A Thesis

entitled

Development of Statistical Learning Techniques for INS and GPS Data Fusion

by

Srujana Adusumilli

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the

Master of Science Degree in Electrical Engineering

________________________________________
Dr. Hong Wang, Committee Chair

________________________________________
Dr. Vijay Devabhaktuni, Committee Co-Chair

________________________________________
Dr. Mansoor Alam, Committee Member

________________________________________
Dr. Weiqing Sun, Committee Member

________________________________________
Dr. Nikolaidis Efstratios, Committee Member

________________________________________
Patricia R. Komuniecki, PhD, Dean
College of Graduate Studies

The University of Toledo

August 2014
An Abstract of
Development of Statistical Learning Techniques for INS and GPS Data Fusion
by
Srujana Adusumilli
Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Electrical Engineering
The University of Toledo
August 2014

Global Positioning System (GPS) and Inertial Navigation System (INS) are two salient technologies delivering vehicles position, velocity, and attitude parameters for land vehicle navigation. GPS provides absolute and accurate navigation parameters over extended periods of time. However, standalone GPS performance deteriorates in certain scenarios such as, when a vehicle passes through urban areas or through forests leading to satellite signal blockages and multipath effects. Whereas, INS is a self-contained navigation technology, capable of providing navigation solution by continuously measuring linear accelerations and angular velocities in three orthogonal directions. However, depending upon INS grade, their standalone accuracy varies, due to several reasons like sensor errors, scale-factor errors, noises, and drifts. Low-cost INS consisting of MEMS sensors are being used practically due to several advantages. For instance, they are cost-effective, small in size, and light in weight. Thus, to overcome the limitations of standalone GPS and INS, an integrated INS/GPS system is required for continuous, accurate, and reliable navigation solution. In an integrated system, GPS aids INS in its error modeling process thereby improving its long-term accuracy. On the other hand, INS bridges GPS gaps and assists in signal acquisition and reacquisition thus reducing the
time and search domain required for detecting and correcting GPS cycle slips. Thus for an improved, reliable, and continuous navigation, their synergistic combination is preferred while simultaneously overcoming the individual unit drawbacks.

This thesis aims at developing novel statistical learning algorithms, namely Random Forest Regression, hybrid of Principal Component Regression and Random Forest Regression, and Quantile Regression Forests, for INS and GPS data fusion. The performance of the proposed techniques is evaluated using real field test data. The test results demonstrated the improved positioning accuracy and reduced positional drift in comparison to existing techniques during GPS outages. Through experimental demonstration, the Quantile Regression Forests has shown improved performance by providing a maximum of 87% improvement in prediction accuracy in comparison to conventional Artificial Neural Networks.
To my parents and friends for their love and support
Acknowledgements

This journey would not have been possible without the support of my family, professors and supervisors, and friends. First and foremost I express my deepest gratitude to my supervisors Dr. Hong Wang and Dr. Vijay Devabhaktuni for their continued support, guidance, and encouragement throughout my graduate studies. Their meticulous comments were an enormous help in materializing this thesis. I would like to offer my special thanks by sharing credit of my work with Deepak Bhatt for his constructive comments and generous support during the research. Financial support from the EECS department and AFOSR is greatly acknowledged as well.

It gives me great pleasure in acknowledging my lab mates for shaping a creative and exhilarating work environment. I cannot find words to express my gratitude to my friends Anuroopa, Kavya, Snigdha, Jayaram, Prem, Vishu, Harish, Pavan, and Srimanth for their continuous support and unforgettable memories. Above all, I would like to thank my parents, Andhra Kumari and Jagannadha Rao for their unconditional support and encouragement through all walks of life.
# Table of Contents

Abstract ........................................................................................................................................ iii

Acknowledgements .................................................................................................................... v

Table of Contents ...................................................................................................................... vi

List of Tables ............................................................................................................................... ix

List of Figures ............................................................................................................................. x

List of Abbreviations .................................................................................................................. xii

1 Introduction ............................................................................................................................. 1

1.1 Background ......................................................................................................................... 2

1.2 Research Objective and Contributions ............................................................................. 4

1.3 Thesis Outline ...................................................................................................................... 5

2 Literature Review ................................................................................................................... 6

2.1 Global Positioning System (GPS) .................................................................................. 6

2.1.1 GPS Principle of Operation .................................................................................... 8

2.1.2 GPS Signal Errors ................................................................................................ 8

2.2 Inertial Navigation System (INS) ................................................................................ 13

2.2.1 INS Mechanization Process .................................................................................. 14

2.2.2 Inertial Measurement Unit Errors ......................................................................... 17

2.3 INS/GPS Integration ........................................................................................................ 22

2.3.1 Overview of ANN based PUA Model .................................................................... 24
A.3  R Software Commands for executing ‘quantregForest’ package ................. 79

Appendix B ................................................................................................................... 80
B.1  Snapshot of GPS Visualizer web page ............................................................... 80
B.2  Snapshot of Neuromodeler software ............................................................... 81
Appendix C: Reference Frames ............................................................................... 82
List of Tables

1.1 Contributions of research work .................................................................4
2.1 Typical inertial sensor errors magnitude .....................................................21
2.2 Salient Features of GPS and INS ...............................................................22
2.3 Advantages of INS/GPS Integrated System ...............................................23
3.1 Characteristics of Crossbow IMU and HG 1700 ........................................33
3.2 Position errors for the proposed RFR model and the conventional PUA model...39
4.1 Position errors for the proposed PCR-RFR model and the RFR model ..........53
5.1 Position errors for the proposed QRF model and the RFR model ...............62
5.2 Position errors for the proposed QRF model and the ANN model ..............63
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Global Positioning System</td>
<td>7</td>
</tr>
<tr>
<td>2-2</td>
<td>Errors on GPS Signal</td>
<td>10</td>
</tr>
<tr>
<td>2-3</td>
<td>INS Mechanization Process</td>
<td>16</td>
</tr>
<tr>
<td>2-4</td>
<td>IMU Errors</td>
<td>18</td>
</tr>
<tr>
<td>2-5</td>
<td>Integrated INS/GPS system during GPS availability</td>
<td>23</td>
</tr>
<tr>
<td>2-6</td>
<td>Integrated INS/GPS system during GPS outages</td>
<td>24</td>
</tr>
<tr>
<td>2-7</td>
<td>The PUA MLP network</td>
<td>25</td>
</tr>
<tr>
<td>3-1</td>
<td>Random Forests work flow</td>
<td>31</td>
</tr>
<tr>
<td>3-2</td>
<td>Field test trajectory showing simulated GPS outages (in Blue)</td>
<td>34</td>
</tr>
<tr>
<td>3-3</td>
<td>Performance during GPS outage 1</td>
<td>36</td>
</tr>
<tr>
<td>3-4</td>
<td>Performance during GPS outage 2</td>
<td>37</td>
</tr>
<tr>
<td>3-5</td>
<td>Performance during GPS outage 3</td>
<td>38</td>
</tr>
<tr>
<td>3-6</td>
<td>Performance during GPS outage 4</td>
<td>38</td>
</tr>
<tr>
<td>4-1</td>
<td>Hybrid approach based training model</td>
<td>44</td>
</tr>
<tr>
<td>4-2</td>
<td>Hybrid approach based prediction model</td>
<td>44</td>
</tr>
<tr>
<td>4-3</td>
<td>Field test trajectory showing simulated GPS outages</td>
<td>47</td>
</tr>
<tr>
<td>4-4</td>
<td>Performance during GPS outage 1</td>
<td>48</td>
</tr>
<tr>
<td>4-5</td>
<td>Performance during GPS outage 2</td>
<td>49</td>
</tr>
<tr>
<td>4-6</td>
<td>Performance during GPS outage 3</td>
<td>50</td>
</tr>
</tbody>
</table>
List of Abbreviations

AI ......................... Artificial Intelligence
ANFIS ...................... Adaptive Neuro-Fuzzy Interference System
ANN ......................... Artificial Neural Network
AR ......................... Auto Regressive

CS .......................... Control Segment

DGPS ........................ differential Global Positioning System
DoD ........................ Department of Defense

EKF ........................ Extended Kalman Filter

GM .......................... Gauss-Markov

GPS .......................... Global Positioning System

IMU ........................ Inertial Measurement Unit
INS ............................ Inertial Navigation System

KBNN ........................ Knowledge Based Neural Networks
KF ......................... Kalman Filter
LOS ...................... Line of Site

MEDLL .................. Multipath Eliminating Delay Lock Loop
MEMS ..................... Micro electro mechanical Systems
MLP ...................... Multi Layer Perceptron

NN ....................... Neural Network

PAC ..................... Pulse Aperture Correlator
PC ......................... Principal Components
PCR ........................ Principal Component Regression
PF .......................... Particle Filter
PUA ........................ Position Update Architecture

QRF ..................... Quantile Regression Forests

RBF ........................ Radial Basis Function
RF .......................... Random Forest
RFR ........................ Random Forest Regression
RMSE ..................... Root Mean Square Error

SS .......................... Space Segment
UKF .................. Unscented Kalman Filter

US .................. User Segment
Chapter 1

Introduction

Land vehicle navigation requires information about a vehicle’s position and velocity along all the three dimensions for knowing the current location or the required destination. Global Positioning System (GPS) is the most widely used technology delivering accurate positioning information over extended periods of time, covering any part of the world during day or night [1]. GPS provides navigation parameters only when a GPS receiver has Line-Of-Site (LOS) to four or more GPS satellites [2]. The GPS fails when a vehicle passes through urban areas or through forests leading to multipath effects or blockage of satellite signals. These kinds of signal outages lead to decrease in prediction capability and hence a poor navigation solution. In such scenarios of GPS outages, we employ an alternative navigation technology called Inertial Navigation System (INS) to maintain navigation solution continuity [3]. INS is a self-contained system which provides navigation solution using an in built Inertial Measurement Unit (IMU) and an on board computer. The IMU outputs the vehicle’s linear acceleration and angular velocity using three sets of orthogonally placed accelerometers and gyroscopes. The onboard computer processes these raw IMU measurements and provides the required velocity and position coordinates. However, sensor errors like biases, scale-factor errors,
noises, and drifts affect the IMU measurements leading to degradation of solution accuracy with time, depending on the grade of INS. In order to overcome the problems of unavailability and high cost, Micro Electro Mechanical Systems (MEMS) based inertial sensors have been utilized for land vehicle navigation. In spite of advantages like low cost, small size, and low power consumption, the measurements from MEMS based sensors contain high amount of error. Thus, we need an integrated low-cost INS/GPS system in which the INS errors are estimated during GPS signal availability and during GPS outages the INS errors can be estimated and compensated thereby obtaining a reliable and accurate navigation solution [4, 5].

1.1 Background

The integration technique, as discussed, initially models the INS error and obtains accurate navigation parameters during GPS signal availability. During outages, the model is utilized to estimate and compensate the INS error resulting in improved navigation accuracy. Many fusion techniques are employed to fuse INS and GPS data, the traditionally employed fusion techniques are Bayesian filtering approaches like Kalman Filter (KF), Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF). The KF cannot be employed for non-linear systems as it has inadequacies relating to Gaussian white noise, ideal dynamics model, and non-linear error linearization thus causing unsatisfactory performance in case of low-cost INS [6]. To make the KF work for non-linear systems Extended Kalman Filter (EKF) is developed which does linearization of non-linear systems [7, 8]. The linearization is done using Taylor’s series expansion which involves derivation of Jacobian matrices. Unfortunately, the
linearization process and filter tuning stage is time consuming, complicated, and leads to filter divergence. To overcome the limitations of EKF in evaluation of Jacobian matrices, Unscented Kalman Filter (UKF) was developed [9, 10]. UKF approximates Gaussian distribution instead of an arbitrary nonlinear function. UKF samples minimal points called sigma points which estimate the true mean and covariance of the Gaussian distribution. The posterior mean and covariance are obtained by propagating these sigma points through the nonlinear system. However, for highly nonlinear data, the UKF does a very poor approximation of Gaussian distribution.

To overcome the limitations of KF and EKF, Particle Filter is implemented. In PF, a cluster of random particles are used to represent the posterior distribution instead of a linearized model as in EKF, but this kind of representation requires a large number of particles making the algorithm computationally expensive and not applicable for real-time applications [11, 12, 13]. To overcome the limitations of Bayesian Filtering techniques, Artificial Intelligence (AI) approaches such as Artificial Neural Networks (ANN) are employed due to their ability to handle non-linear input-output relationships effectively [14, 15]. The most widely used ANNs are Multi-Layer Perceptron (MLP), Radial Basis Function (RBF), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Knowledge Based Neural Networks (KBNN) [16-24]. These Neural Networks learn the inherent input-output relationship of GPS and INS data during training and the trained model is further utilized to estimate the navigation parameters during GPS outages. However, input-output relationship to be modeled is highly complex in the case of low-cost INS. This leads to decrease in generalization capability of ANN.
1.2 Research Objective and Contributions

As described in the previous section, an integrated INS/GPS system provides accurate navigation parameters. The research objective is thus oriented towards developing algorithms to integrate GPS and INS data and estimate navigation errors during GPS signal outages. As a first step, Random Forest Regression (RFR) is introduced for low-cost INS and GPS data fusion. Next, to further improve the prediction accuracy of RFR, a hybrid of Principal Component Regression (PCR) and Random Forest Regression (RFR) is implemented. Finally, to model highly non-linear data and improve the stand-alone INS navigation accuracy, the research introduces Quantile Regression Forests (QRF). The major contributions of this research are listed in Table 1.1.

**TABLE 1.1: Contributions of research work**

<table>
<thead>
<tr>
<th>Publication</th>
<th>INS and GPS fusion approaches</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A low-cost INS/GPS integration methodology based on random forest regression</td>
</tr>
<tr>
<td>2</td>
<td>A novel hybrid approach utilizing principal component regression and random forest regression to bridge the period of GPS outages</td>
</tr>
<tr>
<td>3</td>
<td>Improving low-cost INS/GPS positioning accuracy using Quantile regression forests</td>
</tr>
</tbody>
</table>
1.3 Thesis Outline

Chapter 2 presents the literature review of INS and GPS systems, the importance of INS/GPS integrated system, and an overview of existing Artificial Neural Networks for INS/GPS integration.

Chapter 3 describes the Random Forest Regression (RFR) method for INS/GPS integration. The chapter provides details of RFR implementation and is compared with existing Artificial Neural Networks.

Chapter 4 discusses the implementation of Hybrid of Principal Component Regression (PCR) and Random Forest Regression (RFR) for fusing low-cost INS and GPS data. The results depict the advantage of proposed methodology over RFR.

Chapter 5 presents the Quantile Regression Forests (QRF) for low-cost INS/GPS data fusion. The test results with real field test data are compared with ANN and RFR.

Chapter 6 concludes the research and provides meaningful recommendations for future work.
Chapter 2

Literature Review

2.1 Global Positioning System (GPS)

Land vehicle navigation requires information about a vehicle’s position and velocity along all the three dimensions for knowing the current location or the required destination. Global Positioning System (GPS) is the most widely used technology delivering accurate positioning information over extended periods of time, covering any part of the world during day or night. GPS is a space-based satellite navigation system developed and maintained by the U.S. Department of Defense (DoD). It was originally run with 24 satellites at 20, 200 Km above the earth. These satellites are distributed over six orbits around the earth and rotate with approximately 12 hour periods. For GPS to provide navigation parameters, it should have Line-Of-Site (LOS) with a minimum of four satellites as shown in Figure 2-1.
FIGURE 2-1: Global Positioning System

A GPS consists of three major segments, namely, Space Segment (SS), Control Segment (CS) and User Segment (US). The space segment consists of 24 satellites transmitting signals to users. GPS control segment is composed of a master control station, an alternate master control station, a host of dedicated and shared ground antennas, and monitor stations to track the GPS satellites, monitor their transmissions, perform analyses, and send commands and data to the satellites. The user segment is composed of thousands of users of the secure GPS Precise Positioning Service and millions of civil, commercial, and scientific users of the Standard Positioning Service.
2.1.1 GPS Principle of Operation

The basic principle of operation is that each GPS satellite sends out a navigation signal, along with a set of orbital parameters called ephemeris data. A GPS receiver captures this data, and can use the ephemeris to calculate the position of the satellite at any point in time during a four-hour window. (The ephemeris data each satellite transmits is updated regularly by the ground stations that monitor the satellites.) The navigation signal, time-stamped with the satellite’s time of transmission, is used to calculate the receiver’s range to the satellite. This is done by multiplying the time difference between the signal reception and the signal transmission by the speed of light. This measurement of range is called pseudorange because it is not a true range: the receiver’s clock is not perfectly synchronized with the satellite’s clock, and this causes an error in the receiver’s time of signal reception. With a minimum of four GPS pseudorange measurements, a receiver can calculate its position. Four satellites are needed for a position solution because four variables are solved: the position of the receiver in three dimensions and the clock error of the receiver (which is needed to remove the large error it induces in the pseudoranges) [25].

2.1.2 GPS Signal Errors

As discussed in the previous section, the required position coordinates are obtained by calculating pseudorange to a minimum of four satellites. Pseudorange is measured by correlating the incoming signal from satellites with an identical signal re-produced from the GPS receiver. Practically, pseudorange can be described as the true range between a satellite and receiver plus the effects of various error sources. The various errors are due
to effects like atmosphere delays, multipath effects, receiver noise, etc. Thus, the pseudorange is expressed as:

\[
\rho = R + c \ast (\delta t_r - \delta t_s) + D_s + I + T + M + \varepsilon_n
\]  

(2.1)

Where, \( \rho \) = GPS pseudorange (meters)

- \( R \) = True range from satellite to receiver (meters)
- \( c \) = Velocity of light (meters/second)
- \( \delta t_r \) = Receiver clock error (seconds)
- \( \delta t_s \) = Satellite clock error (seconds)
- \( D_s \) = Orbital error (meters)
- \( I \) = Ionospheric delay (meters)
- \( T \) = Tropospheric delay (meters)
- \( M \) = Multipath effect delay (meters)
- \( \varepsilon_n \) = Receiver noise (meters)

The different error sources (Figure 2-2) that introduce delay in the true pseudorange measurement can be broadly divided into three categories:

1. Receiver level error: Multipath effect and receiver noise
2. Space segment error: orbital error and clock error
3. Atmospheric Effect: Ionospheric and tropospheric delays.
(i) **Multipath**

The basic concept of GPS relies on the idea that a GPS signal travels straight from a satellite to the receiver. Practically, the GPS signal is reflected off objects such as tall buildings or large rock surfaces before it reaches the receiver. This results in increase of the travel time of the signal, thereby causing erroneous measurements. The result is a messy signal which is a barrage of signals arriving at the receiver: first the direct one and then a bunch of delayed reflected ones. If the reflected signals are strong enough they can cause erroneous measurements by confusing the receiver. Sophisticated receivers use a variety of signal processing tricks such as Narrow correlator, Multipath Eliminating Technique, Multipath Eliminating Delay Lock Loop (MEDLL), Pulse Aperture...
Correlator (PAC), and Novatel’s Vision Correlator etc. to make sure that they only consider the earliest arriving signals (which are the direct ones) [26-31].

(ii) **Receiver noise**

Receiver noise is generated due to the noise picked up by the receiver loop along with receiver electronics or high frequency thermal noises [32]. The jitter for an early late one chip spacing correlator is given by

\[
\sigma_{dl} = \frac{\alpha B_c d}{\sqrt{c/n_0}} \lambda_c 
\]

(2.2)

Where, \( \alpha \) = DLL discriminator correlator factor

\( B_c \) = Code loop bandwidth (Hertz)

\( d \) = Correlator chip spacing

\( c/n_0 \) = Carrier to noise density ratio

\( \lambda_c \) = Code wavelength

(iii) **Ionosphere delays**

The ionosphere is the upper layer of the atmosphere ranging in altitude from 50 to 500 km. It consists largely of ionized particles which cause perturbing effect on GPS signals. The GPS signals while propagating through the ionosphere collide with ionized particles thus causing disturbance. The delay in GPS signal is dependent on the density of the ionized particles and the elevation of the signal. The density is affected by the sun; hence, there is less ionosphere effect during nights. In addition, low elevation satellite signals
have to travel longer distances, thus, they have longer ionosphere delays. This delay is eliminated by using sophisticated receivers, which contain an elevation mask and the signals below the mask are not used for position computation. Most of the errors induced by the ionosphere are removed using mathematical modeling [33-35].

(iv) Troposphere delays

The Troposphere is closer to the Earth surface and encompasses the weather. The troposphere is full of water vapor and delay is due to variations in temperature, pressure, and humidity. The troposphere delay can be modeled with a function that takes temperature and water vapor as arguments [41].

(v) Orbital error

Orbital errors are also called ephemeris errors and are caused due to inaccuracies of the satellite’s reported location. The satellite coordinates are obtained using 16 broadcasted ephemerides, which involves curve fitting of the ephemeris parameters to predict the satellite coordinates. There errors are minimized by techniques like modeling.

(vi) Clock error

The satellite Oscillator instabilities produce the clock error. The deviation of satellite clock corresponds to satellite clock offsets and is monitored by master control stations. A prediction model is considered which generates clock bias, drift, and drift rate which are used to model the clock offset. These generated parameters are used by the receiver in a user segment to estimate the clock error $E_c$, given by Eq. 2.3.
\[ E_c = e_0 + e_i(t - t_0) + e_r(t - t_0)^2 \]  \hspace{1cm} (2.3)

Where,

- \( e_0 \) = clock bias term
- \( e_i \) = clock drift term
- \( e_r \) = clock drift rate

### 2.2 Inertial Navigation System

Inertial Navigation System (INS) is a self-contained navigation technique in which the position and orientation of an object are tracked relative to a known starting point, orientation and velocity. INS consists of two main units namely, the Inertial Measurement Unit (IMU) and an On-Board Computer. IMUs typically contain three sets of gyroscopes and accelerometers placed orthogonal to each other, measuring angular velocity and linear acceleration respectively. The signals from these devices are processed to obtain the required navigation parameters, i.e. position and orientation. The INS needs to be provided with initial position and velocity from another source which is mostly a GPS satellite receiver. Later it computes its own updated position and velocity components utilizing the information received from the IMU. As said, the gyroscopes measure angular velocity of the system in the inertial reference frame. The orientation of the inertial reference frame is taken as the initial orientation and the current orientation is obtained by integrating the angular velocity. Similarly, the accelerometer measures the linear acceleration of the system in the inertial reference frame. Considering the original velocity and position as initial velocity and position, the current velocity and position can be obtained by integrating the linear acceleration. Integrating the acceleration yields the
velocity of the system and integrating it again yields the position. The process of integration is done using a set of kinematic equations and is performed using the on-board computer. The process of integrating the linear and angular accelerations to obtain the required navigation parameters is called the mechanization.

2.2.1 INS Mechanization Process

The calculation of the navigation solution necessitates reasonably accurate initial parameters as it is an iterative process and it depends on previous epoch solutions. The initial parameters are usually initialized by an external source as GPS. The output from gyroscopes and accelerometers are obtained in body frame ($b$-frame). However, navigation is often conducted in a local level frame ($l$-frame) and therefore, the measured signals have to be converted into $l$-frame from $b$-frame. The $b$-frame parameters are transformed to navigation frame parameters by attitude initialization and the process is called alignment process. Several researches have explored the feasibility of different alignment methods [37-40]. The mechanization equations are sets of equations that transform the output of the IMU into useful position, velocity, and attitude parameters. Typically, an IMU consists of three single-axis accelerometers and gyroscopes for each X, Y, and Z direction and a processor. The accelerometers and gyroscopes measure linear and angular velocities respectively, in the $b$-frame.

The measurements from an IMU are generally in $b$-frame coordinates and have to be transformed into navigation frame components, while compensating for all the sensors errors, gravity, change in orientation of the $l$-frame with respect to the $e$-frame, and the Earth’s rotation. This process of determining the navigation parameters from raw IMU
measurements is performed by the mechanization equations [37]. The mechanization equations in the $l$-frame (North-East-Down) are shown in Eq. 2.4.

$$
\begin{pmatrix}
\dot{r}^l \\
\dot{v}^l \\
\dot{R}^l_b
\end{pmatrix} = \begin{pmatrix}
D^{-1}v^l \\
R^l_b f^b - (2\Omega^l_{ie} + \Omega^l_{el})v^l + g^l \\
R^l_b (\Omega^l_{e_b} - \Omega^l_{g})
\end{pmatrix}
$$

(2.4)

Where,

$r^l$ = Position vector containing latitude ($\phi$), longitude ($\lambda$), and height ($h$) in $l$-frame,

$v^l$ = Velocity vector in $l$-frame ($v_{East}, v_{North}, v_{Down}$),

$R^l_b$ = b-frame to l-frame rotation matrix,

$g^l$ = Earth’s gravity in l-frame,

$\Omega^l_{e_b}$ = Skew-symmetric matrix of angular velocity $w^b_{e_b}$ measured by gyroscopes,

$\Omega^l_{e_l}$ = Skew-symmetric matrix of angular velocity $w^b_{e_l}$,

$\Omega^l_{i_e}$ = Skew-symmetric matrix of angular velocity $w^b_{i_e}$,

$\Omega^l_{i_l}$ = Skew-symmetric matrix of angular velocity $w^b_{i_l}$,

$D^{-1} = 3 \times 3$ matrix whose non zero elements are functions of the user’s latitude and height

$$
D^{-1} = \begin{pmatrix}
1 & 0 & 0 \\
\frac{1}{M + h} & 0 & 0 \\
0 & \frac{1}{(N + h) \cos \phi} & 0 \\
0 & 0 & -1
\end{pmatrix}
$$

(2.5)
Where, M and N are meridian and prime vertical radius of curvatures.

The detailed numerical solution of above mechanization equations are discussed in several references [39]. The mechanization solution is schematically represented in Figure 2-3.

**FIGURE 2-3: INS Mechanization Process**

The mechanization equations consist of three basic steps as given below [41].

(i) Correction of raw inertial sensor measurement error,

(ii) Attitude update after accounting for errors,

(iii) Velocity and position update.
(i) **Correction of raw inertial sensor measurement error**

In case of low-cost IMU’s, i.e. IMU’s with MEMS sensors, the accelerometer and gyroscope measurements (specific force and angular velocity) are outputted without integration. Thus, these raw measurements need to be compensated for IMU errors like bias and scale factor. This correction results in proper measurements of specific force and angular rates.

(ii) **Attitude Update**

The gyroscope measures vehicles angular rates along with the effect of Earth’s rotation ($w_{ew}$) and change of orientation of the $l$-frame with respect to the Earth ($w_{el}$). To obtain the actual angular rate of the moving body along with the updated rotation matrix, the gyro measurements need to be compensated for these components.

(iii) **Velocity and Position Update**

The updated rotation matrix is utilized to rotate the accelerometer output (specific force) from the $b$-frame to the $l$-frame as shown in Figure 2-3. The output of accelerometer corresponds the sum of true vehicle acceleration and effect of Earth’s gravity and coriolis acceleration. Coriolis acceleration is due to coriolis force, i.e. produced due to deflection in object flight, due to rotation of Earth. After compensating for these effects, this acceleration is integrated to obtain a vehicle’s velocity, which again needs to be integrated to obtain the required position.

### 2.2.2 Inertial Measurement Unit Errors

As discussed in the previous section, the advantage of INS is that it requires no external reference after initialization and thus is immune to jamming and deception.
Therefore, INS is used in wide range of applications including but not limited to navigation of spacecraft, submarines, and guided missiles. However with the recent advancement of MEMS technology, low-cost, small sized, and light weight inertial sensors are available and have opened its applicability to low-cost commercial applications [42]. MEMS have enabled the sensor technology to evolve from restricted, expensive, and inflexible units to miniaturized, low-cost, and low-power silicon-based units. In spite of advantages like, being small in size and light in weight, MEMS sensors experience large errors like turn-on to turn-on biases, in-run biases, scale factor drifts, and other environment dependent errors [43]. The inertial sensor errors can be divided into two main categories: Deterministic Errors and Random Errors as shown in Figure 2-4.

FIGURE 2-4: IMU Errors
(i) **Deterministic errors**

Deterministic/Systematic errors should be calibrated and removed from the sensor data before further processing on the data takes place [39, 40, 44]. The prominent deterministic errors are listed as follows.

(a) **Bias**

Ideally, the output signal received from the sensor should read zero when no input is applied to the sensor. However, this is not the case and an offset, called the bias, exists. In other words, bias is considered as the average of accelerometer or gyroscope output over a time period that has no relation with input linear or angular acceleration. The uncompensated bias, in case of accelerometer, introduces an error proportional to time $t$ in the velocity (Eq. 2.6) and an error proportional to $t^2$ in position (Eq. 2.7).

$$v = \int b \, dt = b \, t$$

(Eq. 2.6)

$$p = \int v \, dt = \int b \, t \, dt = \frac{1}{2} b \, t^2$$

(Eq. 2.7)

Similarly, an uncompensated gyroscope bias will introduce an angle error proportional to the time $t$, given by Eq. 2.8.

$$\delta \theta = \int \delta \theta \, dt = \delta \theta \, t$$

(Eq. 2.8)

(b) **Scale factor**

The ratio of change in the output to a change in the intended input to be measured is called the scale factor [45]. Scale factor is evaluated as the slope of a straight trend line that can be fitted to the input-output data. Ideally, the sensors scale factor is 1 and thus
any deviation of scale factor above or below 1, is termed as scale factor error. The accelerometer scale factor error causes position error proportional to $t^2$ while gyroscope scale factor error causes position error proportional to $t^3$.

(c) **Non-Orthogonality**

Axes non-orthogonality error is caused due to imperfections in mounting individual sensors at the time of manufacturing. Along with non-orthogonality errors, there could be misalignment error which is caused due to misalignment in mounting sensor axis of inertial sensors and the orthogonal axis of the body.

(d) **In-run errors (Thermal)**

In-run errors occur due to drift in bias or scale factor error during a run. The in-run bias or scale factor errors are due to effect of environmental parameters like temperature. These are compensated by developing efficient thermal models to reduce the effect of temperature dependent errors on the output signals [46].

(ii) **Random errors**

Random errors are random variations of bias and scale factor errors over time and lead to bias and scale factor drift. The drift may also occur due to the sensor noises that interfere with the output signal. These sensor noises contain a low frequency and a high frequency component [47]. The low frequency component has correlated noise characteristics while the high frequency is characterized by white noise. The high frequency components are removed using de-noising methods like using a low pass filter, a wavelet, or neural networks. The low frequency component needs to be modeled using random processes like, Random constant model, Random walk model, Gauss-Markov (GM) model, and
Auto Regressive (AR) model [48, 49-53]. These processes model the IMU error by calculating autocorrelation or Allan variance function of the noise. The obtained autocorrelation or Allan variance function produces first-order GM or higher order auto-regressive model parameters [54-59]. The development of an accurate stochastic model is required to compensate IMU errors and thus builds a reliable Inertial Navigation System.

As discussed above, with the advent of MEMS technology, very-low cost IMU systems have come into reality. These errors build up over time, corrupting the precision of the measurements and degrading the navigation solution. Depending upon the grade of IMU, accelerometer bias and gyro drift rate vary. Typical values are as depicted in Table 2.1.

**TABLE 2.1:** Typical inertial sensor errors magnitude

<table>
<thead>
<tr>
<th>Grade</th>
<th>Gyro Angle Random Walk[deg/\sqrt{hr}]</th>
<th>Accelerometer Bias Error [mg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation</td>
<td>.002</td>
<td>.025</td>
</tr>
<tr>
<td>Tactical</td>
<td>.07</td>
<td>.3</td>
</tr>
<tr>
<td>Industrial</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Automotive</td>
<td>5</td>
<td>125</td>
</tr>
</tbody>
</table>

For low-cost navigation, industrial grade IMU’s based on MEMS technology are the preferred choice. However, due to the high bias and drift errors associated with them, their standalone navigation accuracy degrades rapidly. Therefore, for seamless and continuous navigation solution from MEMS sensors, the modeling of errors and their reliable estimation or compensation is mandatory.
2.3 **INS/GPS Integration**

As discussed in previous sections, GPS signals face outages and INS system contain errors which multiply with time thus degrading the INS positioning accuracy. Thus, to obtain a continuous and reliable navigation, an integrated INS/GPS system is preferred which overcomes the standalone GPS and INS drawbacks. In an integrated system, GPS aids INS in its error modeling process thereby improving its long-term accuracy. On the other hand, INS bridges GPS gaps and assists in signal acquisition and reacquisition thus reducing the time and search domain required for detecting and correcting GPS cycle slips (Farrell, 1998; Godha, Salmond, & Smith, 1993; Wong, Schwarz, & Cannon, 1988). Thus for an improved, reliable, and continuous navigation, their synergistic combination is preferred while simultaneously overcoming the individual units drawbacks. The main strengths and drawbacks of these systems are summarized in Tables 2.2 and 2.3.

**TABLE 2.2: Salient Features of GPS and INS**

<table>
<thead>
<tr>
<th>Global Positioning System (GPS)</th>
<th>Inertial Navigation System (INS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Excellent position and velocity accuracy over extended periods of time</td>
<td>• Short term position, velocity, and attitude accuracy</td>
</tr>
<tr>
<td>• Poor measurement output rate</td>
<td>• Excellent measurement rate</td>
</tr>
<tr>
<td>• Frequent signal outages</td>
<td>• No signal outages</td>
</tr>
<tr>
<td>• Ambiguity resolution is required</td>
<td>• No ambiguity resolution problem</td>
</tr>
<tr>
<td>• Uniform accuracy is achieved</td>
<td>• Accuracy decreases with time</td>
</tr>
</tbody>
</table>
The integration of INS and GPS is shown in Figures 2-5 and 2-6. The basic principle of integrated system is to combine INS output with an external aiding source i.e. GPS to limit INS error. The INS error is corrected by taking the difference between INS and GPS data.

**TABLE 2.3:** Advantages of INS/GPS Integrated System

<table>
<thead>
<tr>
<th>INS/GPS Integrated System</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Excellent Position, velocity, and attitude accuracy</td>
</tr>
<tr>
<td>• High data rate</td>
</tr>
<tr>
<td>• GPS signal outages bridged by INS output</td>
</tr>
<tr>
<td>• Ambiguity resolution is feasible</td>
</tr>
</tbody>
</table>

The integration of INS and GPS is shown in Figures 2-5 and 2-6. The basic principle of integrated system is to combine INS output with an external aiding source i.e. GPS to limit INS error. The INS error is corrected by taking the difference between INS and GPS data.

**FIGURE 2-5:** Integrated INS/GPS system during GPS availability
During GPS signal availability, called the training phase (Figure 2-5), a learning algorithm is trained to learn the difference between INS and GPS data, i.e., the INS error. The trained model is used to estimate the INS error during GPS signal outages (Figure 2-6). The difference of INS output and learning algorithm prediction provides the corrected output. Thus, a proper learning algorithm plays a vital role in defining the accuracy of navigation parameters. Various learning algorithms like Kalman Filter, Extended Kalman Filter, and Particle Filter existed in the literature to integrate INS and GPS data. Owing to limitations of filtering techniques as discussed in chapter 1, Artificial Intelligent techniques like Artificial Neural Networks (ANNs) came into existence. The implementation of ANNs is discussed as follows.

2.3.1 Overview of ANN based PUA Model

Artificial neural networks (ANNs) are being widely used as they are capable of modeling any nonlinear system with ease [22, 23, 24]. Basic ANN contains one input layer and one output layer which, in general, have to capture the inherent relation between the input and output. In the article scope, the ANN has to capture the nonlinear relationship relating GPS and INS data. On this regard, we use multi-layer feed forward
networks like the multilayer Perceptron (MLP). Apart from the basic input and output layers, MLP contains one or more hidden layers. Present function of ANN is to fuse the INS and GPS data taking advantage of a multi-layer Perceptron network with three layers namely the input layer, hidden layer, and output layer. All the three layers employ a different activation function. Here we implement Position Update Architecture (PUA) to MLP neural network to integrate the INS and GPS data [14, 22, 60]. A MLP utilizing PUA is shown in Figure 2-7.

**FIGURE 2-7:** The PUA MLP network
A MLP network is trained during the GPS availability, and the trained model is used to predict the required output parameters at the time of outages. As can be seen in Figure 2-7, the inputs to the neural network are INS velocity $V_{INS}(t)$ and azimuth $\Phi_{INS}(t)$, whereas the outputs are the position coordinate differences between two consecutive epochs. The efficiency of the model is characterized by the mean square error. The mean square error can be reduced by proper weight update which is done by making a comparison between the output of the PUA and the actual coordinate differences obtained from the GPS.

$$E(w) = \frac{1}{2} \sum_{p=1}^{N} \sum_{q=1}^{N_y} (y_{pq} - \hat{y}_{pq})^2$$  \hspace{1cm} (2.7)

Where, $N$ represents number of training samples, $N_y$ represents the number of neurons in the output layer, $y_{pq}$ is the desired output, and $\hat{y}_{pq}$ is the predicted output given by:

$$\hat{y}_{pq} = \sum_{k=1}^{m} (z_k \cdot v_{kq}) + v_{oq}$$  \hspace{1cm} (2.8)

$$z_k = \sigma \left( \sum_{i=1}^{n} x_i \cdot u_{ik} + u_{ok} \right)$$  \hspace{1cm} (2.9)

Where $v_{oq}$ represents the $q^{th}$ neuron bias weight parameter in the output layer. The weight link between the hidden layer $k^{th}$ neuron and output layer $q^{th}$ neuron is given by $v_{kq}$. The output of the $k^{th}$ hidden neuron is represented by $z_k$ given in (2.9s), where $\sigma(.)$ is the activation function of the hidden neuron. $x_i$ is the $i^{th}$ input neuron. $u_{ik}$ is the weight link between the $k^{th}$ hidden neuron and $i^{th}$ input neuron and $u_{ok}$ is the $k^{th}$ hidden neuron bias parameter. So, the process of comparing and adjusting the neural network weights is done
iteratively during the GPS signal availability. Thus during outages, the output position coordinate differences are predicted by this trained model. The implementation of PUA utilizing ANN is given in Algorithm 1.

Algorithm 1: ANN implementation for Position Update Architecture (PUA)

Step 1: Identification of inputs and outputs for the present architecture;

Step 2: For the set of inputs and outputs, we train the MLP neural network;

Step 3: We obtain the estimation error by comparing the ANN output with the reference value (GPS values);

Step 4: Updating the weights to reduce the estimation error;

Step 5: Repeat steps 2-4 until GPS outage occurs;

Step 6: During GPS outage we utilize this trained model to get the navigation parameters.

The major drawback with this PUA utilizing MLP is the performance limitation for low cost INS. This limitation is due to the nonlinear input-output functional relationship. To deal with such complex relationships, we need to enhance the prediction accuracy. Hence, we adopt a new methodology called Random Forest Regression (RFR) which is capable of handling complex input-output functional relationships.
Chapter 3

Random Forest Regression for INS/GPS Integration

3.1 Overview of Random Forest Regression (RFR)

Random Forest, proposed by Breiman in 2001, is an improved classification and regression tree method that gained popularity for its robustness and flexibility in modeling the input-output functional relationship appropriately [61, 62]. Such method consists of a collection of regression trees trained using different bootstrap samples of the training data. Each tree acts as a regression function on its own, and the final output is taken as the average of the individual tree outputs. Moreover, due to the RFR built-in cross validation capability carried with the help of out-of-bag samples, it provides a realistic prediction error estimates during the training process, and hence, it is suitable for real time implementation. Furthermore, unlike Neural Networks (NNs), RFR handles the high dimensional data effectively [63]. It is worth to point out here that RFR is being applied to various fields such as language modeling for speech recognition, bioinformatics, species distribution modeling, and ecosystem modeling [64-67].
3.2 RFR Methodology

Random Forest Regression is a non-parametric regression approach. It consists of a set of $M$ trees $\{T_1(X), T_2(X), \ldots, T_M(X)\}$, where $X=\{x_1, x_2, \ldots, x_p\}$, is a p-dimension input vector that forms a forest. The ensemble produces $M$ outputs corresponding to each tree $\hat{Y}_1 = T_1(X), \ldots, \hat{Y}_M = T_M(X)$, where $\hat{Y}_m, m=1, \ldots, M$, is the $m^{th}$ tree output. To obtain the final output, an average of all tree predictions is calculated.

Given an input-output dataset corresponding to PUA i.e., $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, where $X_i, i=1, \ldots, n$ is an input vector containing INS velocities and azimuth and $Y_i$ as the GPS position coordinate differences between two consecutive epochs, the training procedure is adopted as follows:

1. From the available dataset, draw a bootstrap sample i.e., a randomly selected sample with replacement.

2. Evolve a tree using the bootstrap sample with the following modifications: at each node choose the best split among a randomly selected subset of $m_{\text{try}}$ descriptors. Here, $m_{\text{try}}$ acts as an essential tuning parameter in the algorithm. The tree is grown to the maximum size (i.e., until no further splits are possible) and not pruned back.

3. Step 2 is repeated until user defined number of trees are grown.

For each regression tree construction, a new training set (bootstrap samples) is drawn with replacement from the original training set. So, while choosing the bootstrap samples some of the training data may be left out of the sample and some may be repeated in the
sample. These left out data sample constitute the out-of-bag samples. A total of two third of the new training sample is utilized for deriving the regression function whereas one third constitutes the out-of-bag sample. Thus, each time a regression tree is constructed using randomized drawn training sample from the original dataset; an out-of-bag sample is used to test its accuracy. This in-built validation features improves the generalization capability of the random forests when an independent test data is utilized.

To obtain the total learning error, an average of the prediction error estimate of each individual tree constructed using their out-of-bag sample is obtained given by (3.1). In (3.1), \( \hat{Y}(X_i) \) is the predicted output corresponding to a given input sample whereas \( Y_i \) is the observed output and \( n \) represents the total number of out of bag samples. This error estimates determine how efficient the random forest prediction would be when it is exposed to unknown/unseen samples.

\[
MSE \approx MSE^{OOB} = n^{-1} \sum_{i=1}^{n} \left( \hat{Y}(X_i) - Y_i \right)^2
\]

(3.1)

Figure 3-1 represents the random forests workflow. The input training samples \( \{ X_1, X_2, \ldots, X_n \} \) shown on left are used to grow the user defined number of trees. However, to define the out-of-bag error estimate, the testing samples i.e., the inputs for prediction (as shown in Figure 3-1) are passed along the trees and the final output is the average of individual tree estimates. Thus, based on the final predicted and observed output, the out-of-bag error estimate is calculated using (3.1).
3.3 Application to INS/GPS Integration

In our study, a Random Forest Regression methodology using Position Update Architecture (PUA) is proposed. PUA model, as explained in chapter 2.3.1, utilizes INS derived velocity and azimuth components as input and the output as the corresponding position coordinate differences between two consecutive epochs, taken from the GPS. Thus, the PUA is trained using a suitable modeling technique as long as the GPS signals are available. In case of outages, the trained PUA model predicts the position coordinate differences using INS solution as input. Originally, the PUA model proposed by El-Sheimy is based on neural network [14]. In an effort to improve the PUA prediction accuracy, a RFR model is utilized to develop an input-output functional relationship between the input-output dataset. In RFR, M trees are fully grown using the available training samples. The newly constructed regression trees can then be utilized for
prediction corresponding to unknown samples. The detailed implementation of the proposed RFR based PUA is given in algorithm 2.

**Algorithm 2:** Proposed RFR based PUA training procedure

**Step 1:** Identify the input and output for developing Random Forest Regression model as per Position Update Architecture (PUA);

**Step 2:** Decide the number of trees to be grown in forests.

**Step 3:** Grow each tree using bootstrapped sample taken from the original dataset;

**Step 4:** Evaluate the prediction error corresponding to the regression trees grown, using out of bag sample and obtain the average error;

**Step 5:** During GPS outages obtain the trees output for a given input (i.e., INS velocity and azimuth) and average the predictions to obtain the final output.

The quality of the developed RFR model depends on the selection of optimal number of trees to be grown in forests. The proposed RFR model is validated and compared with existing Artificial Neural Network (ANN) model using real field test data obtained using low grade MEMS based IMU, high grade IMU and Differential GPS (DGPS) solution under both GPS outages and no outages conditions.

### 3.4 Results

The effectiveness of any integration methodology depends on the percentage of reduction in the standalone INS positional error compared to the existing technique, during the period of GPS outages. In our study, we evaluated our proposed RFR based INS/GPS integration technique against existing ANN based PUA, as it has been proven to work
better than KF. The field test data was collected using Crossbow IMU 300CC-100, reference high grade IMU by Honeywell (HG 1700), Novatel OEM GPS receivers and a computer. The IMU data collection rate was 100Hz and their specifications is shown in Table 3.1 [18].

**TABLE 3.1: Characteristics of Crossbow IMU and HG 1700**

<table>
<thead>
<tr>
<th></th>
<th>Crossbow</th>
<th>HG 1700</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gyroscope</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>$&lt;\pm 2.0^\circ/s$</td>
<td>$1.0^\circ/hr$</td>
</tr>
<tr>
<td>Scale factor</td>
<td>$&lt; 1%$</td>
<td>$150 \text{ ppm}$</td>
</tr>
<tr>
<td>Random Walk</td>
<td>$&lt;2.25^\circ/\sqrt{\text{hr}}$</td>
<td>$0.12^\circ/\sqrt{\text{hr}}$</td>
</tr>
<tr>
<td>Accelerometer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>$\pm 30.0 \text{ mg}$</td>
<td>$1.0 \text{ mg}$</td>
</tr>
<tr>
<td>Scale factor</td>
<td>$&lt; 1%$</td>
<td>$300 \text{ ppm}$</td>
</tr>
<tr>
<td>Random Walk</td>
<td>$&lt;0.15 \text{ m/s/}\sqrt{\text{hr}}$</td>
<td>$0.019 \text{ m/s/}\sqrt{\text{hr}}$</td>
</tr>
</tbody>
</table>

The novel hybrid method is implemented using a programming language and statistical software environment called R [68]. The R is an open source software developed by Ross Ihaka and Robert Gentleman from Auckland University. R incorporates all standard statistical tests, models, and analyses including a comprehensive language for managing and manipulating data. The R computing structure is organized in packages which are a combination of respective codes, data and documentation. R has over 4800 packages for various applications like econometrics, data mining, spatial-analysis, and bio-informatics. The package ‘randomForest’ is used to implement Random Forest Regression. The ANNs are implemented using *Neuromodeler* software [21, 69]. The results from the R
and Neuromodeler software’s are used to create maps using online tool called GPS Visualizer and are demonstrated in further sections.

Figure 3-2 depicts the field test trajectory that comprises of all the real-life scenarios encountered by a typical land vehicle which includes high speed highway section, suburban roads with hills, trees and winding turns, urban streets with frequent stops and sudden vehicle accelerations/decelerations. Four simulated GPS outages of 30 seconds each were considered to evaluate the proposed methodology effectiveness against PUA, as shown in Figure 3-2. These outages were considered under diverse conditions such as straight portions, turns, slopes, high speeds, and slow speeds as is encountered in real time.

**FIGURE 3-2:** Field test trajectory showing simulated GPS outages (in Blue)
The proposed Random Forest Regression model is trained during GPS availability, and the trained model is then used to predict the position coordinates during outages. The training samples contain the INS velocity and azimuth and the output is the GPS position coordinate differences between consecutive epochs. The training is done by selecting a bootstrap sample from the training data and thus growing trees. These newly constructed regression trees are then used for predicting the output for unknown samples corresponding to GPS outages. The final output will be the average of predictions of all the trees.

The PUA model considered in this study utilizes INS velocity and azimuth as input and the corresponding position coordinate differences between two consecutive epochs (taken from GPS) as the desired output. Thus, the PUA model based on ANN is trained as long as the GPS signals are available whereas in the case of outages, the trained model is utilized to predict the position coordinates difference [14, 70]. The ANN is trained using Quasi-Newton training algorithm because of its faster convergence ability [71, 72]. The model performance parameter is evaluated with Root Mean Square Error (RMSE), given in (3.2); by comparing the predicted position components obtained using the proposed methodology (RFR) and the existing neural network based PUA with the reference solution (GPS).

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{N} (\hat{y}_p(x_i, w) - y_p)^2}{N}}.
\] (3.2)

Where \( \hat{y}_p \) and \( y_p \) are the predicted and the desired output and \( N \) corresponds to the GPS outage duration.
Figure 3-3 illustrates the 1st GPS outage of duration 30 sec. The drift in the positional error using RFR methodology (in red) is less when compared to PUA (in green). The RMSE is calculated for both the methodologies and is obtained as 35.55m for RFR and 46.82m for PUA. Thus from the values of RMSE we observe that the percentage improvement in the positioning accuracy is 24.06%.

![Figure 3-3: Performance during GPS outage 1](image)

The second outage is considered to be a more challenging one as it corresponds to the portion of the trajectory along a curve. The predicted trajectories obtained using the proposed and existing model, along with reference trajectory are as shown in Figure 3-4. Here the RFR method obtained a reduction in RMSE from 274.6m to 128.2m, thereby showing a 53.3% improvement in positional error when compared to PUA.
From Figure 3-5 it is clearly evident that for $3^{rd}$ outage the RFR trajectory shows a negligible positional drift from reference trajectory (blue) compared to PUA. The RFR model also reduces the positional error which is 107.2m for neural network based PUA model to 47.44m thus demonstrating an overall percentage improvement of 55.75% in positioning accuracy.

**FIGURE 3-4:** Performance during GPS outage 2
Figure 3-6 depicts the predicted trajectory of both PUA and RFR methodologies for the 4th outage. The proposed RFR model produces a RMSE of 82.5m which is less in comparison to standard PUA which is 136.6m. Thus, apart from improvement in positional drift, a total of 39.6% improvement in positional accuracy was demonstrated.
For all the four simulated GPS outages considered in this study, the RFR algorithm was able to reduce the time growing positional error associated with standalone INS solution effectively. A quantitative comparison of the accumulated position error using our proposed RFR algorithm in comparison to the conventional ANN based PUA model is shown in Table 3.2.

**TABLE 3.2:** Position errors for the proposed RFR model and the conventional PUA model

<table>
<thead>
<tr>
<th>GPS outage length (m)</th>
<th>Total positional error (m)</th>
<th>Percentage improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ANN</td>
<td>RFR</td>
</tr>
<tr>
<td>Outage 1 (30 sec)</td>
<td>317</td>
<td>46.82</td>
</tr>
<tr>
<td>Outage 2 (30 sec)</td>
<td>348</td>
<td>274.64</td>
</tr>
<tr>
<td>Outage 3 (30 sec)</td>
<td>472</td>
<td>107.2</td>
</tr>
<tr>
<td>Outage 4 (30 sec)</td>
<td>394</td>
<td>136.33</td>
</tr>
</tbody>
</table>

From Figures 3-3 to 3-6 we can notice that the navigation accuracy of the proposed RFR model was found to outperform conventional neural network based PUA model. We performed this study considering a low-grade INS which may produce huge positional drifts for longer periods of GPS outages. Hence we considered outage periods of 30 seconds each and thus can observe that the proposed RFR model produces less positional drift compared to conventional PUA. The proposed methodology is capable of delivering stable output for the case where vehicle experience sudden change in its movement as demonstrated through outage 2 (Figure 3-4). We trained the RFR model for a fixed
number of regression trees, which are 500. The system accuracy depends on the number of trees to be grown. Thus by varying the number of trees the error value and the positional drift can be varied in case of RFR model.

3.5 Conclusion

In an effort to develop an improved integration methodology for INS and GPS integration, this chapter introduced the Random Forest Regression model. RFR based non-linear modeling has various advantages such as it avoids data over fitting and offers high dimensionality. It models the non-linear functional relationship between the INS solution and the corresponding GPS position coordinate differences between two consecutive epochs, when the GPS signal is available. However, in the case of outages, the developed model utilizes the INS solution as inputs and predicts the corresponding position coordinate differences and thus provides the reliable estimates. This research fulfills the basic goal of reducing the drift in the predicted position coordinate differences using low-cost IMUs. Through the simulated GPS outages considered in this study, it has been shown that RFR model improves the standalone INS accuracy in comparison to ANN based PUA model. Overall the percentage improvement in the position accuracy was found to be 56%.
Chapter 4

A Hybrid PCR-RFR INS and GPS Fusion Algorithm

4.1 Overview of Principal Component Regression (PCR)

Principal Component Regression (PCR) [73, 74] is a multivariate linear regression method used to model linear input-output relationships. Most of the linear regression approaches face the problem of multicollinearity [75] i.e., the independent variables being correlated with each other, but PCR removes multicollinearity by employing a multivariate statistical technique called Principal Component Analysis (PCA) [76,77]. PCA maps the input data into a set of independent variables called Principal Components (PC) which are a linear combination of input variables. The number of principal components is equal to number of independent variables of the training data. The PC’s are ordered such that the first PC explains the maximum variance with the data and the remaining PC’s provide the variance that has not been accounted by its predecessors. Then multiple linear regressions are performed on the obtained PC’s i.e. a regression
equation is built to make the required predictions. The PCR approach is summarized as below.

Do a principal component analysis on the training data to produce a set of principal components, \( PC = \{ PC_1, PC_2, \ldots, PC_N \} \), where \( N \) is number of independent variables.

Each principal component is obtained by (4.1)

\[
P C_i = a_{i1}X_1 + a_{i2}X_2 + \ldots + a_{iN}X_N
\]  

(4.1)

Where, \( X = \{ X_1, X_2, \ldots, X_N \} \) is the input data set, \( i = 1, 2, \ldots, N \), and \( a_i \) is the eigenvector related to a specific input variable. The eigenvectors are obtained using (4.2) and (4.3).

\[
| R - I \lambda | = 0
\]

(4.2)

Where, \( I \) is the unit matrix, \( R \) is variance-covariance matrix and \( \lambda \) is eigenvalue from which we obtain the eigenvectors using (4.3), where \( V \) represents the eigenvector.

\[
(R - I \lambda)V = 0
\]

(4.3)

The obtained PC’s are used to build a linear regression equation given by (4.4).

\[
Y_i = \beta_0 + \beta_1PC_{i1} + \ldots + \beta_NPC_{Ni} + \epsilon_i
\]

(4.4)

Where \( i = 1, 2, \ldots, M \), \( M \) is the number of outputs, \( Y = \{ Y_1, Y_2, \ldots, Y_M \} \), \( \beta_i \) is the regression coefficient, \( PC_i \) represents the principal components and \( \epsilon_i \) is the error associated with regression. The value of these unknown parameters i.e. \( \beta_i \) is estimated using least squares method. The major drawback of the existing PCR is, it fails in handling non-linear data and the existing RFR fails in handling linear data. Thus, a hybrid
approach of PCR and RFR is proposed to effectively handle both linear and non-linear data.

4.2 Proposed Hybrid PCR-RFR Algorithm

To improve the prediction accuracy of a regression model, a hybrid of linear regression model and non-linear regression model has proven to be effective [78, 79, 80]. This hybrid approach is proposed to overcome the limitations of principal component regression (PCR) and random forest regression (RFR). PCR is a linear model, capable of handling only linear data and RFR is a non-linear model which can effectively model non-linear data. Therefore, this hybrid approach combines the advantages of the existing RFR and PCR techniques and effectively handles the data containing linear and non-linear components, thus increasing the prediction accuracy. In our research for the first time, the hybrid approach is proposed and implemented for INS/GPS integration. During GPS availability the PCR-RFR model is trained and the trained model is utilized to obtain navigation parameters thereby maintaining the continuity during GPS outages (as shown in Figure 4-1). As mentioned, a linear regression model is built using PCR which models linear input-output relationship and then the residuals that constitute the non-linear part are modeled using RFR, which is capable of handling non-linear input-output relationships.
**FIGURE 4-1:** Hybrid approach based training model

**FIGURE 4-2:** Hybrid approach based prediction model
Thus, the final output will be combination of linear component predicted by PCR and non-linear component by RFR, given in (4.5) and illustrated in Figure 4-2.

\[ y_i = G_i + N_i \]  

(4.5)

Where the linear component is represented by \( G_i \) and non-linear component by \( N_i \). The inputs for PCR are INS velocity and azimuth represented as \( X = \{ X_1, X_2, \ldots, X_N \} \) and outputs are position coordinate differences between two consecutive epochs represented as \( Y = \{ Y_1, Y_2, \ldots, Y_M \} \). The estimation of these two components is explained next.

First, the linear component \( (G_i) \) is estimated using the principal component regression. The predictions from the PCR are then modified according to (4.6). As the predictions represent the linear component, the residuals from the linear model will represent the non-linear components of the data.

\[ e_i = Y_i - \hat{G}_i \]  

(4.6)

Where \( e_i \) is the residual from the PCR model for \( i^{th} \) input sample, \( Y_i \) is the observed output and \( \hat{G}_i \) is the predicted output by PCR model. Later, RFR is used to model the residuals from PCR model which are non-linear components of the data. Thus, the inputs to the RFR are INS velocity and azimuth and output are residuals from the PCR model. The prediction from RFR is represented as \( \hat{J}_i \). Thus, the combined prediction is expressed as shown in (4.7).

\[ \hat{Y}_i = \hat{G}_i + \hat{J}_i \]  

(4.7)
The implementation of the proposed PCR-RFR method is explained in Algorithm 2.

**Algorithm 2: Proposed PCR-RFR training procedure**

*Step 1:* Identify the input and output for training PCR-RFR model.

*Step 2:* During GPS availability initially perform PCR on the training data.

*Step 3:* Obtain the non-linear residuals from PCR.

*Step 4:* Then employ RFR to model the non-linear residuals from PCR.

*Step 5:* During GPS outages, obtain the output for a given input (i.e., INS velocity and azimuth) and combine the predictions from PCR and RFR to obtain the final output.

The proposed hybrid approach of PCR-RFR is implemented and compared with existing RFR method using real field test data.

### 4.3 Results

For the present application we use two packages namely ‘pls’, which performs Principal Component Regression and ‘randomForest’ for performing Random Forest Regression. The field test trajectory is as shown in Figure 4-3 which contains all real-life scenarios encountered by a typical land vehicle including high speed highway section, suburban roads with hills, trees and winding turns, urban streets with frequent stops and sudden vehicle accelerations/decelerations. To evaluate the proposed hybrid model performance, five GPS outages of varying duration and diverse conditions are considered. For each outage, we compared predicted trajectory obtained using hybrid model and RFR model to the reference trajectory, as shown in Figures 4-4 to 4-8.
The proposed hybrid model is trained during GPS availability and the trained model is utilized for predictions during GPS outages. The PCR, a linear model, is trained with INS velocity and azimuth as inputs and GPS position coordinate differences as output. The RFR is trained with INS velocity and azimuth as inputs and the residuals from the linear model i.e., PCR, as outputs. The trained linear and non-linear models are used to predict the GPS position coordinate differences during outages which are summation of predictions from PCR and RFR. The RFR is trained with INS velocity and azimuth as inputs and GPS position coordinate differences as outputs. During the training, a set of decision trees are grown with different bootstrap samples, randomly taken from the training data. The fully grown trees are used to predict the position coordinate differences during GPS outages. The overall prediction is taken as average of predictions from individual decision trees.

**FIGURE 4-3:** Field test trajectory showing simulated GPS outages
The performance of the proposed PCR-RFR model and the ordinary RFR are evaluated by considering Root Mean Square Error (RMSE). RMSE is calculated by comparing the predictions of the hybrid model and RFR method with the reference solution i.e. GPS data. The RMSE is calculated using (3.2). Figures 4-4 to 4-8 shows the field trajectories obtained using the predictions of RFR and the proposed hybrid model along with reference trajectory, during GPS outages.

Figure 4-4 depicts the first outage which is of 30 sec duration. The trajectory of the proposed PCR-RFR (red) model is observed to have less positional drift from reference trajectory in comparison to the RFR (green). Apart from decrease in positional drift, increase in positioning accuracy is also observed by calculating the RMSE for both, RFR and PCR, methods. The RMSE for the proposed model is obtained to be 19.30 m which is a reduction over RFR method which is 25.23 m. Thus a 23.51 % reduction in positional drift is obtained.

**FIGURE 4-4:** Performance during GPS outage 1
The second outage is considered for 40 sec duration as shown in Figure 4-5. Here, the trajectory of the proposed model shows a very little positional drift from the reference trajectory. The RMSE for the PCR-RFR model is 95.54 m which is a huge improvement over RFR method which is 168.46 m. This shows a 45.06 % improvement in positioning accuracy.

FIGURE 4-5: Performance during GPS outage 2

The third outage corresponds to portion of the trajectory for the vehicle motion along a curved path for duration of 30 sec. Figure 4-6 shows the reduction in positional drift for the hybrid model trajectory in comparison to the RFR model. During this outage, PCR-RFR model was capable of reducing the RMSE from 67.79 m to 58.50 m, in comparison to RFR, thereby resulting in 13.71 % improvement in positioning accuracy even under diverse conditions.
The fourth outage is taken along a straight line for duration of 50 sec. The predicted and the reference trajectories are as shown in Figure 4-7. The positioning error reduces from 80.75 m (RFR) to 59.70 m (PCR-RFR). This leads to a 26.07 % improvement in positional accuracy. Thus, the proposed model shows an improvement even during extended periods of GPS outages.
FIGURE 4-7: Performance during GPS outage 4

The fifth outage taken for a period of 30 sec is shown in Figure 4-8. It is clearly evident that the proposed model is capable of reducing the positional drift effectively compared to RFR. A total of 22.97 % improvement in positional error is observed with RMSE of 39.81 m for PCR-RFR model and 51.69 m corresponding to RFR.
Thus, the proposed hybrid approach of PCR-RFR has shown a reduction in positional drift in comparison to the RFR for all the five GPS outages considered in this study. Corresponding reduction in INS associated time growing position error was also observed. The positioning errors for all the outages for PCR-RFR in comparison to RFR are tabulated in Table 4.1.
Tables 4.1: Position errors for the proposed PCR-RFR model and the RFR model

<table>
<thead>
<tr>
<th>GPS outage length (m)</th>
<th>Total positional error (m)</th>
<th>Percentage improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RFR</td>
<td>PCR-RFR</td>
</tr>
<tr>
<td>Outage 1 (30 sec)</td>
<td>161</td>
<td>25.23</td>
</tr>
<tr>
<td>Outage 2 (40 sec)</td>
<td>426</td>
<td>168.46</td>
</tr>
<tr>
<td>Outage 3 (30 sec)</td>
<td>355</td>
<td>67.79</td>
</tr>
<tr>
<td>Outage 4 (50 sec)</td>
<td>191</td>
<td>80.75</td>
</tr>
<tr>
<td>Outage 5 (30 sec)</td>
<td>358</td>
<td>51.69</td>
</tr>
</tbody>
</table>

Figures 4-4 to 4-8, clearly demonstrate the enhanced performance of proposed PCR-RFR model when compared to RFR model for INS/GPS integration. A low-grade INS is considered for the study which may produce huge positional drifts and increased positioning error for longer periods of GPS outages. Hence, we considered GPS outages of short durations of 30 sec for which the proposed model showed an improvement in reduction of positional drift and positional error. The accuracy of the hybrid model is further tested for long duration of GPS outages. The proposed model demonstrated improved prediction accuracy even for extended periods of GPS outages i.e., 40 sec (Figure 4-5) and 50 sec (Figure 4-7). The accuracy of the proposed model is also tested under diverse condition such as along curved path (Figure 4-6) and hence demonstrated improved performance.
4.4 Conclusion

This chapter introduces a hybrid PCR-RFR model for integrating GPS and INS data to bridge the GPS signal outages. The PCR handles the linearity present in the data and RFR captures the non-linearity left in the residuals that PCR couldn’t capture. This hybrid approach aggregates the advantages of both linear regression and non-linear regression models leading to a substantial decrease in positional drift and positioning error, given by RMSE, in comparison to existing RFR method. The proposed model which has been validated for different outages under diverse conditions and varying durations has shown a total of 14-45% improvement in prediction accuracy in comparison to random forest regression model. Thus, the results obtained demonstrates the potential of PCR-RFR model for bridging the period of GPS outages effectively and thereby making the navigation continuous and reliable.
Chapter 5

Quantile Regression Forest Integration Methodology

5.1 Overview of Quantile Regression Forest (QRF)

Quantile Regression Forests (QRF), proposed by Meinshausen, is a generalization of Random Forests (RFs) [81, 82]. RFs approximate the conditional mean, whereas QRFs hold the values of all the samples in a node and assess the conditional distribution of the response variable by accurately approximating its conditional quantiles [83]. Thus, QRFs provide a non-parametric and accurate way of estimating conditional quantiles for high-dimensional predictor variables.

5.2 QRF Methodology

Let $X = \{x_1, x_2, \ldots, x_p\}$, be a p dimensional predictor variable, which is used to grow a set of N trees $\{T_1(X), T_2(X), \ldots, T_N(X)\}$ to form a forest and $Y = \{y_1, y_2, \ldots, y_n\}$ is the response variable where, $y_n n = 1, 2, \ldots, N$ is the output of $n^{th}$ tree. From the available training data, a boot strap sample is drawn with replacement. During training, N number
of trees are grown using different bootstrap samples and these fully grown trees are used for prediction. The primary difference between Quantile Regression Forests and Random Forest Regression (RFR) is that, for each node in each tree, RFR keeps only means of the observations that fall into this node and neglects all other information. Thus, Random Forests predict by estimating the conditional mean which is analogous to weighted mean of the observed response variable. The estimation of conditional mean is done by minimizing the expected squared error loss as given in (5.1).

\[
E(Y / X = x) = \arg\min_{\mu} E\{(Y - \mu)^2 | X = x\}, \tag{5.1}
\]

Where, \(X\) and \(Y\) are real valued predictor and response variables respectively and \(\mu\) is the estimate of the response variable. However, Quantile regression forests assess the conditional distribution and not just their mean by keeping the value of each observation in a node. The conditional distribution is estimated by weighted distribution of observed response variable by approximating the conditional quantiles. The conditional quantiles are obtained by minimizing the expected loss as given in (5.2)

\[
Q_\alpha(x) = \arg\min_q E\{L_\alpha(Y, q) | X = x\}, \tag{5.2}
\]

Where, \(Q_\alpha\) is the \(\alpha\) - quantile and \(L_\alpha\) \((0 < \alpha < 1)\) is the loss function given as (5.3)

\[
L_\alpha(Y, \mu) = \begin{cases} 
\alpha |Y - \mu| & Y > \mu \\
(1-\alpha)|Y - \mu| & Y \leq \mu
\end{cases} \tag{5.3}
\]

Where, \(Y\) and \(\mu\) are actual and estimate response variables. Thus, in this paper we propose Position Update Architecture (PUA) based QRF to integrate INS and GPS data. The training samples to the QRF model utilizes INS derived velocity and azimuth as inputs, and the desired outputs are the corresponding GPS position coordinate differences
between consecutive epochs. The model continues to train as long as the GPS signals are available. However, during outages, the trained model is utilized to predict an accurate and reliable navigation solution utilizing the INS solution as input. The training is done by selecting different bootstrap samples from the training data and thus growing trees. The final output is taken as the average of all the tree predictions.

**Algorithm 3:** Proposed QRF working procedure

- **Step 1:** Identify the inputs (INS derived velocity and azimuth) and outputs (GPS position coordinate differences) to develop a QRF model;
- **Step 2:** Decide the number of trees \( N \) to be grown in the forest;
- **Step 3:** Draw a bootstrap sample from the training data and grow a tree to maximum size without pruning;
- **Step 4:** Repeat step 3 to grow user defined number of trees until GPS signal outage occurs;
- **Step 5:** During GPS outages, utilize the fully grown trees to predict the output for a given input.

The proposed QRF is validated using real field test for various GPS outages and is compared to existing ANN and RFR models.

### 5.2 Results

The R software package ‘quantregForest’ is utilized to implement the QRFs. The proposed model is evaluated under diverse conditions by considering four GPS outages of varying duration as shown in Figure 5-1. For each outage the proposed Quantile
Regression Forest model is compared with Random Forest Regression (RFR) model and conventional Artificial Neural Networks (ANN). The RFR model is trained by drawing a bootstrap sample from training data and growing user defined number of trees. The training samples contain INS velocity and azimuth as inputs and GPS position coordinate differences between consecutive epochs as desired output. These fully grown trees are used for prediction during GPS signal outages. The final output is taken as average of individual tree predictions. The ANN model considered here is a 3 layer multi layer Perceptron (MLP), having single input, hidden and output layers. The MLP is trained using quasi Newton algorithm for INS derived velocity and azimuth as inputs and output being GPS position coordinate differences. The trained MLP is tested for unknown INS velocity and azimuth inputs during GPS outages. The effectiveness of the proposed model is compared by evaluating the Root Mean Square Error (RMSE) as given in (3.2)
The first GPS outage is taken for duration of 30 sec as shown in Figure 5-2. It can be observed that the QRF (red) has minimum positional drift from reference trajectory, in comparison to RFR (green) and ANN (blue). The positioning accuracy is evaluated based on RMSE. Thus, QRF has an RMSE of 9.89 m which is a 63.16 % improvement in positioning accuracy comparison to RFR (26.85 m) and 85.84 % compared to ANN (69.86 m).

Figure 5-3 represents second GPS outage of 50 sec duration. In comparison to ANN, the QRF has proven to be effective by reducing the RMSE from 240.79 m to 31.06 m,
showing 87.1 % improvement in positioning accuracy. The QRF even outperformed RFR by reducing the RMSE from 170.18 m to 31.06 m, which is 81.75 % improvement in navigation solution. Thus, the proposed QRF has shown decreased positional drift and improved positioning accuracy in comparison to RFR and ANN.

![FIGURE 5-3: Performance during GPS outage 2](image)

The effectiveness of the proposed model is tested by considering the third outage along a curved path. Figure 5-4 depicts the predicted and reference trajectories. It is clearly evident from the Figure 5-4 that, proposed QRF has reduced the positional drift in comparison to existing RFR and ANN. QRF outperformed both RFR and ANN by reducing the RMSE from 93.89 (RFR) and 106.33 (ANN) to 38.213. Thus, there is a 59.3 % reduction in positional error in case of RFR and 64.06 % compared to ANN.
The accuracy and reliability of the proposed QRF is tested by considering GPS outage for longer duration. Figure 5-5 represents the fourth GPS outage for duration of 60 sec. QRF has an improved navigation solution by reducing the positional drift and improving the positioning accuracy. The reduction in positional drift is clearly evident by observing the predicted and reference trajectories shown in Fig 5-5. The improvement in positioning accuracy is achieved by reducing the positional error from 134.76 m (RFR) and 163.59 m (ANN) to 64.29 m. Thus, the proposed QRF obtained a 52.29 % (RFR) and 60.7 % (ANN) improvements in positioning accuracy.
FIGURE 5-5: Performance during GPS outage 4

For all the four outages the proposed QRF effectively reduced the positional drift and time varying positional error, thus improving the standalone INS accuracy during GPS outages. The positional errors for all the outages for QRF in comparison to RFR and ANN are tabulated in Tables 5.1 and 5.2.

**TABLE 5.1: Position errors for the proposed QRF model and the RFR model**

<table>
<thead>
<tr>
<th>Outage</th>
<th>Total positional error (m)</th>
<th>GPS outage length (m)</th>
<th>Percentage improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outage 1 (30 sec)</td>
<td>26.85</td>
<td>349</td>
<td>63.17</td>
</tr>
<tr>
<td>Outage 2 (50 sec)</td>
<td>170.18</td>
<td>488</td>
<td>81.75</td>
</tr>
<tr>
<td>Outage 3 (40 sec)</td>
<td>93.89</td>
<td>456</td>
<td>59.3</td>
</tr>
<tr>
<td>Outage 4 (60 sec)</td>
<td>134.76</td>
<td>773</td>
<td>52.29</td>
</tr>
</tbody>
</table>
Moreover, the efficiency of QRFs in handling complex data and preventing model overfitting can be attributed to the fact that QRFs consider the entire spread of the response variable and also chooses predictors with high explanatory power. Thus, QRFs are a promising technique to provide reliable, accurate and continuous navigation solution during GPS outages.

### 5.3 Conclusion

This chapter fulfills the basic aim of the research to improve the low-cost INS/GPS integrated system accuracy. This is achieved by introducing Quantile Regression Forests (QRF) which effectively models the non-linear relationship between INS and GPS data. The proposed QRFs are tested on real-field test data by considering GPS outages under diverse conditions and varying durations. The effectiveness of the proposed model in improving the positioning accuracy is established by comparing QRFs to existing Random Forest Regression (RFR) and Artificial Neural Networks (ANN) models. The
QRF has shown 52-81 % improvement in positioning accuracy in comparison to RFR and 60-87 % compared to ANN. The property of approximating the conditional quantiles allows QRF to outperform existing techniques even for limited training data and during longer GPS outages.
Chapter 6

Conclusion and Future work

6.1 Conclusion

The goal of the research was to improve the accuracy of a low-cost INS and GPS integrated system. A proper learning algorithm has to be utilized to learn the INS error during GPS signal availability. This trained model is used to predict the INS error during GPS outages. Existing Bayesian Filtering and Artificial Intelligent techniques failed to model highly non-linear data, affecting the navigation parameters obtained from INS/GPS integrated system. To overcome the limitations of existing techniques, statistical learning approaches were proposed. Chapter 3 introduces Random Forest Regression (RFR) which effectively models highly non-linear data and prevents data overfitting. The proposed RFR was compared with existing Artificial Neural Networks (ANN) and showed a 56% improvement in positioning accuracy. Though RFR had improved performance, it failed to handle both linear and non-linear data effectively. Thus, chapter 4 introduces a hybrid of Principal Component Regression (PCR) and RFR for bridging GPS cycle slips. Here, PCR modeled the linearity present in the data and RFR modeled the non-linearity. This proposed hybrid approach was compared with RFR
and showed a maximum of 45% improvement in positioning accuracy. To further improve the INS/GPS integrated system accuracy chapter 5 introduced Quantile Regression Forests (QRF). The effectiveness of QRF was established by comparing it with existing ANN and RFR. QRF showed a maximum of 81% improvement in positioning accuracy in comparison to RFR and 87% in comparison to ANN. Thus, all the three proposed statistical learning algorithms proved to be effective than existing techniques in providing continuous, accurate and reliable navigation.

### 6.2 Future work

This thesis demonstrated the advantages of statistical learning techniques in improving the accuracy of a low-cost INS/GPS integrated system. The implemented RFR and QRF involve growing user defined number of trees during GPS signal availability. The effectiveness of proposed techniques can be further extended by varying the number of trees during the training phase. The hybrid of PCR-RFR effectively handles both linear and non-linear data, where PCR models linear data. Thus, other statistical techniques can be explored to model linear data. All these approaches require training data before implementation. Further improvements can be made by implementing Online learning algorithms.
References


70


[49] [cited 2011; Available from: http://www.cloudcaptech.com/crista_imu.shtm.]


Appendix A
Source Code

A.1 R Software Commands for executing ‘randomForest’ package

```r
train_data<-read.xlsx("training.xlsx",1)

test_data<-read.xlsx("testing.xlsx",1)

train_inp<-subset(train_data, select = c(-y4,-y5))

train_out<-subset(train_data, select = c(y4,y5))

test_inp<-subset(test_data, select = c(-y4,-y5))

test_out<-subset(test_data, select = c(y4,y5))

rf<-randomForest(train_inp, train_out[,1], test_inp, test_out[,1], keep.forest = TRUE)

pred<-predict(rf,test_inp)

y<-data.frame(pred,test_out[,1])

write.xlsx(y, file = "pred_test.xlsx", row.names = FALSE)
```

A.2 R Software Commands for executing ‘pls’ package

\begin{verbatim}
train_data<-read.xlsx("training.xlsx",1)

test_data<-read.xlsx("testing.xlsx",1)

train_inp<-subset(train_data, select = c(-y4,-y5))

train_out<-subset(train_data, select = c(y4,y5))

test_inp<-subset(test_data, select = c(-y4,-y5))

test_out<-subset(test_data, select = c(y4,y5))

gps<-pcr(y4~y1+y2+y3, scale=TRUE, data=train_data)

pred<-predict(gps,test_inp)

y<-data.frame(pred,test_out[,1])

write.xlsx(y, file = "pred_test.xlsx", row.names = FALSE)
\end{verbatim}
A.1 R Software Commands for executing ‘quantregForest’ package

train_data<-read.xlsx("training.xlsx",1)

test_data<-read.xlsx("testing.xlsx",1)

train_inp<-subset(train_data, select = c(-y4,-y5))

train_out<-subset(train_data, select = c(y4,y5))

test_inp<-subset(test_data, select = c(-y4,-y5))

test_out<-subset(test_data, select = c(y4,y5))

qrt<-quantregForest(train_inp, train_out[,1])

pred<-predict(qrt,test_inp)

y<-data.frame(pred,test_out[,1])

write.xlsx(y, file = "pred_test.xlsx", row.names = FALSE)
Appendix B

Snapshots of GPS Visualizer and Neuromodeler

B.1 GPS Visualizer web page

FIGURE B-1: Snapshot of GPS Visualizer web page
B.2  *Neuromodeler* software main window

![Neuromodeler software main window](image)

**FIGURE B-2:** Snapshot of *Neuromodeler* software main window
Appendix C

Reference Frames

The motion of a navigation system is defined in a reference frame. The various reference frames used in this thesis (chapter 2.2) are described here [39, 40]. Reference frames are broadly classified into two types:

(i) Inertial reference frame

(ii) Non Inertial reference frame

(i) Inertial Reference Frame (i-frame)

Inertial reference frame is an ideal reference frame where all the bodies are with zero acceleration i.e. either stationary or moving with constant speed. The alignment of an axis in an i-frame (as shown in Figure C-1) is:

Origin – The center of mass of the Earth

X axis – The x axis point towards the vernal equinox i.e. the intersection of Earth’s equatorial plain and orbital plane.

Z axis – This axis is parallel to Earth’s polar axis.

Y axis – The y axis is considered perpendicular to X and Z axes.
(ii) Non Inertial Reference Frame

Non Inertial reference frames are considered to be undergoing acceleration. The most widely used non inertial frames of reference are:

a) Earth Centered Earth Fixed Frame ($e$-frame)

The $e$-frame, as shown in Figure C-2, is a non-stationary frame which rotates with respect to the $i$-frame at an angular rate of 15.04 degree/hour about the polar axis.

Origin – Earth’s center of mass

X axis – The x axis passes through the Greenwich meridian in equatorial plane ($0^\circ$ latitude and $0^\circ$ longitude).

Z axis – The z axis is pointed along the Earth’s polar axis.

Y axis – The y axis is considered perpendicular to both x and z axes.
FIGURE C-2: Earth Centered Earth Fixed frame

b) Local Level Frame (l-frame)

The l-frame can be either a North-East-Down (NED) or a East-North-Up (ENU) system.

The coordinates for the l-frame are (Figure C-3):

Origin – Any point relative to Earth is considered as center.

For NED system:
X axis – The x axis points towards the ellipsoidal north (N)
Z axis – The z axis is orthogonal to reference ellipsoidal pointed downwards (D)
Y axis – The y axis is considered perpendicular to x and z axes along East (E)

For ENU system:
X axis – The x axis is on horizontal plane, pointing towards geodetic East (E)
Z axis – The z axis is pointed Upwards (U) along the ellipsoidal normal
Y axis – The y axis is considered perpendicular to x and z axes along North (N)
c) Body Frame (\(b\)-frame)

The \(b\)-frame is shown in Figure C-4, it is represented with respect to a particular vehicle.

The axis of the \(b\)-frame depends on the used \(l\)-frame.

Origin – Center of mass of the vehicle

For NED \(l\)-frame:

X axis – The x axis is aligned with forward direction of the vehicle

Z axis – The z axis is pointed downwards in the direction of gravity

Y axis – The y axis is the lateral direction
FIGURE C-4: Body frame related to NED and ENU frames

For ENU $l$-frame:

X axis – The x axis is aligned in the lateral direction

Z axis – The z axis points upwards

Y axis – The y axis is aligned with the forward direction of the vehicle