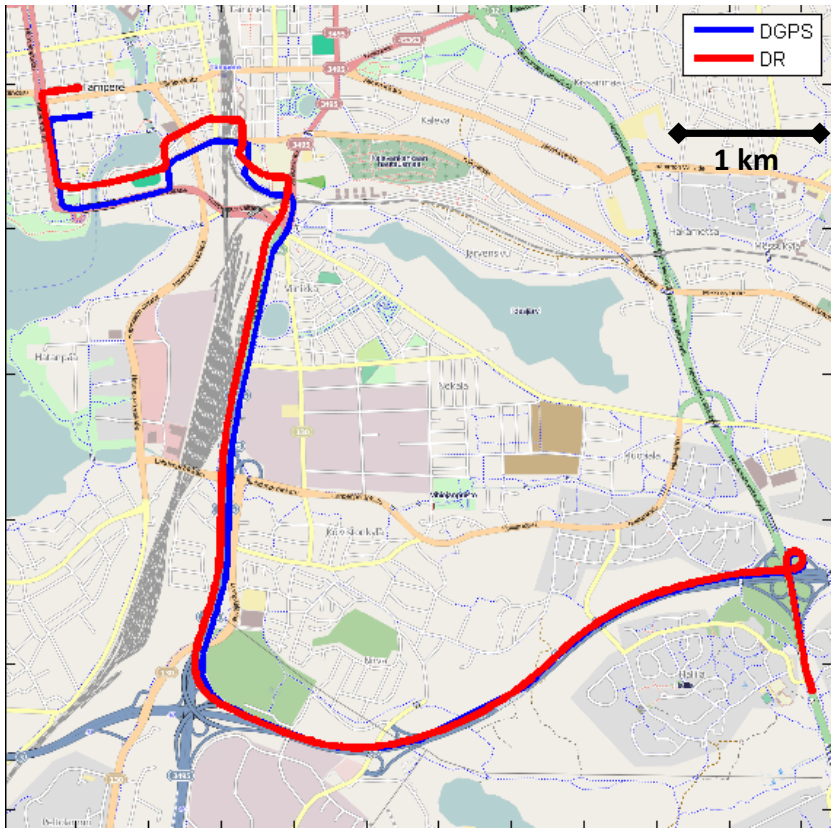

Fig. 3.7. *The 12 km test route.*

acy was analyzed against the vehicle position as determined from the high accuracy Novatel differential GPS receiver with carrier phase capability. It should be mentioned that GPS was not used for navigation, although GPS signals were available most of the time. The purpose of these tests was to show that map aided low-cost dead reckoning navigation system can provide accurate navigation for long period of time without using any GPS data. The 12 km test route included a mix of high-speed multilane highways, road interchanges, regular street, roads, and several turns (Fig. 3.7). The actual car location is shown by blue dots. The uncorrected dead reckoning solution is shown by red dots.

An example of map matching algorithm performance during the turn is shown in Fig. 3.8. The progression of the original uncorrected dead reckoning solution is shown by the green squares. The output rate of this solution is 1 Hz, and the last square corresponds to the currently estimated vehicle position. The magenta aster-

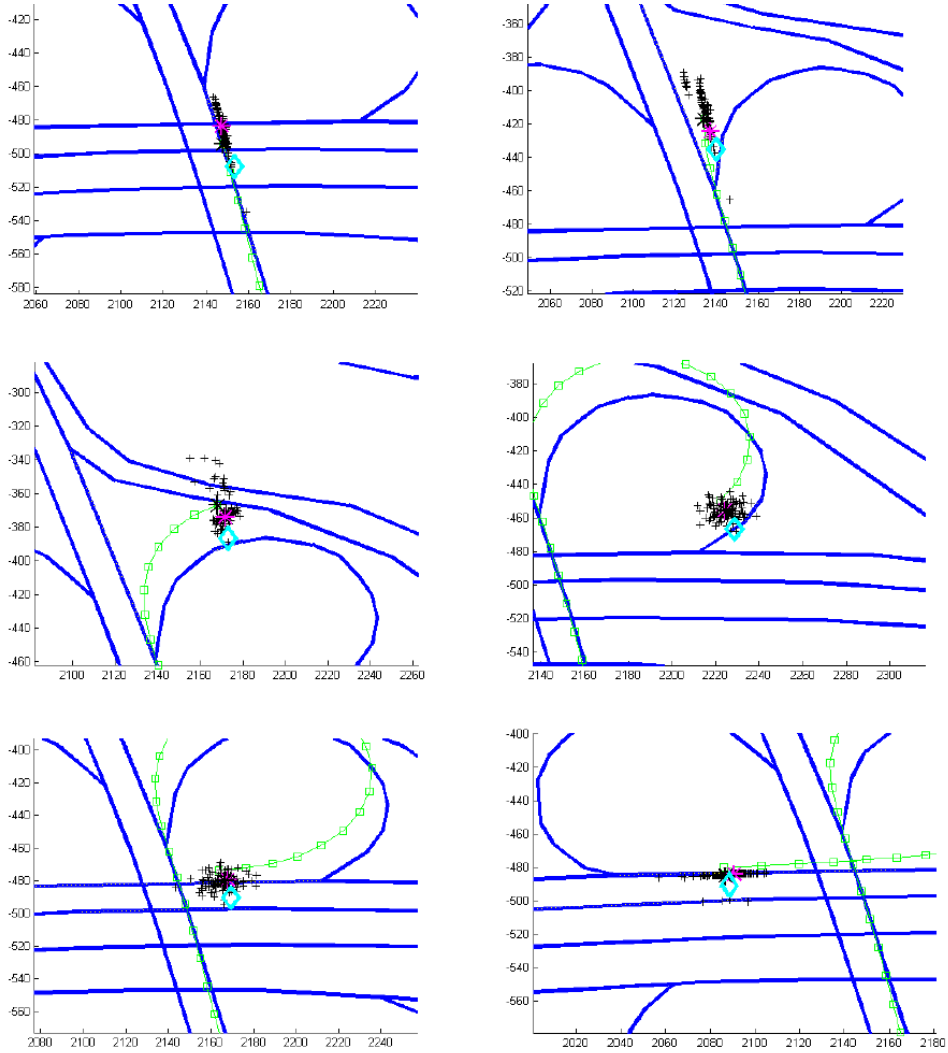

Fig. 3.8. *Map matching algorithm performance during the turn. (Squares: original, uncorrected DR solution; Points: particle locations; Large asterisk: weighted mean of particles; Diamond: true location from DGPS).*

isk designates the corrected dead reckoning position estimate. Black points are the particles with their weighted mean location shown by the star. The cyan diamond shows the true vehicle location as it determined by the high accuracy DGPS receiver. This example shows that before the turn the along track error of the map matching algorithm was approximately 60 m. This error was reduced after the turn to less than 5 m. The corrected dead reckoning solution was updated after the turn so its

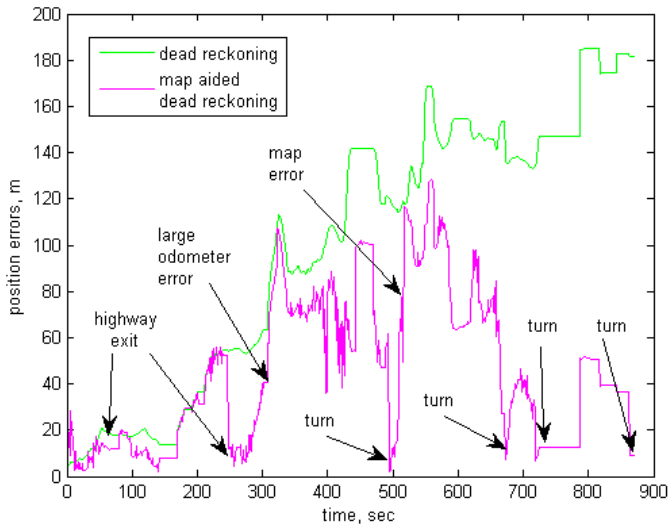

Fig. 3.9. *Comparison of DRMS of horizontal position errors for dead reckoning only and map aided dead reckoning navigation algorithms.*

error becomes less than 10 m. The calculated position offset will be now added to the original dead reckoning solution until next position offset. The corrected dead reckoning position will be used as a position measurement when performing position update of the particles.

Fig. 3.9 shows the DRMS of horizontal position errors for dead reckoning only and map aided dead reckoning navigation algorithms during the 12 km urban test drive. The performance of map aided dead reckoning algorithm depends very much on the vehicle trajectory, especially turns. When vehicle is moving along the straight or slightly curved stretches of the road only cross track errors can be eliminated. After the turn also the along track error can be reduced. During the first 250 sec when the vehicle was moving on the highway the position accuracy improvement was not significant. After the exit from the highway the position error of the map matching algorithm was significantly reduced.

The final part of the test route included several turns. Therefore, the position error was relatively small: $5-10$ m after the turn and increasing gradually because of distance error accumulation. The odometer accuracy has significant impact on along track error growth.

Another potential source of errors is error in digital road map. One road along our test route was built recently. This road was not included in our map database. Therefore, the map matching algorithm identified the wrong road segment and the position estimation error suddenly increased from 5 m to approximately 120 m. It took for the algorithm about two minutes to recover from this incident, identify the correct road link, and start accurately estimating vehicle position again.

3.7 Conclusions

This chapter has shown how map matching algorithm can improve car navigation system performance. This becomes very important when the position calculation is based on dead reckoning sensors. The examples show that the position errors of map aided dead reckoning navigation system can be kept bounded as opposed to the unbounded error growth of the conventional dead reckoning. The accuracy depends on how often turns occur. It was also demonstrated that the particle filter-based map matching algorithm is robust to errors in the predetermined map.

The performance of the proposed particle filter based map matching algorithm is limited by the following factors:

- Position errors in digital road network. Based on the results of road tests we estimated that in Finland this errors usually do not exceed $2-5$ m.
- Uncertainty of vehicle location on the road. The road link is described by its centerline but the actual vehicle location can be slightly different.
- Errors in velocity and heading-rate sensors limit the along-track error estimation during turns and along straight stretches of the route.
- Amount of turn and frequency at which these turns occur.

The particle filter performance can be adjusted by changing ground speed noise and position and heading variances. Increasing ground speed noise improves the particles diversity. Position and heading variances have to match the position and heading measurement errors of onboard sensors. If GPS is used to measure heading, the heading errors can be quite large during the turns and when the vehicle speed is low. This can also limit the accuracy of along-track error estimation.

Compare to other particle filter based algorithms (Hall (2001); Gustafsson et al. (2002)) our algorithm gives reliable and robust solution, which is able to cope with some inaccuraces in map database. During 12 km test route the algorithm never diverge from the correct path. One of the biggest advantages is that the algorithm does not include any explicit decision making process.

Compare to topological (Greenfeld (2002); Srinivasan et al. (2003); Quddus et al. (2003); Quddus (2006); Fouque et al. (2008)) and fuzzy logic algorithms (Syed and Cannon (2004); Kim and Kim (2001); Fu et al. (2004)) the proposed method has the following advantages:

- Can reduce the along-track position error
- Works reliably even when position measurements are less accurate than those of GNSS under open skies
- Can work with dead reckoning sensors in autonomous mode for long period of time. Provide position and heading correction to the dead reckoning system

4. MAP AIDED AUTONOMOUS NAVIGATION INDOORS

The previous chapter described the land vehicle navigation algorithm, which fuses the autonomous sensors data and road network database. Map matching was used to fit an estimated path into the maps. It was shown that the road network data can improve the accuracy of positioning and correct the position and heading computed by the autonomous vehicle navigation system. The goal of this chapter is to extend the map aided autonomous navigation algorithm to indoor applications. The proposed algorithm can be applied to vehicular navigation systems under certain assumptions:

- The displacement and heading are obtained using autonomous sensors such as accelerometers, gyroscopes, odometer or speed sensor, magnetometer, or combination of several sensors.
- The non-holonomic constraint is valid.
- The indoor maps or building floor plans are known before the actual deployment.

Similar to the case of street navigation the fusion of autonomous sensor data with building plans helps to curb the divergence of position and heading errors and achieve long term stability for fully autonomous navigation systems. The major focus will be on correction of position and heading errors of the dead reckoning solution.

Mobile robot navigation indoors is quite different from vehicle navigation on the street because their movements are less constrained and, therefore, much more difficult to model than the movement of a typical car on the street. They can move across a much more diverse area, quickly change direction, use lateral movement, or back step. In the case of indoor navigation the movement is restricted only by the walls of a building. The ability to correctly identify the true path stems from the fact that any path originated from a given starting point has a unique signature.

In this thesis, we consider only 2D planar movement but the results can be extended to include the case of vertical movement by adding vertical links in the map database such as elevators, staircases, and escalators. The proposed algorithm is formulated by taking into account the historical trajectory of the vehicle and topological information of the building floor plan (e.g., connectivity and orientation of walls and other obstacles) to precisely identify the user location. In this chapter, the numerical approach that was described in Chapter 3 is modified to solve a map matching problem indoors. The solution is also based on particle filtering.

In addition to more accurate position estimation and displaying vehicle location on the map, the computed position can be used to correct the output of navigation system. This is important in the case of autonomous navigation systems and GNSS receiver in high multipath areas. It is also shown how the accumulated position and heading errors errors of the dead reckoning system can be compensated via an interaction with the map database and association of the measured position with the street network. The proposed algorithms serve to compute position and heading corrections to the vehicle navigation system as well as recalibration of the autonomous sensors in the case of dead reckoning system. If it is known that the vehicle is located in a certain room of the building then the position of the dead reckoning system can be updated.

4.1 State of the Art Methods

There are currently several approaches to map aided navigation indoors: topological map matching, probabilistic map matching based on particle filtering, reduction of heading error by comparison with building cardinal heading. The purpose of these algorithms is to improve positioning and heading by fitting the estimated path into building plans.

Similar to the street map matching the topological algorithms for indoor applications use a link-node representation of a building plan Gilliéron et al. (2004). A node is a point defined by its coordinates including altitude. The altitude can be given in terms of a floor number. The nodes correspond to the junctions and to the points of interest in the building. A link is a straight line connecting two consecutive nodes. The links may correspond to the corridors, staircases, entrances to elevators, passages between buildings, etc. The consecutive links are connected and represent a network similar to

a road network. The link-node model can be obtained from AutoCAD building floor plan. However, developing a set of map matching functions for pedestrian navigation is a challenge because the trajectory of people is not always similar to the geometry of the mapping data. Development of algorithms indoors is based on the comparison of topological elements from the trajectory and the database. The goal is to identify the correct link and then location of a user on this link.

The topological map matching approach for indoor applications was proposed by Gilliéron et al. (2004); Spassov (2007). Both of these methods rely on the similarity of the trajectory geometry and the topological features of the link-node graph. The first method is based on the Bayesian inference where the estimation is computed considering the walked distance and azimuth. The second method represents a new application of the Fréchet distance as a degree of similarity between two polylines.

Indoor versions of probabilistic map matching algorithm, which are based on particle filtering were developed by Klepal and Beauregard (2008); Beauregard (2007); Woodman and Harle (2008); Ascher et al. (2012); Krach and Robertson (2008); Khider et al. (2008). These algorithms use a set of random samples (*particles*) to represent the posterior density of the unknown position in a dynamic state estimation framework where floor plan information is used. The particles are distributed over the digital building plan where walls represent impassable obstacles and give an approximation of the probability density function of the user's position. If a particle collides with a wall, it is excluded from the Monte Carlo simulation.

In this particular application, a particle filter provides a natural and intuitive way of incorporating building plan information into position estimation, by applying the direct constraint to each particle as was implemented in Beauregard (2007); Klepal and Beauregard (2008); Woodman and Harle (2008); Ascher et al. (2012). All these approaches use the same method of building constraints implementation. The differences lie in the transitional prior and measurement update computation. These algorithms also use inertial sensors to compute the transitional prior.

Klepal and Beauregard (2008) investigated the use of partial mapping for indoor navigation. They described the scenario of a first responder arriving at an incident with knowledge of only the building footprint. This work demonstrates that the use of only minimal constraints in a particle filter can also be very useful. A weighting model that was used in this work, simply eliminates particles, which crossed walls. The

performance improvement can be obtained if a more subtle weighting approach is applied. Khider et al. (2008) use different movement models to increase the robustness. Among the models were the stochastic behavioral model, the diffusion movement model and map-enhanced combined model. However, elevators or ladders for industrial facilities are not addressed.

Khider et al. (2009) uses 3D diffusion movement models and building plans for pedestrian navigation indoors. He also proposes an intermediate virtual floor between staircases, which he calls x1/2 level, to compute altitude. The new floor level is set depending on the model based information. User motion inside the staircase is represented by an extended Markov model. For this algorithm the map must be known well, since the direction of the stairs helps the map aided algorithm to find a user position inside the staircase. Even the speed is adjusted in the staircase, depending on the user climbing up or down. In this approach the height information is not used although it could increase the robustness of the floor transition estimation.

Kaiser et al. (2011) proposed the use of an angular probability density function for weighting particles within the particle filter. In this work, wall crossing constraints were not the only constraint. First, the particles were weighted according to their direction with respect to an angular movement model, derived from complete mapping of a building. Second, particles, which crossed walls were naturally de-weighted in this model. The authors stated that use of weighting based on particles heading performs better than an equal particle weighting approach, especially when dealing with multiple particle groups. However, this method works well only when the walls are closely spaced. The performance of this method degrades in open areas or when using maps with no internal wall constraints.

In Woodman and Harle (2008); Ascher et al. (2012) the map matching algorithm is also based on particle filters. They extended the algorithm by taking into account multiple floors. In Woodman and Harle (2008) the differential height information of the foot mounted pedestrian navigation system is used to watch the transition from one step to another. For very detailed maps, this approach seems to be a good solution. However, obtaining maps with that level of detail in a real-world mission might be a problem, as every step in a staircase needs to be known; if one step is missing in the map, the proposed map matching algorithm might fail in a staircase. Woodman and Harle (2008) did not address ladders and elevators.

To reduce computation time, the degree of freedom of this 3D problem often is reduced to 2D, so the height of a trajectory is not considered. In Ascher et al. (2012) staircases are represented as sloping rectangles and elevators and ladders as vertical rectangles that can be crossed by the user. The accurate height estimation from IMU and barometer measurements allows to impose an additional constraint for each particle, finally matching the estimated trajectory to the multi-floor map. Even slightly inaccurate height profiles due to barometric drift lead to correct estimation results, which shows the robustness of this approach.

Woodman (2010) investigated the use of building constraints for an unknown initialization and convergence of the path to a uni-modal position solution. In this work, a large number of particles were used, especially at initialisation, with numbers ranging from $2.5 \cdot 10^4$ to $4 \cdot 10^6$. Adaptive resampling was used to vary the number of particles used depending on the complexity of the PDFs, thus fewer particles were used after initialization and in the "tracking" mode. In this work, the filter weighting scheme was based upon the agreement of the height change from the INS with the height change obtained from the map data. However, no attempt was made to investigate cases of incomplete or incorrect mapping.

In Krach and Robertson (2008) the importance of having the map of the environment to reduce position drift of an inertial bases navigation system is pointed out again. Depending on the map and walked path, particle filter based map matching can completely eliminate estimation drift. Even an unknown starting point can be estimated after some time. The indoor maps can be also used for position and heading correction.

Building floor plan data can help not only obtain better positioning but it also can correct inertial navigation system position and heading used under GNSS-denied conditions. The idea of using Cardinal Heading Aided Inertial Navigation (CHAIN) was proposed by Abdulrahim et al. (2011a,b) and later by Pinchin et al. (2012). This algorithm generates heading measurements from the basic knowledge of the orientation of the building in which the navigation system is operating. This measurement is used in an Extended Kalman Filter in the form of an observation of heading error. The solution is based on the assumption that most buildings are constructed with a rectangular layout where most rooms and corridors are also rectangular, thus constraining the direction in which the user can move throughout the building into one of four principal headings. Another assumption is that error in the INS Course Over

Ground (COG) over one step is identical to the error in the INS heading state. A measurement of this error is obtained by differencing the INS COG with the cardinal headings. The minimum difference is taken as a measurement of heading error and weighted before being used as a filter input. The disadvantage of these methods is that the user can actually move in a different direction than the "cardinal heading" of the building. Therefore the method may fail in the following cases (Abdulrahim et al. (2011a)): continuous walking in circles or curvilinear lines for long period of time, if the building does not conform to the simple geometry, and when internal rooms and corridors are not parallel to external walls.

The robustness of CHAIN algorithm can be improved by weighting the heading measurements to reflect the degree to which the assumptions hold in a particular environment. In a tightly constrained environments, e.g., narrow corridors the measurements are highly weighted while in a more loosely constrained environment, e.g., a car parking the observations can be given a lower weight. Since heading is the primary source of position error in a foot mounted INS aided by zero-velocity observations, CHAIN is very effective at controlling position accuracy drift. Tests have demonstrated that an EKF with the CHAIN observations is capable of controlling heading drift over long periods of navigation, keeping position error below 5 m in 40 minutes walk for unaided foot mounted INS (Abdulrahim et al. (2011b)).

Borenstein et al. (2009) introduced a method called "Heuristic Drift Reduction" (HDR). HDR makes use of the fact that many streets or corridors are at least partially rectilinear. At any moment, the HDR method estimates the likelihood that the user is walking along a straight line; if that likelihood is high, HDR applies a correction to the gyro output that would result in a reduction of drift if indeed the user was walking along a straight line. If the algorithm decides that the user is not walking along a straight line, then HDR does nothing. According to Borenstein and Ojeda (2010) the limitation of the HDR method is that when not moving straight, at best we can expect HDR to notice that and suspend its operation. During that time, gyroscope drift accumulates and the integration of the rate of turn results in heading errors. Then, when moving straight again, new heading errors are prevented, but those heading errors that were accumulated while HDR was suspended remain in the system and cannot be eliminated. The effect of the HDR method is thus that it reduces heading errors due to drift substantially, but cannot totally prevent the unbounded growth of heading errors.

Jiménez et al. (2011) analyzed the shortcomings of HDR algorithm, which can even degrade the navigation solution when used in complex buildings where corridors are curvilinear and not aligned to a rectangular layout or where there are large open areas. They proposed a method, called improved Heuristic Drift Elimination (iHDE), that includes a motion analysis block to detect straight-line paths and an adaptive on-line confidence estimator for the heading corrections.

4.2 Novel Map Aided Indoor Navigation Algorithm

The thesis contribution to map aided autonomous indoor navigation is in development of an approach for position and heading correction. A method for preventing the unbounded error growth by correcting position and heading errors of autonomous navigation systems operating indoor has been proposed. This method can be applied to vehicle navigation and it comprises three steps: (a) the autonomous sensor data is processed to obtain position, velocity, and attitude, (b) map matching corrections are applied to the trajectory calculated by the dead reckoning system, and (c) the most accurate estimation of vehicle's path is computed as optimal fusion of map matching and dead reckoning solutions. The proposed algorithm consists of two parts: (a) particle filter based map matching, and (b) novel algorithm for position and heading correction. The map matching algorithm is based on the two dimensional algorithm presented by Woodman and Harle (2008); Beauregard et al. (2008).

The chapter presents also test results with actual digital building floor plan and real-world heading rate and ground speed data. The position measurements were calculated by vehicle dead reckoning system. These measurements were combined with the floor plan to correct the position of the vehicle inside the building. The results shown in this chapter demonstrate that the proposed map-aided navigation algorithm is reliable and accurate. It is able to correct significant errors in dead reckoning position and heading by applying the map constraints.

Woodman (2010) showed that map data can be used to solve the initialization problem by only applying wall constrains on the path. The position solution converges from an unknown position to an actual location. However, this method of initialization often requires large number of particles in the filter and may be impractical for real time applications. For this reason we investigate a more realistic scenario

whereby the initial position and orientation is roughly known and only requires refinement. In many situations, the location of vehicle inside the building is known with some position error, which depends on circumstances, navigation equipment, and scenario. For example, when vehicle enters the building the position can be known because GNSS was available prior to the entrance. In this case, since the position error is relatively small, the particles weight computation may include the condition of wall crossing and also proximity to the position measurement computed by the navigation system.

4.3 Indoor Maps

Currently, there is no common standard for indoor map data. In most practical cases, it is still necessary to compile a digital map suitable for map matching using different sources like 2D or 3D plans in a computer-aided process supervised by a human operator. Most of the databases for the maintenance of buildings are based on a 2D graphical representation inherited from design plans. The content of such a database is composed of many objects (corridor, room, elevator, etc.), which are useful for positioning purposes.

The minimum requirements to the indoor map include information about the walls, doors and staircases. In addition to this, the topological algorithms require information about transitions between rooms and other topological relationships. Gilliéron et al. (2004) proposed to use link/node representation created with the aid of graph theory. An alternative approach, which considers a Voronoi diagram of the environment was proposed by Woodman (2010). A Voronoi diagram consists of a set of points with a locally maximal distance from all surrounding objects. Such points naturally form the edges of a graph-like structure. According to Woodman (2010), Voronoi diagram can make the map matching algorithm more efficient since the particles are more constrained and far fewer are required. The main drawback is that it is not possible for the particle cloud to represent positions that lie near the edge of a room or corridor. However, this limitation may not have a significant effect on the positioning accuracy, especially when the path is staying clear of nearby obstacles.

For unmapped buildings, laser based Simultaneous Localization and Mapping (SLAM) methods for indoor navigation can be used to create 2D or 3D maps. The example of

commercially available mapping system includes the Trimble Indoor Mapping Solution (TIMMS). This technology does not rely on GNSS and can capture the spatial data indoor enabling the creation of accurate, real-life representations (maps, models) of interior spaces and all of its contents. Every object in the interior space, including desks, chairs, stairs, and doors appear in the plan. The created maps are geo-located, meaning that the real world positions of each area of the building and its contents are known. TIMMS is a manually operated push-cart designed to accurately model interior spaces without accessing GNSS. It consists of 3 core elements: LiDAR and camera systems engineered to work indoors in mobile mode, computers and electronics for completing data acquisition, and data processing workflow for producing final 2D/3D maps and models.

Currently available indoor map databases include the indoor OpenStreetMap (OSM) and Google indoor map. Indoor OSM utilizes existing OSM methodologies (nodes, ways, relations and keys) and include the following features: mapping of indoor spaces including different levels (floors), mapping of doors and windows (inside as well as facade), 3D properties (e.g. height etc. are also included). Currently there are thousands of locations available including airports, hotels, universities, schools, museums, train stations, shopping malls.

Google indoor maps include over 10,000 locations available around the world and the number is growing. Indoor Google Maps create a more convenient and enjoyable visitor experience. Visitors can access a building's floor plan when indoor maps are available. For buildings with multiple floors, visitors can switch between floors to see the respective layouts. Floor plan labels help visitors easily find different stores within shopping malls, departments within retail stores, gates within airports, as well as ATMs and restrooms. Visitors can spend more time enjoying their experience, discover new points of interests, and avoid time spent searching for building directories.

4.4 Particle Filter Based Map Matching for Indoor Navigation

The map matching algorithm is an essential part of the proposed concept. Constraining the estimated user trajectory onto a building plan leads to strongly non-Gaussian and possibly multimodal distributions, which makes application of the common Kalman filter, or its nonlinear extensions, impractical. Although map matching has been shown to be possible using Kalman-type filters (Perälä and Ali-Löytty (2008)), we

will use an approach based on particle filtering similar to the algorithm that was described in Chapter 3.

The goal of the particle filter is to find an approximation for the *posterior* distribution $p(x_{0\ldots k}|y_{1\ldots k})$, i.e., the conditional distribution of the states x at time instants $0,1,\ldots,k$ given the sequence of observations $y_1,\ldots,y_k$. Suppose that the state can be described as a discrete-time Markov process such that the state at time step k depends on the previous state according to the probabilistic model $p(x_k|x_{k-1})$. Then, we can decompose the conditional probability of the states given the measurements y similar to Eq. (3.6)

$$p(x_{0\ldots k}|y_{1\ldots k}) \propto p(y_k|x_k)p(x_k|x_{k-1})p(x_{0\ldots k-1}|y_{1\ldots k-1}). \tag{4.1}$$

Because the distribution in (4.1) cannot be solved for in closed form in the general nonlinear and non-Gaussian case, a particle filter will approximate it by using a cloud of *particles*. The ith particle is a candidate state vector $x^{(i)}$ and has a weight $w^{(i)} \in [0,1]$. The propagation of i th particle can be described by the following equation

$$\begin{bmatrix} N_{k+1}^{(i)} \\ E_{k+1}^{(i)} \\ \psi_{k+1}^{(i)} \end{bmatrix} = \begin{bmatrix} N_k^{(i)} \\ E_k^{(i)} \\ \psi_k^{(i)} \end{bmatrix} + \begin{bmatrix} (L_k + \Delta L_k^{(i)})cos\psi_k^{(i)} \\ (L_k + \Delta L_k^{(i)})sin\psi_k^{(i)} \\ \Delta\psi_k + \Delta\psi_k^{(i)} \end{bmatrix} \tag{4.2}$$

where the distance traveled from time instance t_k to t_{k+1}, L_k and increment in vehicle's heading during this time $\Delta\psi_k$ are measured by the vehicle's navigation system. For each particle the distance traveled by this particle $\Delta L_k^{(i)}$ and heading $\Delta\psi_k^{(i)}$ are perturbed using some model of measurement errors. In our case they are drawn from Gaussian distribution with zero mean and constant variance. The variance of this distributions for both distance and heading is one of the design parameters, which affects diversity of the particles.

4.4.1 Options for Measurement Update

Most of the previous papers include only the measurement update based on building floor plan constraint (impassable walls). In this case, the idea of the map matching algorithm is to fit trajectory into the building plan. If the trajectory is long enough and it includes turns the process of fitting this trajectory into the map will be unique and it will result in a trajectory, which is close to the user's actual path. However, in some cases this can create the ambiguity and a solution may be not unique.

It is not always possible to determine the user's position because the user has not yet passed through a sufficient number of asymmetries, or because the environment is not sufficiently asymmetric to make localization possible. Typically, buildings exhibit high degree of symmetry. In particular, it is common for multi-storey buildings to have very similar layouts on each floor. This limitation of particle filter based map matching algorithms was outlined in Woodman and Harle (2008); Woodman (2010).

In such cases it is necessary to provide more information to the filter to allow it to complete the localization process. One option is to use some form of absolute positioning to obtain an approximate position for the user. Unfortunately, this is not always possible. If the positioning provided by the dead reckoning is accurate (position error is smaller than 10 m) then this information can be also used for measurement update.

When only building plan information is used as a measurement update it can be incorporated into map matching algorithm through the weight computation for each particle according to Eq. (3.7). In this case, the measurement likelihood $p\left(y_k|x_k^{(i)}\right)$ is defined as

$$p\left(y_k|x_k^{(i)}\right) = \begin{cases} 0 & \text{if there is a wall between } x_{k-1}^{(i)} \text{ and } x_k^{(i)} \\ 1 & \text{otherwise.} \end{cases} \tag{4.3}$$

In other words, particles that cross walls are discarded. Alternatively, the likelihood of wall-crossing particles can be set to a small positive number in order to account for possible errors in the map. However, in the tests conducted in this study, zero likelihood was used for these particles.

Another option is to consider in addition to map information the measurements of the user's position from the on-board DR navigation system. In this case, the likelihood is calculated for each particle based on the proximity between the position fix and the particle according to

$$p(y_k|x_k^{(i)}) \propto \left\{ -\frac{\left\| x_k^{(i)} - x_k^{meas} \right\|^2}{2\sigma_{pos}^2} \right\} \tag{4.4}$$

where $x_k^{(i)}$ is the ith particle's coordinate, x_k^{meas} is the measured user position, and σ_{pos}^2 is the position measurement variance. In addition to the measurement update the condition of not crossing the walls is also checked. The path increment $x_k^{(i)} - x_{k-1}^{(i)}$

for each particle is checked for crossing the walls. The particle's weight is reduced to zero according to Eq. (4.3) if it crossed the wall. The second option requires fairly accurate position measurements with the position error not exceeding $10-15$ m. In this case the navigation algorithm can keep track of the user's position and heading, and improve the dead reckoning solution applying map constraints. Correction of the dead reckoning solution is based on the approach, which will be presented in Section 4.5.

4.4.2 The Map Matching Algorithm Indoor Test

The proposed navigation algorithm was tested with actual digital building floor plan and real-world heading rate and ground speed data. The test vehicle was a four-wheeled cart described in Pekkalin et al. (2010) that was pushed around in an office building environment. A 15 deg/hr Murata SCR1100-D04 MEMS gyroscope was measuring the heading changes of the cart while the speed was measured using two encoder-based odometers with separate measurement wheels. The encoders were thus not connected to any of the cart wheels, but the measurement wheels were mounted using spring-loaded arms in order to ensure a constant floor contact and to avoid slippage. Both encoders were available, but only one of them was used to compute the results presented in this paper.

The test route used here was a 150 m path inside a building shown in Fig. 4.1. The true vehicle path is very close to a map matching solution shown by the crosses. The estimated path computed by the vehicle dead-reckoning system is shown by the solid line. The initial heading error of 10 deg was intentionally added to the dead reckoning solution. The dead reckoning solution is also used as the position measurement for the particle filter. The speed and ground track are also computed based on the dead reckoning solution.

The particle filter solution is shown in Fig. 4.1 by the crosses. It is represented by the weighted mean of the particles. The particle filter calculations are based on 100 particles. The algorithm position accuracy was analyzed against known landmarks along the test route. The DRMS of horizontal position errors for dead reckoning only and map aided dead reckoning navigation algorithms during the test is shown in Fig. 4.2. From Figs. 4.1, 4.2 it is clear that the knowledge of floor plan can improve the accuracy of position estimation calculated by the dead reckoning system.

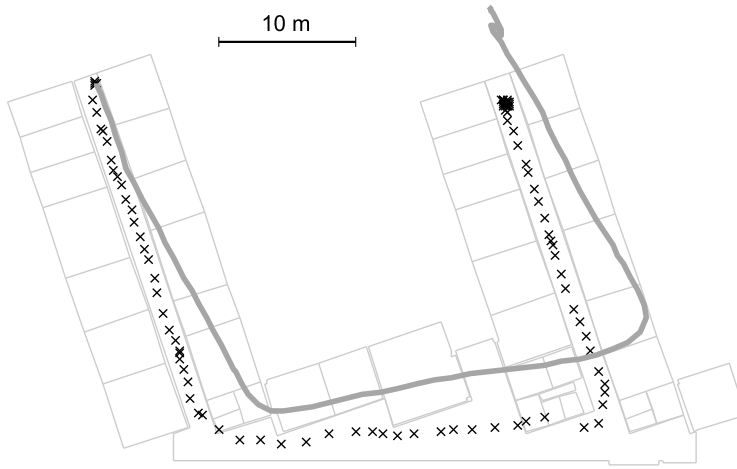

Fig. 4.1. *Dead reckoning solution (thick line) and position estimates from map matching (crosses).*

From these results, it can be seen that the position error of the particle filter solution is smaller than the width of the corridors and doorways. The cross-track position error during this test did not exceed 1.7 m. The approximate limit on the possible position accuracy of this approach is set by the size of the rooms and level of floor plan details. This section has shown how the map matching algorithm can improve the dead reckoning system performance. This becomes very important in the case of autonomous navigation. The dead reckoning solution can be corrected occasionally. Section 4.5 will describe how this correction can be calculated based on the map matching and dead reckoning solutions. The performance of the proposed particle filter based map matching algorithm depends on the following factors:

- Vehicle's movement. The long walking path covering different rooms of the building improves the accuracy of position estimation.
- Size of the rooms and hallways affects the accuracy. The smaller the dimensions the better accuracy can be achieved.

The particle filter performance can be adjusted by changing the ground speed and heading noise in the proposal distribution $p(x_k|x_{k-1})$. Increasing ground speed noise improves the particles diversity. Position and heading variances have to match approximately the position and heading measurement errors of onboard sensors. Using

inertial measurements and building plans only makes the process of positioning entirely autonomous and gives promising results. This method of positioning can be applied to vehicle or pedestrian navigation tasks. In particular, it suits the needs of firefighters and rescue services. The algorithm is suitable for real-time implementation on personal navigation devices.

4.5 Position and Heading Correction

This section describes the fusion algorithm that combines navigation data with building floor plan. The output of this algorithm is the optimal estimation of navigation system position and heading errors.

4.5.1 Motivation

The proposed algorithm provides an accurate means of correcting the accumulation of dead reckoning position and heading errors. The idea is based on the fact that the vehicle's movement indoors is constrained by the walls; if the trajectory is long enough and includes turns, it will be quite unique and it can be identified among

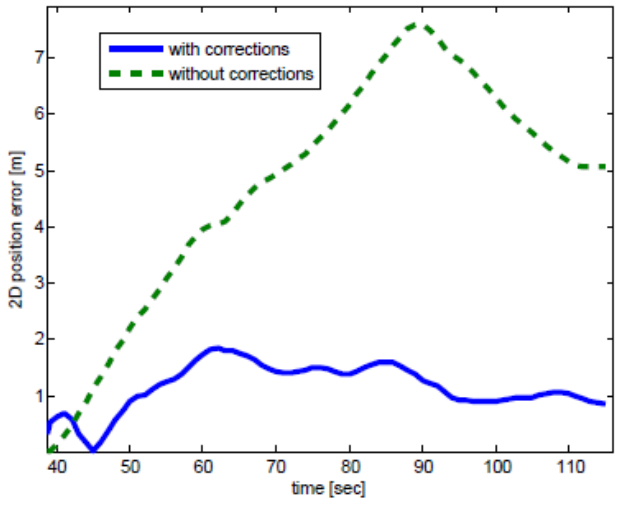

Fig. 4.2. *DRMS of horizontal position errors.*

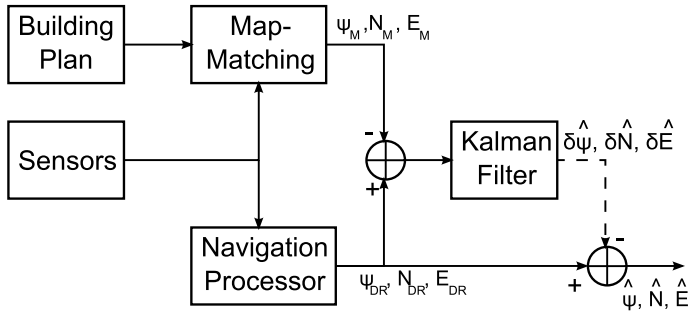

Fig. 4.3. *Block diagram of the feedforward implementation of the algorithm.*

other trajectory candidates. The position and heading extracted from this trajectory are usually more accurate than those computed by the dead reckoning system since they do not suffer from error accumulation which is inherent to dead reckoning navigators. The different error properties and little correlation between these two navigation solutions can be explained by the fact that building floor plans represent a new source of information and map matching is a non-linear operation.

An example of a typical dead reckoning solution for indoor vehicle navigation is shown by a thick solid line in Fig. 4.1. It can be seen that the solution does not match the corridors of the underlying map because of a heading offset of approximately 10 deg. The position accuracy of this solution was significantly improved after applying a map matching algorithm (Fig. 4.2), which is mostly due to the elimination of the heading error; the map-matched position estimates are shown by crosses. This example demonstrates the capabilities of map matching in improving the accuracy of dead reckoning navigation. However, the dead reckoning solution is smoother than its map-matched counterpart which shows larger short-term variations. Therefore, a better estimate of the trajectory is obtained by fusing these two solutions. Since the errors in the two estimates have complementary properties, the fusion algorithm can be, for example, a Kalman filter.

4.5.2 Fusion algorithm

A block diagram of the proposed navigation algorithm is shown in Figs. 4.3, 4.4. In these diagrams, N and E refer to North and East position, respectively, and ψ is the heading; the subscripts M and DR refer to quantities obtained from map matching and dead reckoning, respectively. $\delta\hat{\psi}$, $\delta\hat{N}$, $\delta\hat{E}$ are estimates of the dead reckoning

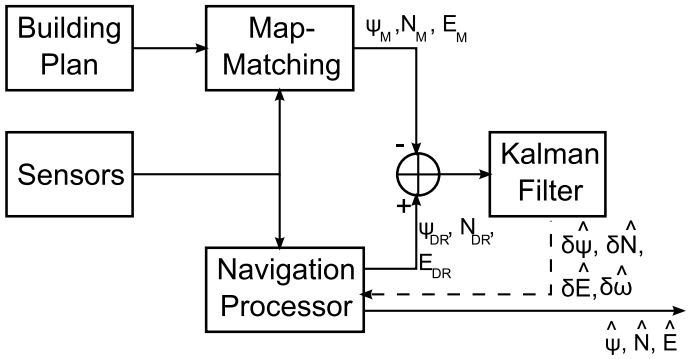

Fig. 4.4. *Block diagram of the feedback implementation of the algorithm. All corrections fed back to the navigation processor.*

system's heading and position errors computed by the Kalman filter. The navigation solution is corrected using the computed estimations of the position and heading errors.

The proposed algorithm consists of the following steps. First, the autonomous sensors data is processed to compute vehicle's position and heading. Then, the map matching algorithm is applied; we used the particle filter based approach, which was described in Section 4.4. Once the map matching has been applied, the resulting trajectory is more accurate than the original dead reckoning solution since constraints imposed by the building walls are accounted for. Finally, the position and heading from the map matching solution is compared with position and heading computed by the dead reckoning system by taking the difference between the respective parameters. This difference is fed to the Kalman filter as a measurement.

If autonomous navigation is performed for a long period of time the navigation parameters of the dead reckoning system can be also corrected. In this configuration, the estimated dead reckoning errors $\delta\hat{\psi}$, $\delta\hat{N}$, and $\delta\hat{E}$ along with an estimate of the gyroscope bias $\delta\hat{\omega}$ are computed by the Kalman filter and fed back into navigation computer to compensate for the gyro measurement errors, as illustrated in Fig. 4.4. The rate at which this correction can be applied depends not only on accuracy of the sensors but also on the building layout. For example, long narrow corridors constrain the heading efficiently and result in a more accurate map matching solution. Since with our sensors, the influence of odometer errors on navigation solution is negligible compared to influence of gyro errors, the estimation of odometer measurement errors were not used to correct the output of odometer.

The Kalman filter is implemented as an error-state filter. The state vector includes the dead reckoning system's North and East position errors, ground speed and heading error along with the gyro drift and odometer scale factor error. The system model is approximated by the linearized error equations for dead reckoning navigation systems based on speed sensor and heading gyro, expressed in continuous time as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\delta N\\ \delta E\\ \delta V\\ \delta S\\ \delta\psi\\ \delta\omega\end{bmatrix} = \begin{bmatrix}0 & 0 & \cos\psi & 0 & -V\sin\psi & 0\\ 0 & 0 & \sin\psi & 0 & V\cos\psi & 0\\ 0 & 0 & 0 & V & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & -1/\tau_g\end{bmatrix}\begin{bmatrix}\delta N\\ \delta E\\ \delta V\\ \delta S\\ \delta\psi\\ \delta\omega\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ 0\\ 0\\ n_g\end{bmatrix} \tag{4.5}$$

where V denotes the vehicle speed, δS is odometer scale factor error and τ_g is the correlation time of the gyroscope bias. n_g is a zero-mean random variable with variance matching the instability properties of the gyroscope. An estimate of the position and heading errors is computed by taking the difference between the position and heading computed by the dead reckoning system and the position and heading from the map matching solution. Hence the measurement model is given by

$$\mathbf{z} = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\delta N\\ \delta E\\ \delta V\\ \delta S\\ \delta\psi\\ \delta\omega\end{bmatrix} + \mathbf{v} \tag{4.6}$$

where $\mathbf{v}$ is a white Gaussian measurement noise with zero-mean and known covariance which is not necessarily constant because the precision of map matching depends on the local geometry of the building. Although the presented algorithm is based on two-dimensional dead reckoning navigation system, it is possible to modify this approach to accommodate other types of navigation systems, for example, a six degrees-of-freedom INS.

4.5.3 Field Tests and Results

To test and validate the proposed position and heading correction algorithms we used the same heading rate and ground speed data, which was described in Section 4.4.2.

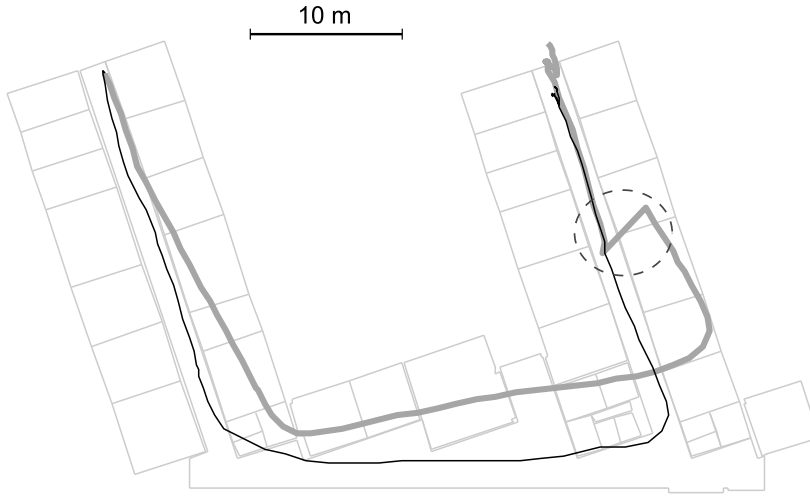

Fig. 4.5. *Dead reckoning solution (thick line) and Kalman filter corrected trajectory (thin line). The Kalman filter solution is fed back to the dead reckoning processing once (dashed ellipse).*

The purpose of these test is to show how the same DR solution can be improved after position and heading corrections are applied. The field test demonstrates how the initial heading error in the dead reckoning computations can be estimated and corrected by using the proposed algorithm. The test route is shown in Fig. 4.5. The actual path followed the marks on the floor and was very close to the path shown by the thin line. The initial heading error of 10 deg was intentionally added to the computed dead reckoning navigation solution which is shown in Fig. 4.5 by a thick line. The test begins with a stationary period of approximately 38 sec. During this time the heading estimation is not possible. Then the cart is moved and the estimation of heading began. The heading error estimation performance is illustrated in Fig. 4.6. It shows that approximately 20 sec after the cart moved the Kalman filter estimation of heading error converged to the actual heading error of 10 deg. After the transition time the uncertainty of heading estimation error did not exceed 1 deg. Note that the estimation performance depends on the building layout and trajectory of the vehicle which set the limit for the accuracy of position and heading error estimation. The position accuracy during this test was approximately 1.5 m and DRMS of horizontal position errors is shown in Fig. 4.2. The optimal estimation of the vehicle's path is computed by correcting the dead reckoning solution using computed by the Kalman

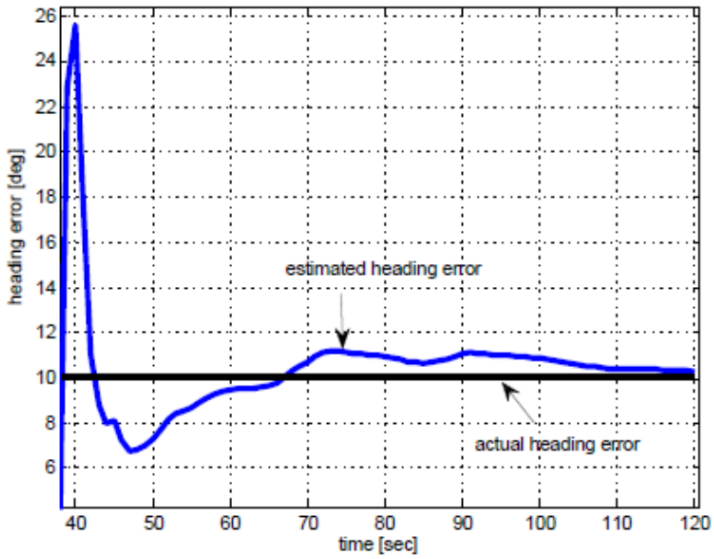

Fig. 4.6. *Estimated vs. actual heading error.*

filter corrections and it is shown in Fig. 4.5 by the thin line.

During the first minute of vehicle's movement the Kalman filter only estimated the number of parameters including position and heading errors of the dead reckoning computations without correcting these errors. Nevertheless, it is possible to correct the dead reckoning solution by eliminating position and heading offsets as it is shown in Fig. 4.5. If the autonomous navigation continues for long time the correction of the dead reckoning computation can be repeated when the accumulated errors become large and map correction can improve the accuracy of navigation. In this way it is possible to keep the positioning errors bounded.

4.6 Conclusions

This chapter has shown how the map corrections can improve performance of autonomous dead reckoning navigation system by offsetting the initial errors as well as accumulated errors of the dead reckoning system. The approach can be applied to vehicle navigation systems operating indoors. Using only self-contained sensors and building plans makes the process of positioning entirely autonomous and gives promising results. The assumption of indoor map availability is justified since nowadays in many

countries the digital building floor plans are mandatory for critical infrastructure and in the future they will be available for most of building. Fire-fighters, rescue services and police have real-time access to this information.

The map matching algorithm generates position and heading pseudo-measurements from knowledge of the building floor plan where the navigation system is operating. Using these pseudo-measurements the Kalman filter calculates errors in dead reckoning system's position and heading. Then these estimated position and heading corrections are held constant and are used to correct the dead-reckoning solution until the next time they are recalculated (typically 40-60 seconds). If the building layout and movement of the vehicle are suitable it is possible to keep small position errors and heading errors for long periods of time. The performance of the proposed navigation algorithm depends on the following factors:

- Vehicle trajectory. The long path covering different rooms of the building improves the accuracy of position estimation.
- Size of the rooms and hallways affects the accuracy. The smaller the dimensions the better accuracy can be achieved.
- The initial position and heading errors of the dead reckoning system. The large errors might cause the map matching algorithm to fail.
- Quality of a gyro and ground speed sensor.

Compare to other approaches for aiding IMU with building heading (Borenstein et al. (2009); Borenstein and Ojeda (2010); Abdulrahim et al. (2011a,b); Jiménez et al. (2011); Pinchin et al. (2012)) the proposed method has the following advantages:

- It can work when the building does not conform to the simple geometry
- It allows continuous walking in circles or curvilinear lines for long period of time
- It can correct not only the heading errors but also position errors as well

5. A NOVEL APPROACH TO AUTONOMOUS PEDESTRIAN NAVIGATION

High-performance autonomous pedestrian dead-reckoning (PDR) systems usually include 6 degrees-of-freedom (DOF) inertial measurements unit (IMU) to calculate position of the user. These systems do not rely on GPS signals or preinstalled infrastructure such as RF beacons, Wi-Fi routers, ultrasonic transmitters etc. Standard inertial navigation system (INS) calculates position by temporal integration of IMU data that comes from three accelerometers and three gyroscopes. Estimated position is calculated at regular time intervals. The unaided INS's position, velocity and attitude errors grow with time and can be quite significant especially when IMU consists of low-cost MEMS sensors. Therefore, traditional unaided INS mechanization is impractical for pedestrian navigation.

One way to curb the divergence of errors in INS is to use external velocity and position aiding. In indoor scenarios without preinstalled infrastructure, the options for external velocity and position aiding are very limited: building floor plans can be used for position update, Doppler radars for velocity update. However, even these options are not always available.

An alternative navigation method takes advantage of biomechanics of walking. Recognizing that people move one step at a time, the pedestrian mechanization restricts error growth by propagating position estimates in a stride-wise fashion, rather than on a fixed time interval. Inertial sensors in a pedestrian dead reckoning system are used to detect the occurrence of steps, and provide a means of estimating the distance and direction in which the step was taken. In this way, position error is proportional to the number of steps or traveled distance.

Most current PDR systems use foot mounted IMU, which allows for zero-velocity update at every stride. However, the foot-mounted IMU is not practical in many applications. In this thesis we propose a novel approach for velocity update based

on knowledge of human walking process. It can be applied to body-mounted IMU at waist or torso, which in most applications is advantageous compare to foot-mounted sensors.

Our approach gives an alternative way to calculate traveled distance and velocity averaged during the step, which can be considered as a virtual measurement. The different characteristics of errors in INS output and in this virtual measurement make it possible to apply complementary filter methodology and significantly improve INS performance by keeping the horizontal velocity and tilt errors small. The processing of corrected IMU output results in accurate estimation of stride length and direction. This chapter presents the real-world results from pedestrian indoor walking tests.

5.1 State of the Art Methods

There are several approaches to the use of the inertial sensors for pedestrian navigation, which according to Groves et al. (2007) may be characterized by

- The location, whether the sensors are mounted on the shoes or the body;
- The number and quality of inertial sensors to be used;
- Whether to use conventional inertial navigation algorithms, supported by zero velocity updates (ZUPT), pedestrian dead reckoning (PDR) or both.

All recent developments in navigation systems for pedestrians use either body mounted or foot mounted inertial sensors. In both cases, the navigation algorithm can be based on traditional INS mechanization with ZUPT or PDR, which is a step and heading type algorithm. There are also examples (Groves et al. (2007); Soehren and Hawkinson (2008)) when inertial navigation and PDR are used together, sharing the same inertial sensors, in which case, inertial navigation is incorporated within the multi-sensor integration architecture as the reference system and PDR as an aiding sensor. In spite of the fact that PDR systems are based on very low cost sensors they perform much better than a traditional INS with the same sensors. This is because of applying additional constraints on the user movement, which are derived from the models of human gait.

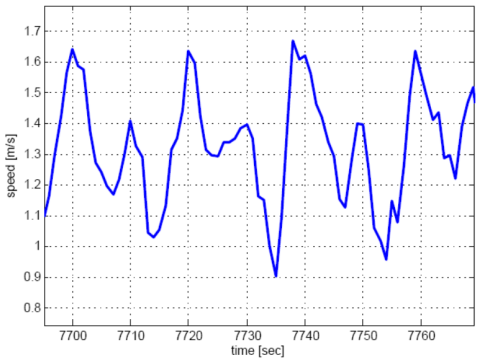

Fig. 5.1. Horizontal speed during walking.

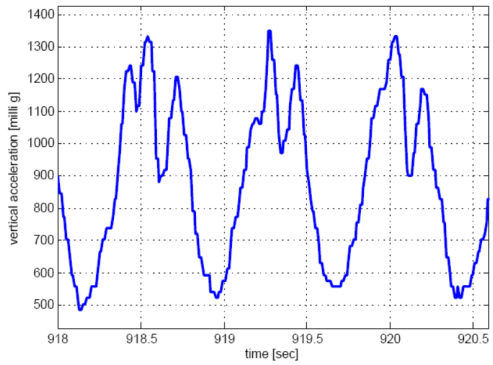

Fig. 5.2. Vertical acceleration during walking.

5.1.1 Biomechanics of Walking

Analysis of the human gait shows that the basic pattern of human motion during a walk is cyclical, repeatable and remarkably consistent between individuals (Stirling et al. (2003)). At normal constant walking speed, the vertical and horizontal components of the body's centre of gravity velocity oscillate smoothly with the frequency equal to the step frequency. Examples of horizontal speed and vertical acceleration during normal walking are shown in Figs. 5.1, 5.2. The vertical acceleration was recorded by the torso mounted accelerometers and the horizontal velocity was measured by a Novatel DGPS receiver.

The previous work related to human gait and motion analysis provides significant insight into biomechanics of walking. Most of the existing research was carried by the medical community for purposes of orthopedic surgery, prosthetic limb design and treatment of neuromuscular disorders (Aminian and Najafi (2004)). Additionally, there is an intense effort at understanding human biomechanics and bipedal motion using MEMS inertial sensor technology. For example, Kwakkel et al. (2007) evaluated the performance of MEMS IMU technology to analyze in situ foot/ankle kinematics. Taki et al. (2004) investigated bipedal motion for the purpose of designing bipedal robots.

Matthews et al. (2010) developed a mathematical model of the human walking pattern, or gait, and determined what kind of navigation information can be extracted. A related application of this model is in the field of biometrics where a gait model can be used to characterize the motion of different individuals, so called biometric

gait recognition. He suggested that the natural walking pattern of a human may vary slightly depending on the situation. For example, people walk differently while in a hurry than while on a leisurely stroll. Additionally, a person's size, mass, level of fitness, etc. have a significant effect on their gait profile. For example, an athlete walks differently from a non-athlete; taller persons have a longer natural stride length;

According to Matthews et al. (2010), consistent motion patterns exist for most if not all modes of bipedal motion such as walking and running. An individual's personal walking pattern is a variation on this overall basic pattern. The existence of a personal walking pattern that is similar across individuals means that a single model only needs to be developed that includes a set of parameters such as limb lengths which can be tuned to accommodate different individuals. Identifying these patterns alone can provide meaningful information to the personal navigator.

The kinetic models developed by Ladetto (2000); Groves et al. (2007) can be used to provide a real-time estimate of stride length. These models use an empirical relationship between a step length and step frequency and take into account individual differences between people. Therefore, they can potentially augment other dead reckoning algorithms commonly used in personal navigation.

Many kinetic models use a single-axis accelerometer to estimate stride length in-situ for straight forward walking. The amplitude of vertical acceleration and vertical displacement are proportional to the step length. As a result, the stride events as well as stride length can be estimated using models of gait dynamics and measured motion parameters. When a foot hits the ground significant vertical acceleration is generated by the impact. Therefore the algorithms for stride occurrence detection can be based on the analysis of the acceleration pattern during walking. Furthermore, the combination of the kinetic model and accelerometer yields a navigation solution of comparable or better performance when compared to the step counting approach.

Average speed during the step can be calculated as the ratio between the step length and step duration. This ground speed can be also resolved into the north and the east velocity components using INS computed heading averaged over the same step. Thus the estimation of pedestrian velocity during the step is obtained. Since the velocity error of this kinetic model based estimator is a result of several independent error sources such as model error, measurement error and inaccuracy in step occurrence events it has a random distribution and can be approximated as Gaussian noise.

5.1.2 Pedestrian Dead Reckoning Systems

A PDR system computes position by integrating the displacement vectors, which represent steps. Most personal dead-reckoning systems detect steps using an accelerometer and move the position estimate forward by the step length in the direction determined by a magnetic compass or yaw gyro (Judd (1997); Ladetto and Merminod (2002)). If no steps are detected, the system is assumed to be stationary. Any PDR system performs the following tasks (Harle (2013)):

- Step count or step segmentation
- Step length estimation
- Step direction estimation

The first task of a PDR is the identification of steps or strides within the data. Simple PDR algorithms only count the steps assuming that the step length is just the average for that user. Advanced systems also perform accurate step segmentation and have the ability to analyze the accelerometer signals to estimate the magnitude of each step individually. After step segmentation is complete next task is to estimate the step length and direction. Step length estimation is usually based on the accelerometer signals. The body-mounted accelerometers detect steps using the vertical component of the acceleration vector, which exhibit cycles typical of a human's walking motion. Some algorithms model step length as a function of the step frequency, acceleration variance and slope (Ladetto (2000)). PDR system can work with a single accelerometer, though better accuracy and robustness are obtained with triaxial sensors. The performance is largely insensitive to the sensor quality, so PDR is suited to operation with very low-cost sensors. However, most of these systems require calibration to an individual user because everyone's gait has different acceleration profiles. The step direction is usually determined based on magnetometers or gyros, or both.

Step Segmentation Methods

Most step segmentation methods utilize the fact that pedestrian motion has a cycling nature. The algorithms search for the repeating data patterns. According to Harle (2013), most step segmentation methods are based on step cycle detection and can be

categorized into following groups: zero crossings, peak detection, auto-correlation, and spectral analysis.

- In a zero-crossing algorithm, the decision on step being made is done by analyzing the sign of magnitude of the acceleration vector, which can be computed by subtracting the local gravity from the measured magnitude of the specific force. When the sign changes from negative to positive, a new step is counted. The zero-crossing method is based on the cyclic movement of the human body and requires a triaxial accelerometer. Since the norm of the acceleration vector is used the orientation of the sensor unit has no effect on the measurement. This is a popular choice for pedometers or activity monitors due to its simplicity (Käppi et al. (2001); Leppäkoski et al. (2002); Weinberg (2002); Saarinen (2009)).

- The body-mounted systems can detect the peaks (maximum or minimum detection) of the vertical acceleration caused by heel impacts (Ladetto (2000); Fang et al. (2005)). The difficulty with this approach is that each foot impact may generate multiple local peaks. This can significantly increase the algorithm complexity, especially, in the case of foot mounted sensors due to the higher forces resulting in sensor bounce.

- Similar to peak detection, auto-correlation based algorithms detect peaks in the mean-adjusted autocorrelation of a sequence of magnitude of measured acceleration vector. This approach can be used for body-mounted or foot-mounted IMU. If a sample sequence of walking data for the same person has previously been collected, cross-correlation with this ”template” data can also identify steps or strides using the same process (Harle (2013)).

- Another version of peak detection algorithm involves computing the frequency spectrum of the cyclic data and identifying strong peaks at typical stepping frequencies (Ladetto (2000)). In this approach the subsets of the data (with a width that includes at least two cycles) are converted to the frequency domain and the dominant frequency taken as the walking frequency (Judd (1997)).

It should be noted that most implementations claim to use only the vertical acceleration, but do not compensate for changes in the global pose of the sensor during a step. Instead they assume that one of the accelerometer axes remains vertical throughout.

This assumption is valid, in particular, for inertial sensors attached to the torso of the body. Another common assumption that most algorithms are developed and tested for its operation on flat surfaces. This is appropriate for the majority of buildings. However, Ladetto (2000) reports that the assumptions used by many PDR systems break down on inclines of 10% or more. More recently, Wang et al. (2009) have demonstrated that different gait patterns corresponding to different inclines can be distinguished autonomously with accuracy exceeding 90%. From this we can conclude that a modified PDR system could cope with long ramps such as those for wheelchair access, which can be found in many buildings.

Step Length Estimation

Step length can be defined as the distance from the heel print of one foot to the heel print of the other foot. This is the distance traveled forward by a single leg. Stride length can mean the distance traveled by the heel of one foot to the next time that same foot strikes down – in other words, two steps, since in that time the other foot has also touched down once. During symmetric walking stride length is approximately twice the step size, therefore the terms stride length, step length and step size can be used interchangeably in most of the cases related to pedestrian navigation. The stride length depends on several factors such as walking velocity, step frequency and height of walker etc.

The algorithms for step length estimation generally fall into two groups: (1) algorithms based on biomechanical models and (2) algorithms based on empirical relationships. The example of the step length estimator based on biomechanical model is a kneeless biped, which is modeled as inverted pendulum with leg length l, and vertical displacement of the center of gravity h. In this case, the estimation of step length is given by Jahn et al. (2010)

$$\Delta L = K\sqrt{2lh - h^2} \tag{5.1}$$

where K is a calibration constant. Body-mounted accelerometers can detect maximum or minimum of the vertical acceleration, which corresponds to the step occurrences. Empirical relation between the vertical acceleration and the step length is given by Weinberg (2002)

$$\Delta L = K\sqrt[4]{a_{max} - a_{min}} \tag{5.2}$$

where a_{max}, a_{min} are the maximum and minimum values of the vertical acceleration during the step. This formula is based on the bounce movement of the hip while walking. These algorithms show good performance during normal walking on flat terrain. According to Fang et al. (2005), Eq. 5.2 computes the step length with estimation error of 3% of traveled distance for the same subject and 8% across the variety of subjects. The performance of these algorithms degrades rapidly when a person is walking, for example, on non-flat terrain or climbing on stairs.

Ladetto (2000) proposed the formula for step length estimation using acceleration in the direction of movement measured by a body-mounted IMU. In this work, it was shown that there is a strong correlation between the step length and step frequency. Estimation of the step length was performed using

$$\Delta L = K_1 + K_2 f + K_3 Var + w \tag{5.3}$$

where K_1, K_2, K_3 are precomputed parameters, f is a step frequency, Var is a variance of the measured acceleration, and w is a white noise. Ladetto (2000) also analyzed the variation in step length by experimenting with 20 persons who walked with a constant frequency. The results indicate that accuracy of step length estimation computed by Eq. 5.3 varies with step frequency from 15% (60 steps/min) to 4% (130 steps/min). From the same tests he concluded that there is a correlation between the step length and step frequency. For example, a mean length of 60 cm was obtained with 60 steps/min and one of 90 cm was obtained with 130 steps/min.

Estimation of the Step Heading or Change in Heading

In personal navigation indoors the main source of error in position comes from the errors in the determination of the azimuth of walk, which is usually computed based on gyros and/or magnetometers. Magnetometers measure the absolute azimuth with respect to the magnetic north. The main drawback of the magnetic compass is unpredictable perturbations of the magnetic field caused by the disturbances, which are usually high indoors because of electric fields and steel structures. Gyros measure a change in heading and, therefore, require initialization. The drawback of a gyroscope is that the angular error accumulates. Thus the update of the azimuth computed by the gyro with external information is necessary. The frequency of this update depends on the gyro's quality. However in the short term, the gyroscope can provide a reliable measure of the change in azimuth.

Combining a gyro and a magnetic compass can be beneficial. The magnetic compass is able to provide the external azimuth to update the parameters of the gyro and the gyro can be used to detect the disturbances that can be as large as tens of degrees. Ladetto and Merminod (2002) proposed using Kalman filter for coupling a magnetic compass with a low-cost gyroscope. In this case, the advantage of one device can compensate the drawback of the other. If we compare the rate of change of both signals while measuring the strength of the magnetic field, it is possible to detect and compensate magnetic disturbances. In the absence of such disturbances, the continuous measurement of the azimuth allows to estimate and compensate the bias and the scale factor of the gyroscope. The reliability of indoor and outdoor navigation improves significantly thanks to the redundancy in the information.

PDRs with one gyro, which measures heading and attached to a torso, always apply the step motion in the forward direction determined by the body-mounted sensor. Even if a person makes step in a different direction, the position will be propagated in the direction where the torso faces. Some implementations attempt to identify backward or sideways steps by the acceleration profiles, but they can never determine the exact direction of individual steps as precisely as the 6 DOF IMU.

5.1.3 Foot-Mounted IMU

At each step the foot-mounted IMU is temporarily motionless and the sensor velocity in the local frame is zero. This can be utilized as a pseudo-measurement of INS velocity in the Kalman filter to update state errors, which is usually called ZUPT. The application of ZUPT means that unaided INS computations only occur during the swing phase of the foot to which the sensor is attached. For such short durations, drift accumulation is small and error grows much slower than in the case of standard unaided INS.

Foxlin (2005) mentioned that the first implementation of foot-mounted sensors for navigation was done for a DARPA project in 1996. It was proposed using foot-mounted inertial sensors with zero-velocity updating, but results were never published. Stirling et al. (2003) described an experiment using a prototype foot-mounted sensors that measure stride length with accelerometers and direction with magnetometers. Instead of gyros, their system measures angular acceleration using pairs of accelerometers. They also did not use a Kalman Filter to make optimal use of zero-

velocity updates; the system simply stops integrating and resets the velocity before each step. Stirling et al. (2003) reported that the error in traveled distance for this system is about 10 to 20%.

Foxlin (2005) was the first who introduced ZUPTs as measurements into the EKF instead of simply resetting the velocity to zero in the shoe-mounted INS. He achieved good performance with small low-cost MEMS gyros with the drift of about 100 deg/hr. He has confirmed experimentally that operating this foot-mounted INS with ZUPTs alone results in good short-term navigation performance but gradually loses horizontal position accuracy because of heading drift.

In addition to horizontal velocities, the EKF is also able to correct pitch and roll using the fact that tilt errors are correlated with horizontal velocity errors through the system dynamics matrix. At certain conditions accelerometer and gyro biases can also be corrected. Yaw (heading) and the yaw gyro bias are the only important EKF states that are not observable from zero-velocity measurements.

Foxlin (2005) explained how EKF uses ZUPT pseudo-measurements to correct the position drift that occurs during the stride phase: ”EKF tracks the growing correlations between the velocity and position errors in certain off-diagonal elements of the covariance matrix. For example, at the end of a stride, a high correlation between the uncertainty in north velocity and the newly accumulated uncertainty in northing position will exist. If the ZUPT indicates that the velocity error at the end of the stride was positive in the north direction, the EKF knows that it has been drifting north and will correct the position to the south and the velocity toward zero”.

Jiménez et al. (2010) tried to improve the algorithm described by Foxlin by reducing the gyro drifts. They proposed a method called zero angular rate update (ZARU), which assumes that the angular velocity of foot during stance phase is zero, and use this condition as a measurement in EKF. In many cases, this assumption is false since the angular velocity of foot is not zero. Thus, if one were to apply a ZARU, the input standard deviation would be so high that it would have no ability to observe the bias. Obviously, this method does not give any significant improvement in heading accuracy, which is seen from the results given in Jiménez et al. (2010). Bancroft and Lachapelle (2012) investigated ZARU and came to conclusions that because of the high angular velocities and low bias stabilities, ZARU is not a viable option for sensors mounted on foot, except, may be the case when the gyros are mounted in the

sole of the shoe.

For shoe-mounted sensors step detection is straightforward as the readings are consistent during the stance phase of a step and varying during the swing phase, as opposed to body-mounted sensors, in which the vertical acceleration or norm of acceleration vector exhibits a double-peaked oscillatory pattern during walking. Various methods of detecting the stance phase exists. However, the experimental results also suggest that it often suffices to use gyro information only.

The major disadvantages of foot-mounted IMU based PDR system are the following:

- Impractical location
- Exposure of IMU to high accelerations and angular velocities
- Quality of ZUPT

Note that shoe-mounted inertial sensors are not practical for soldier and firefighter applications because of impractical location. The forefoot, which is the easiest location to temporarily mount sensors, is an unrealistic location for practical military or first responder applications since it is the most exposed. The upper heel, ankle and shin are also somewhat exposed, but it is conceptually possible to mount the sensors there. However, the quality of ZUPT at these locations is not high. A next alternative would be to mount the sensors in the sole of the boot. In this case the IMU experiences additional movements and high accelerations associated with shoe deformation and bounce.

Another difficulty is that the sensor package must be connected to the GNSS receiver and navigation processor, while a shoe-mounted battery may limit the mission duration. Regardless of wired or wireless connection this is not desired. Connecting cables are too cumbersome for use on a long-term basis where the user may be required to run, climb or crawl as well as walk. Wireless connection has its own disadvantages since it increases power consumption and can be a reason for missing samples during the data transmission from sensor unit to a navigation processor.

According to Bancroft and Lachapelle (2012) the maximum angular velocity and acceleration experienced by foot-mounted sensors during running can reach 2000 deg/sec and $24g$ respectively. Typically, a higher acceleration and angular velocity range

causes more sensor noise and coarser resolution. Gyro performance can also deteriorate because of increased effect from g-dependent bias.

For reliable output ZUPTs must only be applied when the foot (and consequently the IMU) is completely static. Therefore, performance of foot-mounted INS depends significantly on the location of IMU on foot because the foot is not completely motionless when it is on the ground. Issues can arise when the IMU is attached any higher than the ball of foot. The peeling motion associated with the transition from stance to swing means that the heel rises soon after the foot-down event and hence a sensor in the mid-foot will start experiencing an acceleration as the foot levers up (Bancroft and Lachapelle (2012)). These small accelerations occur before the strict end of the stance phase and it is necessary to account for these non-zero velocities by applying a corresponding covariance for the ZUPT pseudo-measurement.

5.1.4 Body-Mounted IMU

Body-mounted IMU overcomes most disadvantages of foot-mounted IMU such as high dynamic range for gyros and accelerometers, impractical location and wires connecting a sensor unit located on foot with a navigation computer which is normally fixed to a jacket. The whole navigation system can be packaged as one unit.

In the case of body-mounted IMU, there are two methods of processing the inertial sensor measurements: PDR, which was described earlier, and standard INS with possible aiding from PDR. The latter requires a complete IMU with three accelerometers and three gyroscopes must be used. Whenever the system is stationary, ZUPT may be used to correct the INS velocity and attitude computations. With a body-mounted IMU, ZUPTs are infrequent, thus in many works the tactical-grade IMU was used to bridge GNSS outages.

Groves et al. (2007) suggested to use position solution computed by PDR as a measurement in EKF. Inertial navigation and PDR may also be used together, sharing the same inertial sensors, in which case, inertial navigation is incorporated within the multi-sensor integration architecture as the reference system and PDR as an aiding sensor. When PDR measurements are detected, a separate, PDR-only, measurement update is performed. The measurement innovation comprises the difference between the PDR-indicated step length and the INS-indicated position change between the step start and stop times.

5.2 Novel Approach for Fusion of IMU Data and Pedestrian Dynamics

In our pedestrian navigation system, the body-mounted IMU contains three gyros and three accelerometers that measure the projections of absolute angular rate and specific force on their sensitivity axes. The navigation computations are performed in the local-level coordinate frame. The transformation matrix from the sensor frame to the local-level frame is calculated using output from gyroscopes. The mechanization equations are implemented as local-level terrestrial navigator without vertical channel. Position, velocity, and attitude errors in a stand-alone INS grow with time. For short period of autonomous operation ($1-1.5$ hours) the propagation of horizontal velocity errors and tilts can be approximated by the Schuler oscillations (Farrell and Barth (1999)). When the INS is assumed to be nominally level with altitude compensation and constant low speed, the single north channel error model is described by the following equations (Salychev (2004))

$$\begin{aligned} \delta\dot{V}_E &= -g\phi_N + B_E \\ \dot{\phi}_N &= \frac{\delta V_E}{R_e} + \varepsilon_N \\ \delta\dot{V}_N &= g\phi_E + B_N \\ \dot{\phi}_E &= -\frac{\delta V_N}{R_e} + \varepsilon_E \end{aligned} \tag{5.4}$$

where $\delta V_N, \delta V_E$ are the north and the east components of velocity error, ϕ_N, ϕ_E are the horizontal tilt errors, $B_N, B_E, \varepsilon_N, \varepsilon_E$ are the projections of accelerometer biases and gyro drifts on the local-level frame, R_e is distance to the Earth center. The equations for the east channel are similar to Eq. (5.4). Note that under above assumptions the north, east and vertical channels are not coupled and can be processed separately. The block diagram representation of INS's north channel is shown in Fig. 5.3 by solid lines (Salychev (2004)).

A well known in control theory approach for oscillation damping is based on feedback loop to control the dynamic behavior of the system. A partially measured output is fed back to the controller where the difference between the reference and the output is amplified to change the input in desired way and obtain improved system performance. The damping of INS errors can be implemented using the external velocity information. The control inputs are shown in Fig. 5.3 by the dashed lines. The gains K_1 and K_2 can be pre-computed or calculated online.

In the case of pedestrian navigation, the kinetic model of gait is considered as a virtual

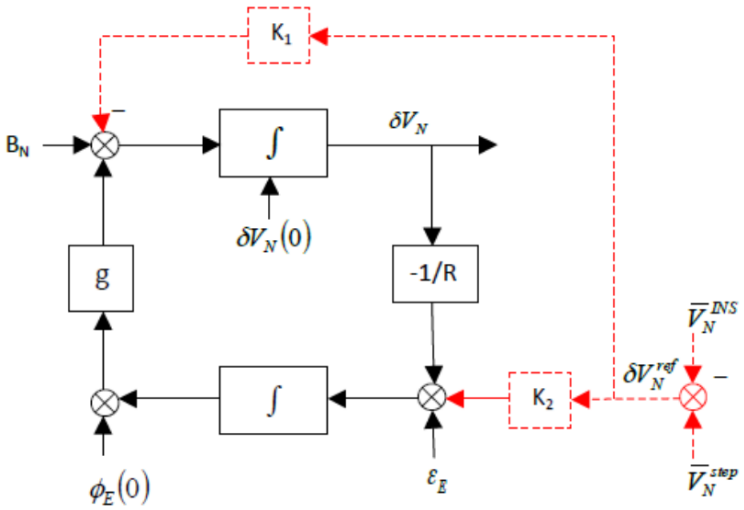

Fig. 5.3. *Diagram of the INS's north channel error damping.*

sensor that can be used to estimate the step events and step size for a person while walking. As a result, the estimations of step size and step duration can be used to calculate average velocity over each step. The error of calculated average velocity consists of the errors in step length model, inaccuracy in determining step duration, and accelerometer measurement errors. The error components are not correlated with each other thus making the error in velocity random with little correlation in time. Since the error of this estimated average velocity is also not correlated with INS velocity error it can be used as an external velocity measurement for INS.

Even when a person is walking with almost constant speed, the velocity profile within each step reveals variations of about 0.3 m/sec around the average walking speed (Fig. 5.1). Since the indicated INS velocity follows the same profile, the comparison of momentarily INS velocity with average velocity is not appropriate. Instead the indicated INS velocity averaged over one step can be compared with average velocity from an external measurement. As a result of this comparison, an estimation of the INS velocity error can be obtained.

Since the INS velocity error changes slowly it can be assumed constant during one step if the gyro drift does not exceed 50 deg/hr. In this case, averaged over one step INS indicated velocity is calculated as the true average velocity plus the velocity

error. The north component is given by

$$\overline{V}_N^{INS} = \frac{1}{(t_{k+1} - t_k)} \int_{t_k}^{t_{k+1}} \left(V_N^{true} + \delta V_N^{INS} \right) dt = \overline{V}_N^{true} + \delta V_N^{INS} \tag{5.5}$$

where δV_N^{INS} is the north component of INS velocity error, $\overline{V}_N^{INS}$ is the average north velocity calculated from the INS indicated velocity. Computations for the east component of the INS velocity error are similar to that of the north component. The measurement of the INS velocity error can be obtained by taking the difference between INS velocity, averaged over one step and the kinetic model computed velocity as follows:

$$\mathbf{z} = \begin{bmatrix} \overline{V}_N^{INS} - \overline{V}_N^{step} \\ \overline{V}_E^{INS} - \overline{V}_E^{step} \end{bmatrix} = \begin{matrix} \delta V_N^{INS} + w_N \\ \delta V_E^{INS} + w_E \end{matrix} \tag{5.6}$$

Here $\overline{V}_N^{step}$ is the average north velocity over one step calculated from the kinetic model, w_N, w_E are the velocity errors of the kinetic model based velocity estimator. While calculating the INS velocity error using Eq. (5.6) it was also assumed that the velocity estimated based on the kinetic model has a random error which is assumed to be distributed as a random Gaussian noise. This virtual measurement of INS velocity error can be performed at every step.

Let us formulate the mathematical system model for horizontal velocities and tilts in terms of error state space. Only the north channel equations are described since the data processing for the east channel is similar. Since in pedestrian navigation systems the accelerometer bias has smaller effect than the other error sources there will be three variables of primary interest for each channel: error in INS indicated velocity, tilt error, and gyro drift rate. In terms of these variables, the state differential equations for the north channels become:

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \delta V_N \\ \phi_E \\ \varepsilon_E \end{bmatrix} = \begin{bmatrix} 0 & g & 0 \\ -1/R_e & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta V_N \\ \phi_E \\ \varepsilon_E \end{bmatrix} + \begin{bmatrix} n_a \\ 0 \\ n_g \end{bmatrix} \tag{5.7}$$

where n_a, n_g, is the measurement noise in accelerometer and gyro output respectively. The measurement to be used as the input to the Kalman filter is the difference between average velocities given by Eq. (5.6). In terms of the error state notation it becomes

$$z(t_k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta V_N(t_k) \\ \phi_E(t_k) \\ \varepsilon_E(t_k) \end{bmatrix} + w(t_k) \tag{5.8}$$

Based upon system and measurement models the Kalman filter can be specified. First consider propagation from sample time t_{k-1} to time t_k. It is assumed that when the measurement is obtained the update computations are performed and the corrective signal applied to the INS. The optimal error estimate at time t_k is

$$\widehat{x}(t_k) = \begin{bmatrix} \delta\widehat{V}_N(t_k) \\ \widehat{\phi}_E(t_k) \\ \widehat{\varepsilon}_E(t_k) \end{bmatrix} \tag{5.9}$$

This estimate serves as the corrective signal to the INS, which is applied at time t_k. Thus the predicted state at time t_{k+1} before the new measurement is obtained is

$$\widehat{x}\left(t_{k+1}^-\right) = \begin{bmatrix} \delta\widehat{V}_N\left(t_{k+1}^-\right) \\ \widehat{\phi}_E\left(t_{k+1}^-\right) \\ \widehat{\varepsilon}_E\left(t_{k+1}^-\right) \end{bmatrix} = 0. \tag{5.10}$$

So there is no need to compute the predicted state explicitly. The covariance propagation equation is

$$P\left(t_k^-\right) = \Phi(t_k, t_{k-1}) P\left(t_{k-1}^+\right) \Phi^T(t_k, t_{k-1}) + Q(t_k) \tag{5.11}$$

where P is an error covariance, Φ is the state transition matrix, Q is a process noise covariance. To update the estimate at the time when new measurement is obtained the filter gain is calculated

$$K(t_k) = P\left(t_{k-1}^-\right) H^T \left(HP\left(t_{k-1}^-\right) H^T + R\right)^{-1} \tag{5.12}$$

here H is a matrix in measurement equation (Eq. (5.8)), and R is a measurement noise covariance matrix. The covariance update is

$$P\left(t_k^+\right) = P\left(t_k^-\right) - K(t_k) HP\left(t_k^-\right) \tag{5.13}$$

Since $\widehat{x}\left(t_k^-\right)$ is zero the optimal state estimate update becomes

$$\widehat{x}\left(t_k^+\right) = K(t_k) z(t_k). \tag{5.14}$$

The last formula represents the control inputs that are fed back into INS to reduce the horizontal velocity and tilt errors. Horizontal gyro drift can be also corrected in similar way.

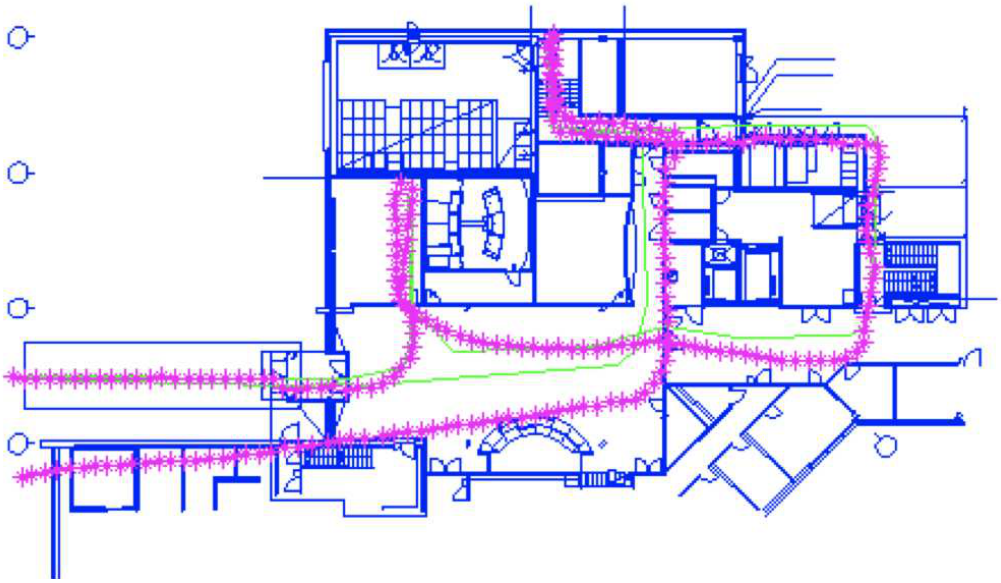

Fig. 5.4. *The walking path inside the building: The true path is shown by the green solid line. The INS computed path is shown by the magenta asterisks.*

5.3 Experimental Results

The proposed method for INS velocity aiding was tested with actual data from the indoor walking tests. The 170 meter test route inside the typical office building is shown in Fig. 5.4. The true pedestrian path is shown by green solid line. The test scenario included different types of movement: walking with variable step length and frequency, going up or down stairs, standing, making sharp turns and opening the doors. The total time of the test was about 5 minutes.

The pedestrian dead-reckoning system included 6 DOF IMU, which is composed of three Murata SCR-1100 combined gyroscope and accelerometer. This system is a self-contained device for inertial data collection. In addition to three gyros and accelerometers it includes also the batteries, memory card and all necessary electronics for data collection. The complete unit is shown Fig. 5.5 and the location of the IMU during the test is shown in Fig. 5.6.

The INS computed path is shown in Fig. 5.4. by the magenta asterisks. Our method can reduce only the distance error in PDR systems. Along-track (distance) error during this test did not exceed 2% of travelled distance. The results can be improved

Fig. 5.5. *The assembly of 6 DOF IMU, batteries and readout electronics.*

when horizontal gyros' drifts are compensated. However, the largest error is caused by heading error. Heading cannot be corrected with this algorithm and other methods, for example, map-matching can be used to reduce the heading error.

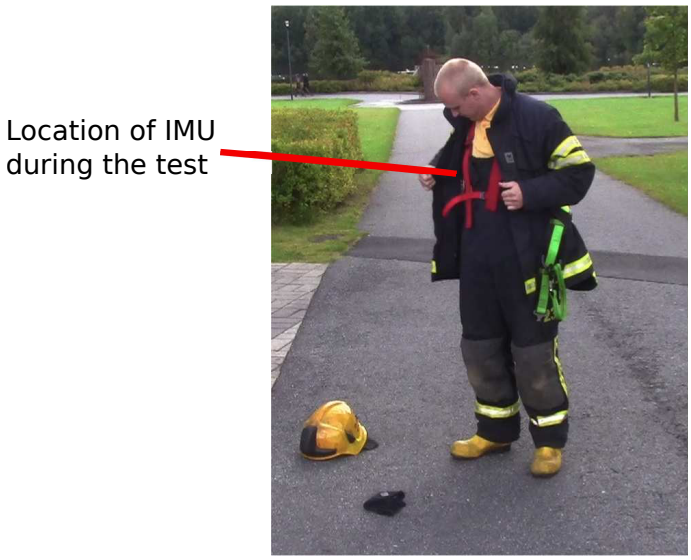

Fig. 5.6. *Location of the IMU on the body during the tests.*

5.4 Conclusions

We proposed a novel approach for INS external velocity aiding in pedestrian navigation systems. The velocity update is based on knowledge of human walking process, which provides an alternative way to calculate traveled distance and velocity averaged during the step. Our approach can be applied to pedestrian navigation systems that include IMU mounted at waist or torso.

The kinetic model of gait is considered as a virtual sensor that can be used to estimate the step events and step size for a person while walking. The term "virtual" emphasizes the fact that there is no separate instrument for direct speed measurement. The speed is estimated using accelerometer measurements and additional information about pedestrian's movement, which is derived from the model of human gait kinematics. The different characteristics of errors in INS output and in this virtual measurement make it possible to apply complementary filter methodology and significantly improve INS performance by keeping the horizontal velocity and tilt errors small.

The processing of corrected IMU output results in accurate estimation of stride length and direction. The navigation system output is based on strapdown INS and, therefore, there are no restrictions on pedestrian movement and body orientation. However, reliable velocity update is possible only during normal walking or standing. The results from pedestrian walking tests showed that the proposed navigation algorithm computes traveled distance with error of about 2% at normal walking conditions on flat terrain.

Compare to foot-mounted IMU the proposed method has the following advantages:

- It solves the problem of impractical location
- The inertial sensors are not exposed to high accelerations, angular velocities and shocks. This allows to reduce the dynamic range for the sensors and, hence, reduce the measurement noise. Besides, gyroscopes perform better in benign dynamics because of linear acceleration effect on drift.
- Motion classification is easier to perform
- ZUPT quality, which is important for foot-mounted IMU can be not good on slippery and spongy surfaces.

Compare to SHS methods with one gyro or magnetometer for heading measurements the proposed method solves the following problems:

- Limitations on person's body orientation. The SHS methods work well only when upper body is in upright position.
- Heading can be different from the direction of movement. Always apply the step motion in the forward direction determined by the body-mounted sensor. The position is propagated in the direction where the torso faces.
- Work not only for walking, but other movements like crawling and climbing on ladder.

6. CONCLUSIONS

In this thesis we proposed three algorithms that can improve performance of PNDs in GNSS denied environments. Two algorithms are concerned with map aided positioning for street and indoor navigation. The third algorithm improves performance of the pedestrian navigation system with body mounted sensors.

The map matching algorithms compare the estimated trajectory of a vehicle or pedestrian with roads or building plans stored in a map database, and the best match is chosen as the position of the vehicle. The first algorithm solves the map matching problem for a car traveling on known road network. The algorithm is based on particle filtering. Compare to the existing algorithms, which solve the map matching problem in the case when the positioning data is fairly accurate (for example GNSS under open skies, or combined GNSS/INS solution). These algorithms are usually not reliable in GNSS denied environment when sensors less accurate than GNSS have to be used. The proposed algorithm is designed for autonomous navigation based on dead reckoning sensors. The algorithm has the ability to correct both cross-track and along track position errors. Along-track errors can be corrected when the vehicle is turning on intersections. After each turn the position error is reduced to below 10 m. The accumulated position errors of the dead reckoning system can be corrected based on the results of map matching. If a vehicle path includes turns at least every 2-3 km the position error will be kept small and the navigation will continue indefinitely.

The second algorithm provides the map matching solution for autonomous navigation systems operating indoors. The major contribution of our algorithm is in method for correction of the autonomous navigation system position and heading errors. The proposed algorithm provides an accurate means of correcting the accumulation of dead reckoning position and heading errors. The idea is based on the fact that the vehicle's movement indoors is constrained by the walls; if the trajectory is long enough and includes turns, it will be quite unique and it can be identified among other

trajectory candidates. The map-matching algorithm generates position and heading measurements from knowledge of the building floor plan where the navigation system is operating. These position and heading corrections are computed by the Kalman filter and can be used to offset the initial errors as well as accumulated errors of the dead reckoning system. If the building layout and movement of the vehicle are suitable it is possible to keep small position errors and heading errors for long periods of time.

The third algorithm improves pedestrian navigation system performance in the case, when the IMU is mounted at waist or torso. The approach uses knowledge of human walking process to reduce the horizontal velocity and tilt errors. The processing of corrected IMU output results in accurate estimation of stride length and direction. The navigation system output is based on strapdown INS and, therefore, there are no restrictions on pedestrian movement and body orientation. However, reliable velocity update is possible only during normal walking or standing. The proposed algorithm overcomes the shortcomings of the existing methods such as foot-mounted INS with ZUPT and SHS. These improvements are listed in Section 5.4.

6.1 Recommendations and Future Work

The following recommendations can be made for the future research based on the results of this thesis.

The map matching algorithm for vehicle navigation discussed in Chapter 3 could be modified in a number of ways to improve performance. For example, by detecting when a vehicle goes off road network. This is important because if vehicle is not traveling on the road and map matching algorithm is applied it can result in inaccurate results. Another possible extension is adaptation of the algorithm to operate with less accurate gyroscope. In this case the algorithm will have a potential to be used in mass market consumer products.

The extension of the map matching algorithm for autonomous indoor navigation described in Chapter 4 can include seamless navigation capability from outdoors to indoors. This will enable a wide variety of navigation applications and value-added services. Also more tests in different environment have to be done to improve the robustness of indoor map matching algorithm and heading and position correction

As discussed in Chapter 5, performance of the pedestrian navigation algorithm is

currently limited by two types of movement: walking and standing. The algorithm can be also modified to operate with all kind of movements, in particular, vertical movements such as climbing on stairs or ladder. In addition to this more tests have to be made to determine how the kinetic model of gait will change when a person is waking on non-flat terrain or tilted surfaces. Finally, the possibility of application of the proposed algorithm to very low-cost inertial sensors (similar to those, which are used in smartphones) has to be also investigated.

BIBLIOGRAPHY

K. Abdulrahim, C. Hide, T. Moore, and C. Hill. Integrating low cost IMU with building heading in indoor pedestrian navigation. *J. Global Positioning Systems*, 10(1):30–38, 2011a.

K. Abdulrahim, C. Hide, T. Moore, and C. Hill. Aiding low cost inertial navigation with building heading for pedestrian navigation. *Journal of Navigation*, 64(2): 219–233, 2011b.

K. Aminian and B. Najafi. Capturing human motion using body-fixed sensors: outdoor measurement and clinical applications. *Computer Animimation and Virtual Worlds*, 15:79–94, 2004.

M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188, Mar. 2002.

C. Ascher, C. Kessler, R. Weis, and G. Trommer. Multi-floor map matching in indoor environments for mobile platforms. In *Proc. of Int. Conf. on Indoor Positioning and Indoor Navigation*, Sydney, Australia, Nov. 2012.

J. Bancroft and G. Lachapelle. Perfomance of pedestrian navigation system as a function of sensor location. In *Proc. NATO Symposium Navigation Sensors and Systems in GNSS Denied Environments*, Izmir, Turkey, Oct. 2012.

S. Beauregard. Omnidirectional pedestrian navigation for first responders. In *Proc. 4th Workshop on Positioning, Navigation and Communication*, pages 33–36, Hannover, Germany, Mar. 2007.

S. Beauregard, Widyawan, and M. Klepal. Indoor PDR performance enhancement using minimal map information and particle filters. In *Proc. of IEEE/ION Position, Location and Navigation Symposium*, Monterey, CA, May 2008.

D. Bernstein and A. Kornhauser. An introduction to map matching for personal navigation assistants. *TIDE Center, New Jersey Institute of Technology*, 1996.

J. Borenstein and L. Ojeda. Heuristic drift elimination for personnel tracking systems. *Journal of Navigation*, 63(4):591–606, 2010.

J. Borenstein, L. Ojeda, and S. Kwanmuang. Heuristic reduction of gyro drift for personnel tracking systems. *Journal of Navigation*, 62(01):41–58, 2009.

R. Brown and P. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 1996.

M. Chowdhary, J. Colley, and M. Chansarkar. Improving GPS location availability and reliability by using a suboptimal, low-cost MEMS sensor set. In *Proc. ION GNSS*, pages 610–614, Fort Worth, TX, Sep. 2007.

P. Davidson and J. Takala. Pedestrian navigation combining IMU measurements and gait models. *J. of Gyroscopy and Navigation*, 4(2):79–84, 2013.

P. Davidson, J. Hautamäki, J. Collin, and J. Takala. Using low-cost MEMS 3D accelerometer and one gyro to assist GPS based car navigation system. In *Proc. 15th International Conference on Integrated Navigation Systems*, St. Petersburg, Russia, May 2008.

P. Davidson, J. Hautamäki, J. Collin, and J. Takala. Improved vehicle positioning in urban environment through integration of GPS and low-cost inertial sensors. In *Proc. ENC-GNSS 2009*, Naples, Italy, May 2009a.

P. Davidson, M. A. Vázquez Lopez, and R. Piché. Uninterrupted portable car navigation system using GPS, map and inertial sensors data. In *Proc. 13th IEEE International Symposium on Consumer Electronics*, pages 836–840, Kyoto, Japan, May 2009b.

P. Davidson, J. Collin, J. Raquet, and J. Takala. Application of particle filters for vehicle positioning using road maps. In *Proc. 23rd ION GNSS*, pages 1653–1661, Portland, OR, Sep. 2010a.

P. Davidson, J. Collin, and J. Takala. GPS integrated with inertial sensors in mobile phones becomes a reality. *Coordinates*, VI(6):15–18, 2010b.

P. Davidson, J. Collin, and J. Takala. Application of particle filters for indoor positioning using floor plans. In *Proc. of Ubiquitous Positioning, Indoor Navigation and Location-Based Service*, Kirkkonummi, Finland, Oct. 2010c.

P. Davidson, J. Collin, and J. Takala. Map-aided autonomous pedestrian navigation system. In *Proc. 18th Int. Conf. Integrated Navigation Systems*, pages 314–318, St. Petersburg, Russia, May 2011a.

P. Davidson, J. Collin, and J. Takala. Application of particle filters to map-matching algorithm. *J. of Gyroscopy and Navigation*, 2(4):286–293, 2011b.

S. Dmitriev, A. Stepanov, B. Rivkin, and D. Koshaev. Optimal map-matching for car navigation systems. In *Proc. 6th Int. Conf. Integrated Navigation Systems*, St. Petersburg, Russia, May 1999.

R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.

N. El-Sheimy. The potential of partial IMUs for land vehicle applications. *Inside GNSS*, 3(3), 2008.

L. Fang, P. Antsaklis, L. Montestruque, M. McMickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie, and X. Xie. Design of a wireless assisted pedestrian dead reckoning system: The NavMote experience. *IEEE Trans. Instrum. Meas.*, 54(5):2342–2358, Dec. 2005.

J. Farrell and M. Barth. *The Global Positioning System and Inertial Navigation*. McGraw-Hill, 1999.

C. Fouque, P. Bonnifait, and D. Bétaille. Enhancement of global vehicle localization using navigable road maps and dead-reckoning. In *Proc. of IEEE/ION Position, Location and Navigation Symposium*, pages 1286–1291, 2008.

E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6):38–46, 2005.

R. French. Land vehicle navigation and tracking. *Global Positioning System: Theory and Applications*, 2:275–301, 1996.

R. French and G. Lang. Automatic route control system. *IEEE Transactions on Vehicular Technology*, 22(2):36–41, 1973.

M. Fu, J. Li, and M. Wang. A hybrid map matching algorithm based on fuzzy comprehensive judgment. In *Proc. of 7th IEEE Int. Conf. on Intelligent Transportation Systems*, pages 613–617, Washington, DC, 2004.

P. Gilliéron, D. Büchel, I. Spassov, and B. Merminod. Indoor navigation performance analysis. In *Proc. of the European Navigation Conference GNSS*, 2004.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. Radar Signal Process.*, 140(2): 107–113, Apr. 1993.

J. Greenfeld. Matching GPS observations to locations on a digital map. In *National Research Council (US). Transportation Research Board*, Washington, DC, 2002.

P. Groves, G. Pulford, A. Littlefield, D. Nash, and C. Mather. Inertial navigation versus pedestrian dead reckoning: Optimizing the integration. In *Proc. ION GNSS 2007*, Fort Worth, TX, Sep. 2007.

F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Tran. on Signal Processing*, 50(2), 2002.

P. Hall. A Bayesian approach to map-aided vehicle positioning. Master's thesis, Linköping University, Sweden, 2001.

R. Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys & Tutorials*, 2013.

C. Hollenstein, E. Favey, C. Schmid, and A. Somieski. Performance of low-cost real-time navigation system using a single frequency GNSS measurements combined with wheel tick data. In *Proc. ION GNSS*, pages 1610–1618, Savannah, GA, Sep. 2008.

J. Jahn, U. Batzer, J. Seitz, L. Patino-Studencka, and J. Boronat. Comparison and evaluation of acceleration based step length estimators for handheld devices. In *Proc. Int. Conf. on Indoor Positioning and Indoor Navigation*, Zürich, Switzerland, Sep. 2010.

A. Jiménez, F. Seco, J. Prieto, and J. Guevara. Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. In *Proc. 7th Workshop on Positioning, Navigation and Communication*, Dresden, Germany, Mar. 2010.

A. Jiménez, F. Seco, F. Zampella, J. Prieto, and J. Guevara. Improved heuristic drift elimination (iHDE) for pedestrian navigation in complex buildings. In *Proc. Int. Conf. Indoor Positioning and Indoor Navigation*, Guimarães, Portugal, Sep. 2011.

T. Judd. A personal dead reckoning module. In *Proc. ION GPS*, pages 47–51, 1997.

S. Kaiser, M. Khider, and P. Robertson. A human motion model based on maps for navigation systems. *EURASIP Journal on Wireless Communications and Networking*, 2011(1), Dec. 2011.

J. Käppi, J. Syrjärinne, and J. Saarinen. MEMS-IMU based pedestrian navigator for handheld devices. In *Proc. ION GPS*, pages 1369–1373, Salt Lake City, UT, Sep. 2001.

C. Keßler, C. Ascher, M. Flad, and G. Trommer. Multi-sensor indoor pedestrian navigation system with vision aiding. *Gyroscopy and Navigation*, 3(2):79–90, 2012.

M. Khider, S. Kaiser, P. Robertson, and M. Angermann. The effect of maps-enhanced novel movement models on pedestrian navigation performance. In *Proc. 12th European Navigation Conference*, Toulouse, France, Apr. 2008.

M. Khider, S. Kaiser, P. Robertson, and M. Angermann. Maps and floor plans enhanced 3D movement model for pedestrian navigation. In *Proc. ION GNSS*, Savannah, GA, Sep. 2009.

S. Kim and J. Kim. Adaptive fuzzy-network-based C-measure map-matching algorithm for car navigation system. *IEEE Trans. on Industrial Electronics*, 48(2): 432–441, 2001.

G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, 5(1):1–25, Mar. 1996.

M. Klepal and S. Beauregard. A backtracking particle filter for fusing building plans with PDR displacement estimates. In *Proc. 5th Workshop on Positioning, Navigation and Communication*, pages 207–212, Hannover, Germany, Mar. 2008.

B. Krach and P. Robertson. Integration of foot-mounted inertial sensors into a Bayesian location estimation framework. In *Proc. 5th Workshop on Positioning, Navigation and Communication*, pages 55–61, Hannover, Germany, Mar. 2008.

S. Kwakkel, S. Godha, and G. Lachapelle. Foot and ankle kinematics during gait. In *Proc. of ION NTM 2007*, pages 9–12, San Diego, CA, Jan. 2007.

Q. Ladetto. On foot navigation: Continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering. In *Proc. ION GPS*, pages 1735–1740, Salt Lake City, UT, Sep. 2000.

Q. Ladetto and B. Merminod. Digital magnetic compass and gyroscope integration for pedestrian navigation. In *Proc. 9th Int. Conf. on Integrated Navigation Systems*, St-Petersburg, Russia, May 2002.

H. Leppäkoski, J. Käppi, J. Syrjärinne, and J. Takala. Error analysis of step length estimation in pedestrian dead reckoning. In *Proc. of ION GPS*, Portland, OR, Sep. 2002.

T. Lezniak, R. Lewis, and R. McMillen. A dead reckoning/map correlation system for automatic vehicle tracking. *IEEE Transactions on Vehicular Technology*, 26 (1):47–60, 1977.

J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *J. Am. Statist. Assoc.*, 93(443):1032–1044, Sep. 1998.

C. Matthews, Y. Ketema, D. Gebre-Egziabher, and M. Schwartz. In-situ step size estimation using a kinetic model of human gait. In *Proc. ION GNSS*, Portland, OR, Sep. 2010.

SCR1100-D04 Single Axis Gyroscope with Digital SPI Interface. Murata Electronics Oy, Vantaa, Finland, 2.0 edition. Data sheet, doc.nr. 82 1226 00.

Y. Oshman and P. Davidson. Optimization of observer trajectories for bearing-only target localization. *IEEE Trans. on Aerospace and Electronic Systems*, 35(3):892–902, July 1999.

J. Parviainen, M. Kirkko-Jaakkola, P. Davidson, M. A. Vázquez Lopez, and J. Collin. Doppler radar and MEMS gyro augmented DGPS for large vehicle navigation. In *Proc. ICL-GNSS Conference*, Tampere, Finland, June 2011.

O. Pekkalin, H. Leppäkoski, J. Hautamäki, J. Collin, and J. Takala. Reference for indoor location systems using gyroscope and quadrature incremental encoder. In *Proc. 23rd ION GNSS*, pages 1192–1197, Portland, OR, Sep. 2010.

T. Perälä and S. Ali-Löytty. Kalman-type positioning filters with floor plan information. In *Proc. 6th Int. Conf. Advances in Mobile Computing & Multimedia*, pages 350–355, Nov. 2008.

J. Pinchin, C. Hide, and T. Moore. A particle filter approach to indoor navigation using a foot mounted inertial navigation system and heuristic heading information. In *Proc. of Int. Conf. on Indoor Positioning and Indoor Navigation*, pages 1–10, 2012.

M. Quddus. *High integrity map matching algorithms for advanced transport telematics applications*. PhD thesis, Imperial College London, United Kingdom, 2006.

M. Quddus, W. Ochieng, L. Zhao, and R. Noland. A general map matching algorithm for transport telematics applications. *GPS Solutions*, 7(2):157–167, 2003.

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

J. Saarinen. *A Sensor-Based Personal Navigation System and Its Application for Incorporating Humans into A Human-Robot Team*. PhD thesis, Helsinki University of Technology, Finland, 2009.

O. Salychev. *Applied Inertial Navigation: Problems and Solutions*. BMSTU Press, Moscow, Russia, 2004.

C. Scott. Improved GPS positioning for motor vehicles through map matching. In *Proc. 7th ION-GPS*, pages 1391–1400, Salt Lake City, UT, Sep. 1994.

I. Skog and P. Händel. In-car positioning and navigation technologies: A survey. *IEEE Trans. on Intelligent Transportation Systems*, 10(1), 2009.

W. Soehren and W. Hawkinson. Prototype personal navigation system. *IEEE Aerospace and Electronic Systems Magazine*, 23(4):10–18, 2008.

A. Somieski, C. Hollenstein, E. Favey, and S. C. Low-cost sensor fusion dead reckoning using a single frequency GNSS receiver combined with gyroscope and wheel tick measurements. In *Proc. ION GNSS*, pages 1645–1652, Portland, OR, Sep. 2010.

I. Spassov. *Algorithms for map-aided autonomous indoor pedestrian positioning and navigation*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, 2007.

D. Srinivasan, R. Cheu, and C. Tan. Development of an improved ERP system using GPS and AI techniques. In *Proc. of IEEE Intelligent Transportation Systems*, volume 1, pages 554–559, 2003.

R. Stirling, J. Collin, K. Fyfe, and G. Lachapelle. An innovative shoe-mounted pedestrian navigation system. In *Proc. of ENC-GNSS*, pages 110–115, Graz, Austria, Apr. 2003.

S. Syed and M. Cannon. Fuzzy logic based map matching algorithms for vehicle navigation system in urban canyons. In *Proc. ION National Technical Meeting*, pages 26–28, San Diego, CA, Jan 2004.

K. Taki, Y. Itoh, S. Kato, and H. Itoh. Motion generation for bipedal robots using a neuro-musculo-skeletal model and simulated annealing. *IEEE Robotics, Automation and Mechatronics*, 2:699–703, 2004.

N. Wang, E. Ambikairajah, S. Redmond, B. Celler, and N. Lovell. Classification of walking patterns on inclined surfaces from accelerometry data. In *Proc. 16th Int. Conf. on Digital Signal Processing*, pages 1–4, July 2009.

H. Weinberg. Using the ADXL202 in pedometer and personal navigation applications. Application note AN-602, Analog Devices Inc., Norwood, MA, 2002.

C. White, D. Bernstein, and A. Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C: Emerging Technologies*, 8(1):91–108, 2000.

O. Woodman. *Pedestrian localisation for indoor environment*. PhD thesis, University of Cambridge, UK, 2010.

O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *Proc. of UbiComp08*, Seoul, Korea, Sep. 2008.

Y. Zhao. *Vehicle Location and Navigation Systems*. Artech House Publishers, 1997.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3174-3
ISSN 1459-2045