1-1-2007

Prediction of High Rate GPS Satellite Orbit Using Artificial Neural Networks

Hamad Yousif
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>

Part of the [Civil and Environmental Engineering Commons](#)

Recommended Citation

Yousif, Hamad, "Prediction of High Rate GPS Satellite Orbit Using Artificial Neural Networks" (2007). *Theses and dissertations*. Paper 129.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

618197401

TL
695
X68
2007

**PREDICTION OF HIGH RATE GPS SATELLITE ORBIT USING ARTIFICIAL
NEURAL NETWORKS**

by

Hamad Yousif

BSc., Khartoum University

1991

A THESIS

SUBMITTED TO RYERSON UNIVERSITY

SCHOOL OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF APPLIED SCIENCE

IN THE PROGRAM OF

CIVIL ENGINEERING

DEPARTMENT OF CIVIL ENGINEERING

RYERSON UNIVERSITY, TORONTO, ONTARIO

October 2007

© Hamad Yousif 2007

PROPERTY OF
RYERSON UNIVERSITY LIBRARY

DECLARATION

I hereby declare that I am the sole author of the thesis. I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or part, at the request of other institutions or individuals for the purpose of scholarly research.

Hamad Yousif

ABSTRACT

Hamad Yousif

PREDICTION OF HIGH RATE GPS SATELLITE ORBIT USING ARTIFICIAL NEURAL NETWORKS

MASc., Civil Engineering, Ryerson University

2007

Precise real-time GPS orbit at a high rate is required for a number of applications, including real-time Precise Point Positioning (PPP), long range RTK and weather forecasts. To support these applications, the International GNSS Service (IGS) has developed a precise orbital service. At present, users may take advantage of the predicted part of the IGS ultra-rapid orbit for real-time and near real-time applications. Unfortunately, however, the data rate of such precise orbits is usually limited to 15 minutes. In addition, the precision of the predicted part of the IGS ultra-rapid orbit is limited to about 10 cm for the 24-hour predicted part, which may not be sufficient for the above applications. This research proposes algorithms for interpolation and prediction methods that are intended to reduce the effect of such limitations.

This research examines the performance of four interpolation methods for IGS precise GPS orbits, namely Lagrange, Newton Divided Difference, Bernese Polynomial, Cubic Spline and Trigonometric Interpolation. In addition, a comparison between this research

and earlier studies was conducted. A new approach that utilizes the residuals between the broadcast and precise ephemeris to generate a high-density precise ephemeris is also introduced in this research.

A three-step neural network-based model is then developed in this research to generate a 6-hour predicted orbital arc. First, an initial predicted orbit is generated by extrapolating a concatenated group of previous precise ephemeris for 5 days. GPS observations from 35 globally distributed tracking stations, corresponding to the 24-hour period preceding the predicted part, are then utilized within the Bernese software to further enhance the predicted orbit. Finally, the predicted orbit is refined by implementing a modular, three-layer feed-forward back-propagation neural network. A comparison is made between our predicted orbit and the IGS ultra-rapid orbit to verify the efficiency of the newly developed neural network-based model. It is shown that the newly developed neural network-based model improved the orbit prediction by 47%, 22%, and 37% for three randomly selected satellites from Blocks IIA, IIR and IIR-M, respectively.

DEDICATION

To the person who praises nothing but noble values, to my dearest mother.

ACKNOWLEDGEMENT

My endless gratitude should be expressed to Dr. Ahmed El-Rabbany for his unlimited support, generosity, and patience. To him I owe the brightest difference in my life. I would also like to express my deep gratefulness to the committee members, Dr. Said Easa, Dr. Khaled Sennah, and Dr. Ahmed Shaker for sparing some time to review my thesis. Special thanks are also extended to my friends, Hassan Ibrahim, Abdulla Alnaqbi, Mohamed El-Sebeay, Amit Joshi, Mohamed El-Diasty and Mahmoud AbdelGalil who made a piece of fun out of study. Without them, things could have been much more difficult than they really are. My thanks are due to Bernese and NRCan technical teams for their valuable support.

This research is funded in part by NSERC, GEOIDE Network of Centres of Excellence, and Ontario Centres of Excellence (OCE).

Table of Contents

DECLARATION	III
ABSTRACT	IV
DEDICATION	VI
ACKNOWLEDGMENTS	VII
TABLE OF CONTENTS	VIII
LIST OF FIGURES	XIII
LIST OF TABLES	XVII
ABBREVIATION	XVIII
1 INTRODUCTION	1
1.1 MOTIVATION	1
1.2 PREVIOUS STUDIES	2
1.3 METHODOLOGY	4
1.4 THESIS OUTLINES	5
1.5 RESEARCH CONTRIBUTION	6
2 GPS SATELLITE ORBIT	7
2.1 GPS SATELLITE CONSTELLATION	7
2.2 REFERENCE COORDINATES SYSTEMS	10
2.2.1 EARTH-CENTERED INERTIAL REFERENCE	10
2.2.2 EARTH-CENTERED EARTH-FIXED REFERENCE	10
2.3 KEPLERIAN ORBIT	11
2.3.1 KEPLER'S LAWS	11

2.3.2	NEWTON'S FORMULATION OF KEPLER'S LAWS	11
2.3.3	PROOF OF KEPLER'S LAWS	13
2.3.4	DEFINITION OF TRUE ANOMALY, ECCENTRIC ANOMALY AND MEAN ANOMALY.....	18
2.3.5	STEPS OF DETERMINING THE KEPLERIAN GPS SATELLITE POSITION	20
2.3.6	ORBIT REPRESENTATION IN EQUATORIAL SYSTEM	20
2.3.7	DIFFERENTIAL RELATIONS	23
2.3.8	PERTURBED SATELLITE ORBIT	26
2.4	BROADCAST EPHEMERIS	28
2.5	PRECISE EPHEMERIS	31
3	PERTURBING FORCES	38
3.1	OBLATENESS OF THE EARTH	39
3.2	DIRECT TIDAL EFFECT	41
3.3	INDIRECT TIDAL EFFECT	43
3.4	SOLAR RADIATION PRESSURE (SRP)	44
3.4.1	ROCK MODEL	44
3.4.2	YAW-BIAS	45
3.4.3	EXTENDED CODE ORBIT MODEL (ECOM)	46
3.5	EARTH'S ALBEDO	47
3.6	AIR DRAG	48
3.7	RELATIVISTIC EFFECT	48
4	ARTIFICIAL NEURAL NETWORKS	50
4.1	WHAT IS ARTIFICIAL NEURAL NETWORKS ?	50

4.2	NEURAL NETWORKS MODEL	51
4.2.1	MCCULLOCH-PITTS MODEL	52
4.2.2	THE PERCEPTRON	57
4.2.3	ADALINE AND MADALINE MODELS	58
4.2.4	FEED-FORWARD NETWORKS	60
4.2.5	RECURRENT NETWORKS	62
4.2.6	RADIAL BASIS FUNCTION NETWORKS	63
4.2.7	SELF-ORGANIZING NEURAL NETWORKS	64
	(a) Linear Vector Quantization (LVQ)	65
	(b) Self-Organizing Map (SOM)	65
4.2.8	MODULAR AND HIERARCHICAL NEURAL NETWORKS	65
4.3	LEARNING METHODS	67
4.3.1	SUPERVISED LEARNING (ERROR CORRECTION LEARNING)	68
4.3.2	UNSUPERVISED LEARNING	68
4.3.3	REINFORCED LEARNING	69
4.3.4	COMPETITIVE LEARNING (WINNER-TAKES-ALL LEARNING)	69
4.3.5	HEBBIAN LEARNING	69
4.3.6	SUPPORT VECTOR LEARNING	70
4.4	LEARNING ALGORITHMS	70
4.4.1	PERCEPTRON LEARNING ALGORITHM	71
4.4.2	HEBBIAN LEARNING ALGORITHM	72
4.4.3	WIDROW-HOFF LEARNING ALGORITHM	73
4.4.4	DELTA LEARNING RULE	74

4.5	MINIMIZATION FUNCTIONS	76
4.5.1	NEWTON ALGORITHM	76
4.5.2	GAUSS-NEWTON ALGORITHM	78
4.5.3	GRADIENT DESCENT	79
4.5.4	LEVENBERG-MARQUARDT ALGORITHM (LMA)	80
5	ANALYSIS AND MODELING METHODOLOGY	81
5.1	ASSESSMENT OF SEVERAL INTERPOLATION METHODS FOR PRECISE GPS SATELLITE ORBIT	81
5.1.1	LAGRANGE INTERPOLATION METHOD	81
5.1.2	NEWTON DIVIDED DIFFERENCE	82
5.1.3	CUBIC SPILNE INTERPOLATION	83
5.1.4	TRIGONOMETRIC INTERPOLATION	85
5.1.5	RESULTS AND DISCUSSION	87
	(a) Lagrange Interpolation Results	88
	(b) Newton Divided Difference Results	95
	(c) Cubic Spline Interpolation Results	96
	(d) Trigonometric Interpolation Results	97
5.2	INTERPOLATION USING BROADCAST EPHEMERIS	98
5.3	PRECISE ORBIT PREDICTION USING ARTIFICIAL NEURAL NETWORKS	100
5.3.1	INITIAL PREDICTION USING BERNESE SOFTWARE	101
	(a) Preparation of orbit fitting and partial derivatives with respect to orbital parameters	101

(b) Generation of improved orbit using RINEX observation files	103
5.3.2 PREDICTION ENHANCEMENT USING ANN	106
5.3.3 PREDICTION RESULTS	107
6 CONCLUSIONS AND FUTURE RECOMMENDATIONS	113
6.1 CONCLUSIONS	113
6.2 FUTURE RECOMMENDATIONS	114
REFERENCES	116
APPENDIX I: SUPPLEMENTARY INTERPOLATION RESULTS	122
I.1 LAGRANGE INTERPOLATION ERROR FOR PRN 03	122
I.2 LAGRANGE INTERPOLATION ERROR FOR PRN 10	123
I.3 LAGRANGE INTERPOLATION ERROR FOR PRN 20	124
I.4 LAGRANGE INTERPOLATION ERROR FOR PRN 29	125
APPENDIX II: PREDICTION COMPARISON BETWEEN IGS, CODE AND ANN	126
II.1 6-HOUR PREDICTION FOR PRN 04	126
II.2 6-HOUR PREDICTION FOR PRN 13	127
II.3 6-HOUR PREDICTION FOR PRN 17	128

List of Figures

Figure 2.1 Keplerian Orbit	12
Figure 2.2 Kepler second law	14
Figure 2.3 Derivation of Keplerian orbit	18
Figure 2.4 Orbit representation in ECEF reference frame	21
Figure 2.5 Keplerian ephemeris error compared to final ephemeris	23
Figure 2.6 Differential relations	24
Figure 2.7 Osculating ellipses	27
Figure 2.8 Variations of Keplerian parameters within one-week period	30
Figure 2.9 Broadcast ephemeris error compared to final ephemeris	31
Figure 2.10 IGS final ephemeris of August 4, 2007	33
Figure 2.11 IGS Rapid versus Final ephemeris	36
Figure 2.12 Ultra-rapid versus Final ephemeris	37
Figure 3.1 Tidal Effect	42
Figure 4.1 The basic units of ANN	51
Figure 4.2 Basic Structure of a Neuron	53
Figure 4.3 Unipolar Binary Function	53
Figure 4.4 Bipolar Binary Function	54
Figure 4.5 Unipolar Sigmoid Function	54
Figure 4.6 Bipolar Sigmoid Function	55
Figure 4.7 Unipolar Piecewise Linear Function	56
Figure 4.8 Bipolar Piecewise Linear Function	56

Figure 4.9 Radial Basis Function	57
Figure 4.10 ADALINE model structure	59
Figure 4.11 MADALINE model structure	59
Figure 4.12 Single-layer Feed-forward Network	60
Figure 4.13 Multiple-layer Feed-forward Network	61
Figure 4.14 Recurrent Networks Structure	62
Figure 4.15 Structure of Radial Bias Function Networks	64
Figure 4.16 Modular Neural Network	66
Figure 4.17 Hierarchical Mixtures of Experts (HME)	67
Figure 4.18 Perceptron Learning Algorithm in a single-layered multi-neuron network	72
Figure 4.19 An intermediate layer in a multi-layered network	74
Figure 4.20 Minimization of a one-dimensional function using Gradient descent	79
Figure 5.1 Resulting error when taking the whole 24-hour data to build 96 Lagrange interpolating terms	89
Figure 5.2 Resulting error when taking 11 Lagrange terms	90
Figure 5.3 Resulting error when taking 9 Lagrange terms	90
Figure 5.4 Segmentation scheme: 24-hour data is segmented into 23 overlapping segments of 9 terms each	91
Figure 5.5 Interpolation error when the GPS orbit is partitioned into smaller segments without overlapping	92

Figure 5.6 Interpolation error when the GPS orbit is segmented into overlapping segments	93
Figure 5.7 Lagrange Interpolation error for Inertial GPS Orbit	93
Figure 5.8 Lagrange Interpolation error for ECEF GPS Orbit	94
Figure 5.9 Newton Divided Difference Interpolation error for Inertial GPS Orbit	95
Figure 5.10 Newton Divided Difference Interpolation error for ECEF GPS orbit	96
Figure 5.11 Cubic Spline Interpolation error for GPS Inertial Orbit	96
Figure 5.12 Trigonometric Interpolation error for GPS INERTIAL orbit	98
Figure 5.13 Trigonometric Interpolation error for GPS ECEF orbit	98
Figure 5.14 Interpolation error using the broadcast ephemeris	100
Figure 5.15 Residuals of polynomial fitting before orbit improvement	102
Figure 5.16 Distribution of selected IGS stations	104
Figure 5.17 Residuals of the polynomial fitting after improvement	106
Figure 5.18 Prediction Errors for PRN04 (Block IIA)	110
Figure 5.19 Prediction Errors for PRN13 (Block IIR)	110
Figure 5.20 Prediction Errors for PRN17 (Block IIR-M)	111
Figure I.1 Lagrange Interpolation Error for PRN 03	122
Figure I.2 Lagrange Interpolation Error for PRN 10	123
Figure I.3 Lagrange Interpolation Error for PRN 20	124
Figure I.4 Lagrange Interpolation Error for PRN 20	125
Figure II.1 Prediction Error for PRN 04	126
Figure II.2 STD Error for PRN 04	126
Figure II.3 Prediction Error for PRN 13	127

Figure II.4 STD Error for PRN 13	127
Figure II.5 Prediction Error for PRN 17	128
Figure II.6 STD Error for PRN 17	128

List of Tables

Table 1.1 GPS Satellite Constellation	9
Table 2.1 Distribution of orbital parameters within the navigation message	28
Table 2.2 Analysis Centers	34
Table 3.1 Percentage of perturbing acceleration with respect to the mean central acceleration	39
Table 5.1 Comparison between interpolation errors generated by partitioned vs. overlapping segmentation	92
Table 5.2 Comparison between Inertial vs. ECEF Lagrange interpolation error	94
Table 5.3 Statistics of Trigonometric Interpolation using different number of terms ...	97
Table 5.4 Prediction Error (cm) and ANN Percentage improvement	112
Table I.1 Statistics of Lagrange Interpolation Error for PRN 03	122
Table I.2 Statistics of Lagrange Interpolation Error for PRN 10	123
Table I.3 Statistics of Lagrange Interpolation Error for PRN 20	124
Table I.4 Statistics of Lagrange Interpolation Error for PRN 29	125

Abbreviations

ADALINE	Adaptive Linear Neuron
ANN	Artificial Neural Networks
CDDIS	Crustal Dynamics Data Information System
CODE	Centre for Orbit Determination in Europe
DoD	Department of Defence
ECI	Earth-Centered Inertial
ECEF	Earth-Fixed Earth-Centered
ECOM	Extended CODE Orbit Model
ERP	Earth Rotation Parameters
ESA/ESOC	European Space Agency/European Space Operation Centers
GFZ	GeoForschungsZentrum
GNA	Gauss-Newton Algorithm
GNSS	Global Navigation Satellites System
GOP	Geodetic Observatory Pecny
GPS	Global Positioning System
HME	Hierarchical Mixtures of Experts
IERS	International Earth rotation and Reference systems Service
IGS	International GNSS Service
IODC	issue of data clock
I ODE	issue of data ephemeris

ITRF	International Terrestrial Reference Frame
JPL	Jet Propulsion Laboratory
LMA	Levenberg-Maquardt Algorithm
LVQ	Linear Vector Quantization
MADALLINE	Multiple Adaptive Linear Neuron
MIT	Massachusetts Institute of Technology
MLP	Multi-Layer Perceptron
NGS	National Geodetic Survey
NOAA	National Oceanic and Atmospheric Administration
NRCan	Natural Resources Canada
PPP	Precise Point Positioning
PRN	Pseudorange Number
RBF	Radial Basis Function
RINEX	Receiver-Independent Exchange Format
RTK	Real Time Kinematic
SIO	Scripps Institute of Oceanography
SOM	Self-organizing Map
SRP	Solar Radiation Pressure
SVM	Support Vector Machine
SVN	Space Vehicle Number
TCAR	Three-Carrier Ambiguity Resolution
USNO	United States Naval Observatory
WGS	World Geodetic Survey

1 INTRODUCTION

1.1 Motivation

Some applications, such as studies of the crustal dynamics of the Earth, require more precise ephemeris data than the broadcast ephemeris. The Precise Point Positioning (PPP) technique has recently evolved, which also requires the availability of precise ephemeris (as well as satellite clock corrections). To support these applications, several institutions such as the International GNSS Service (IGS), the formerly International GPS Service, have developed post-mission precise orbital service. Precise ephemeris data is based on GPS data collected at a global GPS network coordinated by the IGS. At present, IGS precise ephemeris data is available to users with some delay, which varies from 3 hours for the observed half of the IGS ultra-rapid orbit to 17 hours for the rapid orbit, to about 13 days for the most precise IGS final orbit. The IGS ephemeris products are the most accurate ones; however, the fact that they are spaced at 15-minute intervals renders these products incapable of serving GPS applications without being interpolated at a higher rate.

On the other hand, the ultra-rapid ephemeris is the product that is primarily used for real-time and near real-time applications. The ultra-rapid ephemeris consists of two parts, namely, observed and predicted. The observed half can serve the post-processed applications with almost the same accuracy of the rapid ephemeris, which is less than 5 cm. However, the accuracy of the predicted half is, still, less by 50% than the observed half.

1.2 Previous Studies

Several methods of interpolation have been widely used to generate GPS positions and velocities at intermediate points. Previous studies based on Lagrange method were used by the US National Geodetic Survey (NGS), (Remondi, 1991; Schueler, 1998). Polynomial and Trigonometric interpolation were conducted by Schenewerk (2003). Lagrange, Chebyshev and Trigonometric interpolation methods were used by Feng and Zheng (2005).

Users requiring real-time precise orbit data can use the predicted half of the IGS ultra-rapid orbit data, which are accurate to about 10 cm for the 24-hour (IGS, 2007). The call for ultra-rapid product was initially announced at the IGS Workshop held in La Jolla, California, USA in 1999. The ultra-rapid product was intended to satisfy the need for real-time and near real-time applications that could not be met by rapid and final products. The ultra-rapid ephemeris file contains a 48-hour arc in the sp3 format. The first 24-hour arc is obtained by fitting the data and products that are already available over the first 24-hour period while the next 24-hour period is predicted. In February 2000 (week # 1050), Jet Propulsion Laboratory (JPL) analysis center released its first ultra-rapid product twice a day at 00:00 and 12:00 with an accuracy of 20 – 40 cm for the fitted part and 50 cm for the predicted part. The Centre for Orbit Determination in Europe (CODE) analysis center started releasing its ultra-rapid product in June 2003. Today there are eight analysis centers which submit their ultra-rapid products to the IGS analysis coordinator, to be combined on the basis of weighted averages (Beutler et al, 1995) to generate the official IGS precise ephemerides. The 12-hourly ultra-rapid product was switched to a 6-hourly product in week # 1276 (June 20-26, 2004). Each analysis centre implements its own physical

models of geostationary forces, direct and indirect tidal effects, solar radiation pressure, relativistic effect, tropospheric delay, and others, which are handled by a wide variety of software and procedures. As an example, JPL uses GIPSY/OASIS II for processing hourly RINEX data from several stations (IGS, 2007). The method is based on generating multiple orbital arcs by processing 3-hour observations, using the JPL rapid solution/covariance to initialize the 3-hour orbital arc. These arcs are combined by a process called the inverse sequential smoothing process (IGS, 2007). CODE procedure starts by pre-processing the phase observation in a baseline-by-baseline mode using Bernese 5.0. Cycle slips are detected and fixed using triple differences, and bad data are removed. A 72-hour orbital arc is predicted, using multiple previous precise orbits, and RINEX data from several stations are used to improve the osculating orbital parameters for the predicted orbit. The long arc processed by CODE involves a robust and long-lasting model for solar radiation pressure which justifies the adoption of a 9-parameter SRP model, in addition to ROCK4 and ROCK42 models (Dach et al, 2007). ESA/ESOC (European Space Agency/European Space Operation Centers) uses BAHN software to process RINEX data in a zero-difference ionospheric-free combination. The pseudorange, phase observations, and precise orbit ephemeris from 3 previous days are written to one file. This file is later processed with a group of other files, such as station positions, satellite and station clock biases and satellite dynamic models to produce the ultra-rapid product (Romero et al, 2001). The Geodetic Observatory Pecny (GOP), in the Czech Republic, uses the same procedures and software adopted by CODE to produce their ultra-rapid products (Dousa, 2003). The Geodetic Survey Division (GSD) at Natural Resources Canada (NRCan), started producing ultra-rapid ephemeris product on March 20th, 2000 (IGS, 2007). Since then, the GSD products quality and reliability

has been improving remarkably (GEOIDE, 2006). The variety of strategies and techniques adds to the IGS product quality and reduces the possibility of obtaining biased results.

Although the above-mentioned methods are highly efficient, their prediction accuracy is not high. The accuracy of the predicted part of the ultra-rapid orbit is limited to about 10 cm (the 24-hour predicted part), which is not sufficient for the above applications. To overcome this deficiency, the ANN-based prediction model is developed in this research.

1.3 Methodology

A few interpolation methods are discussed in the first part of this research. These methods employ Lagrange, Newton Divided Difference, Cubic Spline and Trigonometric approximation techniques. An algorithm is developed in this research to achieve an ultimate precision and at the same time avoid computer related errors such as round-off errors and numeric overflow. Different numerical data types are bound to a certain capacity defined by a computer operating system and/or computer language. It was found that unpredicted results were produced when the whole 24-hour data was taken as a single input. After performing many tests, we came to realize that the numeric overflow was the reason behind that behaviour. On the other hand, round-off errors arise from the discrepancy between the exact numerical value and approximated or computed value. The best precision for Lagrange and Newton divided difference was achieved by dividing the data into smaller overlapping segments. The cubic spline, which is inherently a third degree polynomial, was found

to be far from providing a reasonable accuracy. The trigonometric method was the one that closely represents the GPS satellite orbit because of its periodic nature.

The second part of this research pertains to the GPS satellite orbit prediction. The advantages of Artificial Neural Networks (ANN), in conjunction with Bernese GPS software, were used to generate a 6-hour predicted orbital arc. GPS precise ephemeris products of many days, as well as GPS observations from a globally distributed IGS stations, were processed by Bernese to obtain the first phase of this procedure. The predicted ephemeris was enhanced by applying ANN afterwards. The resulted prediction precision was compared to that of the IGS. The comparison was conducted on three randomly selected satellites that represent three different generations currently in service, namely, Block IIA, Block IIR and Block IIR-M.

1.4 Thesis Outlines

The first chapter of this thesis discusses the motivation and methodology adopted to achieve the research objective. The second chapter describes how the GPS orbit is represented using the ideal Keplerian orbit as well as the perturbed orbit. The IGS precise and broadcast ephemeris is also illustrated in this chapter. The third chapter exposes the sources of disturbance that cause the GPS satellite orbit perturbation. The fourth chapter demonstrates in some depth the concept of Artificial Neural Networks (ANN) as well as their different types, algorithms and associated mathematical optimization theorems. The fifth chapter details the data and the relevant models that are designed to fulfill the purpose of this research. Finally, in the sixth chapter, some conclusions are drawn and recommendations for future work are proposed.

1.5 Research Contribution

The contribution of the research can be summarized as follows:

- This research provides an optimization algorithm to generate interpolated GPS satellite orbits with a millimetre level of precision.
- The research introduces a new approach of interpolation that makes use of the broadcast ephemeris.
- The implication of Artificial Neural Networks in the prediction of GPS satellite orbit proved to be viable and led to an appreciable enhancement.
- According to this research, the historical GPS data were found to be of great use and thus they could represent a good source of prediction improvement in future.
- It is shown that the newly developed neural network-based model improved the orbit prediction by up to 47%, 22%, and 37% for three randomly selected satellites from Blocks IIA, IIR and IIR-M, respectively.

2 GPS SATELLITE ORBIT

This chapter discusses the GPS satellite constellation and the different satellite generations in chronological order. The Keplerian orbit representation in the orbital plane as well as the Earth-Centered Earth-Fixed reference is also described, followed by the perturbed satellite motion along with their associated equations of motion.

2.1 GPS Satellite Constellation

The Global Positioning System (GPS) is a satellite-based navigation positioning system capable of determining the position in a common reference system anywhere, at anytime and under all weather conditions. It was originally developed by the US Department of Defence (DoD) in 1973 for military purposes. However, the increasing need for precise positioning made it necessary to release it for civilian applications as well, especially after the US President and Congress directed the DoD to promote its civil use (Leick, 2004). In addition to positioning on land, sea or air, GPS provides continuous timing information. The GPS satellite completes one period around the Earth in almost 11 hours and 58 minutes. The GPS satellites are classified into different chronological block types, namely Block I, Block II, Block IIA, Block IIR, Block IIR-M, Block IIF and Block III. Block I satellites, space vehicle numbers (SVN) 1 to 11, were developed by Rockwell International and launched between February 1978 and October 1985, with a design life of 4.5 years. Each Block I satellite was equipped with one Cesium and two Rubidium atomic clocks. Block II satellites, SVN 13 to 21, were also developed by Rockwell International and launched between February 1989 and October 1990, with a design life of 7.3 years. Block II satellites

contain two Cesium and two Rubidium atomic clocks and were designed to operate autonomously for 14 days without any contact from the control station. Between Nov. 1990 and November 1997, Rockwell International supplied Block IIA satellites, SVNs 22 through 40, whose autonomous period is 180 days. More improvements were introduced when Lockheed Martin developed Block IIR satellites, SVNs 41 through 62, where the so-called cross-linked ranging technique was implemented through which full accuracy could be maintained by updating the parameters in the navigation message without contacting the CS. The first Block IIR-M was launched on September 25, 2005 with the new feature of an enhanced modernized power signal. In addition, the C/A code is carried over by L1 as well as L2 carrier. While Block I and Block II have gone out of service, Block IIF and Block III are expected to appear within the next 20 years (El_Rabbany, 2006). New civil codes (L2 CM and L2 CL) and two military codes M-codes, are proposed to be added to future satellites while a third signal carrier L5 will be introduced in Block IIF satellites (El_Rabbany, 2006). The current GPS constellation (September 19, 2007) consists of 31 satellites (Table 1.1, USNO 2007), distributed over 6 orbital planes around the Earth – denoted by A through F – so that four or more satellites will always be visible to the user anywhere around the globe, at an elevation angle of 15° or greater (Dach et al, 2007). The semi-major axis of the satellite elliptical orbit is about 26,560 km and rises up to an altitude of 20,200 km above the Earth's surface. The eccentricity of the GPS satellite orbit is, on average, 0.01 and is inclined to the equatorial plane with an angle of 63° for Block I and 55° for Block II or later (El-Rabbany, 2006). Due to the Earth's gravitational pull, the GPS satellites normally drift off their allocated orbit. Therefore, a repositioning operation known as Delta-V is performed once year to relocate each

satellite to its original orbit. This operation takes approximately 12 hours of unusable time for each satellite (USNO, 2007).

Table 1.1 GPS Satellite Constellation

BLOCK TYPE		PRN	SVN	PLANE	LAUNCH DATE
IIA	IIA-10	32	23	E6	26-Nov-90
	IIA-11	24	24	D6	4-Jul-91
	IIA-12	25	25	A5	23-Feb-92
	IIA-14	26	26	F2	7-Jul-92
	IIA-15	27	27	A4	9-Sep-92
	IIA-16	1	32	F6	22-Nov-92
	IIA-17	29	29	F5	18-Dec-92
	IIA-20	7	37	C5	13-May-93
	IIA-21	9	39	A1	26-Jun-93
	IIA-22	5	35	B4	30-Aug-93
	IIA-23	4	34	D4	26-Oct-93
	IIA-24	6	36	C1	10-Mar-94
	IIA-25	3	33	C2	28-Mar-96
	IIA-26	10	40	E3	16-Jul-96
	IIA-27	30	30	B2	12-Sep-96
	IIA-28	8	38	A3	6-Nov-97
IIR	IIR-2	13	43	F3	23-Jul-97
	IIR-3	11	46	D2	7-Oct-99
	IIR-4	20	51	E1	11-May-00
	IIR-5	28	44	B3	16-Jul-00
	IIR-6	14	41	F1	10-Nov-00
	IIR-7	18	54	E4	30-Jan-01
	IIR-8	16	56	B1	29-Jan-03
	IIR-9	21	45	D3	31-Mar-03
	IIR-10	22	47	E2	21-Dec-03
	IIR-11	19	59	C3	20-Mar-04
	IIR-12	23	60	F4	23-Jun-04
	IIR-13	2	61	D1	6-Nov-04
IIR-M	IIR-14M	17	53	C4	26-Sep-05
	IIR-15M	31	52	A2	25-Sep-06
	IIR-16M	12	58	B5	17-Nov-06

2.2 Reference Coordinate Systems

The GPS satellite orbit is described by an equation that is based on Newtonian mechanics. The satellite and receiver position vectors are always used in GPS computations. Therefore, a common coordinate system is required as a reference frame for the satellite and receiver positions. Two reference systems of special importance are discussed in this research, namely, the Earth-Centered Inertial (ECI) reference and the Earth-Centered Earth-Fixed (ECEF) reference.

2.2.1 Earth-Centered Inertial (ECI) Reference

This is a right handed Cartesian coordinate system, X, Y and Z whose origin coincides with the Earth's center. The Z axis coincides with the Earth's spinning axis. The X axis points towards the direction of the vernal equinox which is determined by the intersection of the equatorial plane of the Earth and the Earth's orbital plane around the Sun. The Y axis completes the right-handed system.

2.2.2 Earth-Centered Earth-Fixed (ECEF) Reference

The Earth-Centered Earth-Fixed (ECEF) reference system is a right-handed system that shares the same origin and Z axis with the Earth-Centered Inertial (ECI) reference system. The X axis is defined by the intersection of the Greenwich meridian and the equatorial plane of the Earth. The Y axis is orthogonal to both X and Z and thus completes the right handed system.

2.3 Keplerian Orbit

Numerous theorems and hypotheses about the planetary motion were released since several centuries ago. Nicolas Copernicus (1473-1543) was the first astronomer to reveal that the Sun is the centre of what is presently known as the solar system. The circular planetary motion of the Copernican theorem was later refuted by Johannes Kepler (1570-1630), who, in 1605 came out with three mathematical laws that described planetary motion around the sun. Since the GPS satellite revolves around the Earth, its orbit is, in essence, similar to the planets' orbits around the Sun.

2.3.1 Kepler's Laws

First Law:

The planet revolves around the sun in an elliptical path with the sun located at one of its foci.

Second Law:

The planet-sun vector sweeps out equal areas in equal time intervals.

Third Law:

If P and a are the orbit's period and semi major axis respectively, then:

$$P^2 \propto a^3$$

2.3.2 Newton's Formulation of Kepler's Laws:

Later, Isaac Newton (1643-1727) proved Keplerian laws using Newton's laws of motion. Analogously, Kepler's laws apply to the GPS satellite, taking into consideration the Earth's gravitational force, which acts on the satellite.

Assume that a satellite of mass m is revolving around the Earth, whose mass is M_E as shown in Figure 2.1

Figure 2.1 Keplerian Orbit

The satellite is subjected to many forces but for now only the gravitational attractive force $\underline{F}$ is considered. The gravitational force will produce an elliptical or Keplerian orbit. According to Newton's Theory of Universal Gravitation, the force $\underline{F}$ is given by:

$$\underline{F} = -\frac{GM_E m}{r^2} \hat{r} \quad (2.1)$$

or

$$\underline{a} = -\frac{GM_E}{r^2} \hat{r} = -\frac{\mu}{r^2} \hat{r} \quad (2.2)$$

where:

$\underline{r}$ is the satellite's position vector

$\underline{\hat{r}}$ is the radial unit vector

$\underline{\hat{\theta}}$ is the along track unit vector

(r, θ) are the satellite's polar coordinates, θ is also known as the true anomaly

G is the universal gravitational constant

μ is the product of GM which is equal to $3986004.418 \cdot 10^8 \text{ m}^3 \text{ s}^{-2}$

2.3.3 Proof of Kepler's Laws:

In the following context, mathematical derivations of Kepler's laws are illustrated (Beutler, 2005). Taking the vector cross product by $\underline{r}$ for each side of equation (2.2):

$$\underline{r} \times \underline{\ddot{r}} = -\frac{\mu}{r^3}(\underline{r} \times \underline{r}) = 0 \quad (2.3)$$

The right hand side of equation (2.3) may be rewritten as:

$$\frac{d}{dt}(\underline{r} \times \underline{\dot{r}}) = 0 \quad (2.4)$$

Integrating both sides with respect to time:

$$\underline{r} \times \underline{\dot{r}} = \underline{h} = \text{constant} \quad (2.5)$$

Figure 2.2 Kepler's second law

$$\Delta A = \frac{1}{2} |\underline{r} \times \underline{r} \Delta t| = \frac{1}{2} |\underline{h}| \Delta t \quad (2.6)$$

Since $\underline{h}$ is constant, it follows that the radius vector sweeps equal areas at equal intervals.

Now to prove the first law, let $\underline{\hat{r}}$ and $\underline{\hat{\theta}}$ denote to the radial and angular unit vectors respectively. Substitute $\underline{r}$ by $r \underline{\hat{r}}$ in equation (2.2):

$$\frac{d^2(r \underline{\hat{r}})}{dt^2} = -\frac{\mu}{r^2} \underline{\hat{r}} \quad (2.7)$$

Given that $\underline{\hat{r}} = \frac{d\theta}{dt} \underline{\hat{\theta}}$, equation (2.7) reduces to (John, 1989):

$$\left[\frac{d^2 r}{dt^2} - r \left(\frac{d\theta}{dt} \right)^2 \right] \hat{r} - \frac{1}{r} \frac{d \left(r^2 \frac{d\theta}{dt} \right)}{dt} \hat{\theta} = -\frac{\mu}{r^2} \hat{r} \quad (2.8)$$

Equating the radial components and angular components in both sides:

$$\frac{d^2 r}{dt^2} - r \left(\frac{d\theta}{dt} \right)^2 = -\frac{\mu}{r^2} \quad (2.9)$$

$$\frac{1}{r} \frac{d \left(r^2 \frac{d\theta}{dt} \right)}{dt} = 0 \quad (2.10)$$

integrating both sides of equation (2.9) with respect to time:

$$r^2 \frac{d\theta}{dt} = h = \text{constant} \quad (2.11)$$

substitute $\frac{d\theta}{dt}$ by $\frac{h}{r^2}$ in equation (2.8):

$$\frac{d^2 r}{dt^2} - \frac{h^2}{r^3} = -\frac{\mu}{r^2} \quad (2.12)$$

substitute $r = \frac{1}{u}$ and eliminate t:

$$\frac{d^2 u}{d\theta^2} + u = \frac{\mu}{h^2} \quad (2.13)$$

which could be solved as:

$$u = \frac{\mu}{h^2} + C \cos(\theta - \theta_0) \quad (2.14)$$

or

$$r = \frac{\frac{h^2}{\mu}}{1 + C \frac{h^2}{\mu} \cos(\theta - \theta_0)} \quad (2.15)$$

which is an ellipse equation in polar form with eccentricity $e = C \frac{h^2}{\mu}$ and focus-

$$\text{directrix} = \frac{h^2}{\mu}$$

The focus-directrix is equal to the length of the radius vector when it becomes perpendicular to the major axis (i.e. when $\theta - \theta_0 = 90^\circ$). It could be expressed in terms of the semi-major and semi-minor axes as $a(1 - e^2)$.

The third law can be proved by integrating both sides of equation (2.6) with respect to t as follows:

$$A = \int_0^P \frac{1}{2} h dt = \frac{1}{2} hP \quad (2.16)$$

where A is the ellipse area and P is the total period.

If a and b are the semi-major and semi-minor of the ellipse respectively then the area

$$A = \pi ab \quad (2.17)$$

Substitute $A = \pi ab$ in equation (2.16) and let P the subject of the formula:

$$P = \frac{2\pi ab}{h} \quad (2.18)$$

Square both sides and substitute $b^2 = a^2(1 - e^2)$

$$P^2 = \frac{4\pi^2 a^2 b^2}{h^2} = \frac{4\pi^2 a^4 (1 - e^2)}{h^2} = \frac{4\pi^2 a^3 \cdot a(1 - e^2)}{h^2} \quad (2.19)$$

The focus-directrix is equal to $a(1-e^2)$ which can be replaced by $\frac{h^2}{\mu}$ as shown previously in equation (2.15). Hence:

$$P^2 = \frac{4\pi^2 a^3 \cdot \frac{h^2}{\mu}}{h^2} = \frac{4\pi^2}{\mu} a^3 \quad (2.20)$$

from which Kepler's third law follows.

Now, starting with appropriate initial conditions ($\theta_0 = 0$) the orbit can be represented as:

$$r = \frac{a(1-e^2)}{1+e\cos\theta} \quad (2.21)$$

where:

a is the semi major axis

e is the eccentricity

from equation (2.20) the revolution period, P , is given by:

$$P = 2\pi \sqrt{\frac{a^3}{\mu}} \quad (2.22)$$

and the mean angular velocity n as:

$$n = \frac{2\pi}{P} = \sqrt{\frac{\mu}{a^3}} \quad (2.23)$$

Figure 2.3 Derivation of Keplerian orbit

2.3.4 Definition of True Anomaly, Eccentric Anomaly and Mean Anomaly

There are three angular quantities that are generally used for the orbit determination. These are the True Anomaly θ , the Eccentric Anomaly E and the Mean Anomaly M . As shown in Figure 2.3 the true anomaly is the geocentric vector argument; and the eccentric anomaly E is the angle subtended by the X axis and the line joining the ellipse centre with S . The mean anomaly M is an abstract quantity which is given by:

$$M = n(t - t_0) \quad (2.24)$$

t is the instantaneous time

t_0 is the initial epoch

The mean anomaly M and eccentric anomaly E are related by the following equation which is known as Kepler's equation:

$$E = M + \sin E \quad (2.25)$$

Equation (2.21) represents the orbit in terms of geocentric distance r and the true anomaly θ . Another convenient representation is adopted in GPS that takes into consideration the relation between the geocentric distance r and the eccentric anomaly E which is shown in Figure 2.3. From the geometry we can deduce that:

$$\underline{r} = a(\cos E - e) \hat{i} + b \sin E \hat{j} = a(\cos E - e) \hat{i} + a\sqrt{1-e^2} \sin E \hat{j} \quad (2.26)$$

where:

b is the semi minor axis

$\hat{i}$ is the unit vector along the x axis

$\hat{j}$ is the unit vector along the y axis

Hence:

$$r = \sqrt{[a(\cos E - e)]^2 + [a\sqrt{1-e^2} \sin E]^2} = a(1 - e \cos E) \quad (2.27)$$

and

$$\theta = 2 \arctan \left[\sqrt{\frac{1+e}{1-e}} \tan \frac{E}{2} \right] \quad (2.28)$$

2.3.5 Steps of Determining the Keplerian GPS Satellite Position

Kepler's laws can be applied to the GPS satellite orbit with the assumption that only geocentric force is driving the satellite. The steps of determining the satellite orbit can be summarized as follows:

- The period P of the GPS satellite is substituted by 11 hours and 58 minutes in equation (2.23) to determine the mean angular velocity.
- Using equation (2.24), the mean anomaly M is calculated at the given epoch.
- Solving equation (2.25) using numerical computation algorithm, the eccentric anomaly E can be computed.
- Having determined eccentric anomaly E , the satellite geocentric distance r is then calculated from equation (2.27).
- Finally, the true anomaly θ is computed from equation (2.28).

2.3.6 Orbit Representation in Equatorial System

The previous orbit representation is based on the satellite's orbital plane. In practice a transformation from the two-dimensional orbital plane to the three-dimensional equatorial system is required. As shown in Figure 2.4 the transformation involves rotating the orbital plane through the argument of perigee ω , the angle of inclination i and the right ascension of ascending node Ω . Those rotations could be expressed in terms of a transformation matrix $R(\omega, i, \Omega)$ which relates the geocentric vectors $\underline{r}$ and $\underline{\rho}$ in the orbital and equatorial systems respectively according to the following equation:

$$\underline{\rho} = R(w, i, \Omega) \underline{r} \quad (2.29)$$

where:

$$R(w, i, \Omega) = \begin{bmatrix} \cos \Omega \cos \omega - \sin \Omega \sin \omega \cos i & -\cos \Omega \sin \omega - \sin \Omega \cos \omega \cos i & \sin \Omega \sin i \\ \sin \Omega \cos \omega + \cos \Omega \sin \omega \cos i & -\sin \Omega \sin \omega + \cos \Omega \cos \omega \cos i & -\cos \Omega \sin i \\ \sin \omega \sin i & \cos \omega \sin i & \cos i \end{bmatrix} \quad (2.30)$$

Figure 2.4 Orbit representation in ECEF reference frame

Substitute with $\underline{r}$ from equation (2.6) in equation (2.30), the three Cartesian coordinates x_e , y_e and z_e in the equatorial system can be expressed as:

$$\begin{aligned} x_e &= a(\cos \Omega \cos \omega - \sin \Omega \sin \omega \cos i)(\cos E - e) - a\sqrt{1-e^2}(\cos \Omega \sin \omega + \sin \Omega \cos \omega \cos i) \sin E \\ y_e &= a(\sin \Omega \cos \omega + \cos \Omega \sin \omega \cos i)(\cos E - e) + a\sqrt{1-e^2}(\sin \Omega \sin \omega - \cos \Omega \cos \omega \cos i) \sin E \\ z_e &= a \sin \omega \sin i (\cos E - e) + a \cos \omega \sin i \sqrt{1-e^2} \sin E \end{aligned} \quad (2.31)$$

In order to fully define the satellite Keplerian orbit, six parameters need to be introduced. These parameters are:

t_0	initial epoch or perigee epoch
a	semi major axis
e	eccentricity
ω	argument of perigee
i	inclination of orbital plane to the equatorial plane
Ω	right ascension of ascending node

It is worth mentioning that the Keplerian orbit is not a proper representation of the satellite orbit since other sources of disturbing accelerations are not considered. Figure 2.5 illustrates the Keplerian ephemeris error during a two-hour period for the three Cartesian coordinates x , y and z as compared to the final precise ephemeris. As shown in Figure 2.5, the negligence of other perturbing forces seriously affects the precession of the satellite orbit. Several corrections need to be made for the six Keplerian parameters, as will be discussed later in this chapter, in order to improve the accuracy of the GPS satellite orbit. To have an idea of how effective the impact of orbital errors on the station coordinates is; one may consider the following formula, which was derived by (Dach *et al*, 2007) to estimate the positioning error Δx resulting from an orbital error Δx^s :

$$\Delta x(m) \approx \frac{D(km)}{25000(km)} \Delta x^s(m) \quad (2.32)$$

where D is baseline length in km .

If we take, for example, a baseline length of 50 km and a Keplerian error of 500 m , the resulting baseline error will be 1 m .

As for precise point positioning, the coordinate solution with precise ephemeris was found to give better results than that produced when broadcast ephemeris is used (El-Rabbany, 2006).

Figure 2.5 Keplerian ephemeris error compared to final ephemeris

2.3.7 Differential Relations

The differential relation of the satellite orbit positions with respect to the Keplerian parameters is a very important factor during the course of orbit improvement.

Figure 2.6 Differential relations

In Figure 2.6, the receiver range measurements at initial and subsequent epochs are denoted by P_0 and $P_0 + \delta P$, respectively. Receiver measurements P_0 could be expressed in terms of the satellite and position vectors $\underline{\rho}^S$ and $\underline{\rho}_R$ as:

$$P_0 \approx \|\underline{\rho}^S - \underline{\rho}_R\| \quad (2.33)$$

which is the magnitude of the resultant vector $\underline{\rho}^S - \underline{\rho}_R$

Similarly, the differential increment δP can be expressed in terms of the following dot product:

$$\delta P = \frac{\underline{\rho}^s - \underline{\rho}_R}{\|\underline{\rho}^s - \underline{\rho}_R\|} \bullet \delta \underline{\rho}^s \quad (2.34)$$

The orbit improvement involves processing receiver measurements (RINEX observations) from a group of globally distributed stations where equations (2.33) and (2.34) are used. The differential element $\delta \underline{\rho}^s$ entails the implication of partial derivatives of satellite position with respect to the orbital parameters. Therefore, the relation between satellite positions and partial derivatives of orbital parameters should be clearly defined.

The differential element $\delta \underline{\rho}^s$ can be expressed in terms of the differential elements of the orbital parameters and their partial derivatives by differentiating equation (2.29):

$$\delta \underline{\rho}^s = R \frac{\partial r}{\partial a} \delta a + R \frac{\partial r}{\partial e} \delta e + R \frac{\partial r}{\partial E} \delta E + \underline{r} \frac{\partial R}{\partial \omega} \delta \omega + \underline{r} \frac{\partial R}{\partial i} \delta i + \underline{r} \frac{\partial R}{\partial \Omega} \delta \Omega \quad (2.35)$$

The partial derivatives of R with respect to ω , i and Ω can be readily obtained from equation (2.30). So, now, we need to find only the partial derivative of $\underline{r}$ with respect to a , e and E .

Taking partial derivatives of equation (2.26) with respect to a , e and E , it can be shown that:

$$\frac{\partial \underline{r}}{\partial a} = \left(\cos E - e + \frac{3a}{2r} M \sin E \right) \hat{i} + \left(\sqrt{1-e^2} \left(\sin E - \frac{3a}{2r} M \cos E \right) \right) \hat{j} \quad (2.36)$$

$$\frac{\partial \underline{r}}{\partial e} = \left(-a \left(1 + \frac{a}{r} \sin^2 E \right) \right) \hat{i} + \left(\frac{a^2 \sin E}{r \sqrt{1-e^2}} (\cos E - e) \right) \hat{j} \quad (2.37)$$

$$\frac{\partial \underline{r}}{\partial E} = (-a \sin E) \hat{i} + (a \sqrt{1-e^2} \cos E) \hat{j} \quad (2.38)$$

2.3.8 Perturbed Satellite Orbit

The Keplerian orbit considers only the geocentric accelerations due to the Earth's gravitational force. For accurate orbit determination, other accelerations should be taken into account. These accelerations are attributed to various sources of disturbance, such as, but not limited to, the Earth's oblateness, solar and lunar effects and tidal effects. The disturbing forces are discussed in depth in Chapter (3).

If additional components of acceleration are considered, equation (2.2) would be written as:

$$\ddot{\underline{r}} = -\frac{\mu}{r^2} \hat{\underline{r}} + d\underline{\ddot{r}} \quad (2.39)$$

The actual satellite orbit produced by solving equation (2.39) is no longer an elliptical orbit. However, to simplify orbital analysis, the satellite orbit at every epoch can be considered a part of a Keplerian ellipse, known as an osculating ellipse, which is defined by six Keplerian parameters (Figure 2.7). The six Keplerian parameters, thus defined, are not constant but are themselves time dependent. Since the orbital parameters are time dependent, they can easily be determined if the rate of change for each parameter with respect to time is known. Let p_{0i} and $p_i(t)$ denote to the i^{th} orbital parameter at epochs t_0 and t respectively. Assume the rate of change of $p(t)$ is $\dot{p}_i$. Then $p_i(t)$ can be computed as follows:

$$p_i(t) = p_{0i} + \dot{p}_i(t - t_0) \quad (2.40)$$

The second term on the left hand side of equation (2.21) is known as the parameter's correction. In addition to parameters corrections, satellite and receiver clock corrections should also be taken into account to produce an accurate satellite orbit.

One should note that the rate of change $\dot{p}_i$ is itself time dependent. However, it can be kept constant for short orbital arcs. For long arcs and precise orbit determination, equation (2.40) is modified as:

$$p_i(t) = p_{0i} + \int_{t_0}^t \dot{p}_i(t) dt \quad (2.41)$$

Numerical integration algorithms are commonly used to compute the above integrals.

Figure 2.7 Osculating ellipses

2.4 Broadcast ephemeris

The satellite orbital parameters are transmitted as part of the broadcast navigation message. These are sixteen parameters which include the six Keplerian orbital parameters along with their rates of change and correction coefficients. The orbital parameters are periodically updated (Leick, 2004). Based on these orbital parameters, the broadcast ephemeris is calculated. The accuracy of the broadcast ephemeris degrades with time until a fresh set of parameters is transmitted. Table 2.1 shows the distribution of sixteen orbital parameters within the record of navigation message.

Table 2.1 Distribution of orbital parameters within the navigation message

PRN / EPOCH / SV CLK	PRN# y m d h s	clock bias (a_{f0})	clock drift (a_{f1})	clock drift rate (a_{f2})
line 1		C_{rs}	Δn	M_o
line 2	C_{uc}	e	C_{us}	$\sqrt{a}$
line 3	t_o	C_{ic}	Ω_o	C_{is}
line 4	i_o	C_{rc}	ω	$\mathcal{E}$
line 5	$\dot{I}$			

The sixteen orbital parameters can be described as follows:

t_o initial epoch (in seconds)

$\sqrt{a}$ square root of semi-major axis in $\sqrt{meter}$

e eccentricity

i_0	initial angle of inclination
ω	argument of perigee
Ω_0	initial right ascension of ascending node
M_0	initial mean anomaly
Δn	correction of mean angular velocity
$\dot{i}$	rate of inclination angle
$\dot{\omega}$	rate of argument of longitude
C_{rs}, C_{rc}	correction coefficients of geocentric distance
C_{is}, C_{ic}	correction coefficients of angle of inclination
C_{us}, C_{uc}	correction coefficients of argument of latitude

In addition to the sixteen parameters, the satellite clock correction coefficients a_{f0} , a_{f1} and a_{f2} are also provided. Changes in ephemeris parameters and clock corrections can be detected by checking the issue of data ephemeris (IODE) and the issue of data clock (IODC). Apart from orbital parameters, the navigation message contains other useful information, such as the signal transmission time and a satellite health indicator, which is set to 0 if the satellite is unusable (El-Rabbany, 2006). Figure 2.8 depicts the variations of Keplerian parameters within a one-week period of time from July 2nd to July 9th, 2006.

Figure 2.8 Variations of Keplerian parameters within one-week period (July 2nd to July 9th, 2006)

The broadcast ephemeris can be calculated using equations (2.23) to (2.31) along with the above mentioned corrections. The accuracy of the broadcast ephemeris is around

160 cm (IGS, 2007). Figure 2.9 compares the accuracy of the broadcast ephemeris to that of the IGS final ephemeris for the first two hours of July 2nd, 2006.

Figure 2.9 Broadcast ephemeris error compared to final ephemeris for the first two hours of July 2nd, 2006

2.5 Precise Ephemeris

Many GPS applications, whether real-time or post-mission, require a GPS precise orbit that is more accurate than the broadcast ephemeris. Since 1985, the US National Geodetic Survey (NGS) has been producing precise ephemeris in an ASCII format known as Standard Product 1 (SP1) and Standard Product 2 (SP2), (Remondi, 1985). The SP1 files contain satellites' positions and velocities at equal intervals of 15 minutes while SP2 files contain satellite positions only to compact file size. The satellite velocities need no longer be included since they can be calculated by differentiating the satellite's position vector with respect to time. The SP3 format was

proposed in 1989 and adopted in 1991 (Remondi, 1991). The SP3 format contains additional information, such as GPS week and seconds associated with first epoch and satellite clock corrections, which are intended to serve single-receiver mode applications (Spofford and Remondi, 1994). In 1998, SP3b was introduced to include GLONASS ephemeris in conjunction with GPS ephemeris. More information was added to the recent SP3c format, namely the standard deviation of each Cartesian component and correlation coefficients between satellites' coordinates and clock corrections (Hilla, 2002). Currently IGS provides three types of precise ephemeris to GPS users with different levels of accuracy. These are the final, rapid and ultra-rapid ephemeris. At present, eight analysis centers (summarized in Table 2.2) submit ephemeris products to the International GNSS Service (IGS). Those centres are listed on the header of the IGS precise ephemeris SP3 file (see Figure 2.10). The ephemeris products are combined on a weight basis to produce the IGS precise product. The most precise ephemeris product is the IGS final ephemeris, which contains a 24-hour arc with 1 cm precision and almost two weeks delay. Similarly, the IGS rapid ephemeris is a 24-hour orbit that has 17 hours latency and less than 5 cm accuracy (IGS, 2007). While final and rapid ephemeris are used for post-processed applications, the ultra-rapid ephemeris is the only product primarily used for real and near real-time applications. Figure 2.10 shows an example of an IGS final orbit product.


```

#cP2007 8 4 0 0 0.00000000 96 ORBIT IGS05 HLM IGS
## 1438 518400.00000000 900.00000000 54316 0.00000000000000
+ 31 G01G02G03G04G05G06G07G08G09G10G11G12G13G14G16G17G18
+ |G19G20G21G22G23G24G25G26G27G28G29G30G31G32 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 3 3 3 3 3 2 2 2 3 3 3 3 3 3 3 3 3
++ 3 3 3 3 3 2 2 3 2 3 5 3 3 5 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
%e G cc GPS eee eeee eeee eeee eeee eeee eeee eeee
%e cc cc eee eee eeee eeee eeee eeee eeee eeee eeee
%f 1.2500000 1.025000000 0.00000000000 0.000000000000000
%f 0.0000000 0.000000000 0.00000000000 0.000000000000000
%i 0 0 0 0 0 0 0 0 0 0 0 0
%i 0 0 0 0 0 0 0 0 0 0 0 0
/* FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
/* cod emr esa gfz jpl mit ngs sio
/* REFERENCED TO IGS TIME (IGST) AND TO WEIGHTED MEAN POLE:
/* PCV:IGS05_1421 OL/AL:FES2004 NONE Y ORB:CMB CLK:CMB
* 2007 8 4 0 0 0.00000000
PG01 19000.284114 12774.683806 -13312.474018 143.879888 10 11 9 176
PG02 -16422.446769 -179.547140 -21020.144606 126.450122 8 11 9 170
PG03 20488.422631 4275.564470 16326.143865 96.240846 11 11 9 163
PG04 -11006.327466 -12254.546723 -20836.607674 3.552720 11 12 9 175
PG05 -13726.471570 15389.729679 -16904.612276 253.512518 13 8 9 189
PG06 2373.427072 21290.374310 -15490.147384 242.891955 11 9 9 169

```

Figure 2.10 IGS final ephemeris of August 4, 2007

Table 2.2 Analysis Centers

ACRONYM	ORGANIZATION	LOCATION	SOFTWARE	PRODUCT
CODE	Center for Orbit Determination in Europe	University of Berne, Switzerland	Bernese 5.0	CODwwwn.EPH, CODwwwn.EPH_R, CODwwwn.EPH_Pi
EMR-NRCan	Natural Resources Canada	Ottawa, Canada	GIPSY/OASIS II Bernese 5.0	emwwwwn0- 6_hh.sp3 emrwwwwn0-6.sp3
ESA/ESOC	European Space Agency/ European Space Operation Centers	Darmstadt, Germany	BAHN, GPSOBS and BATUSI	ESAwwwwd.eph
GFZ	GeoForschungsZentrum	Potsdam, Germany	EPOS.P.V2	GFUwwwwn0-6.sp3 GFZwwwwn0-6.sp3
JPL	Jet Propulsion Laboratory	Pasadena, USA	GIPSY/OASIS II	JPLwwwwn.sp3
MIT	Massachusetts Institute of Technology	Cambridge, USA	GAMIT 10.02, GLOBK 5.08	mitwwwwn.sp3
NOAA/NGS	National Oceanic and Atmospheric Administration/ National Geodetic Survey	Silver Spring, USA	page5 and gpscom	ngrwwwwn.sp3, ngswwwwn.sp3
SIO	Scripps Institute of Oceanography	La Jolla, USA	GAMIT 10.02, GLOBK 5.08	siuwwwwn_hh.sp3 sirwwwwn.sp3 siowwwwwn.sp3

IGS ultra-rapid product is a 48-hour arc with two different levels of accuracy. The first 24-hour arc represents the observed orbit, which is determined by processing RINEX observations, with less than 5 cm accuracy and the second 24-hour arc is the predicted part whose accuracy is around 10 cm. IGS ultra-rapid is made available to GPS user four times a day at 03:00, 09:00, 15:00 and 21:00 UT with a latency of 3 hours for the observed part and 6 hours for predicted part (IGS, 2007). Both IGS rapid and ultra-rapid accuracy can be validated by IGS final ephemeris product which is commonly used as reference. Figures 2.11 and 2.12 illustrate the IGS rapid and ultra-rapid ephemeris accuracy respectively as compared to IGS final ephemeris for three satellites PRN 04 (block IIA), PRN 13 (block IIR) and PRN 17 (block IIR-M) on Feb. 10th, 2007.

Figure 2.11 IGS Rapid versus Final ephemeris

Figure 2.12 Ultra-rapid versus Final ephemeris

3 PERTURBING FORCES

Apart from the Earth's gravitational attraction force, the GPS satellite is subjected to other perturbing forces that affect the ideal satellite Keplerian trajectory. In addition, the orbital plane itself is affected by perturbing forces in the sense that it does not maintain a fixed inclination to the equatorial plane. Besides this, the line of nodes is slightly shifted due to these forces, which varies the right ascension of ascending node and argument of perigee. Perturbing forces can be grouped into two main categories namely gravitational and non-gravitational forces. The gravitational forces include Earth oblateness, tidal effects, solar attraction, lunar attraction and planetary attraction. The non-gravitational forces are solar radiation pressure, air drag and relativistic effects, and Yaw-bias. Even though the greatest perturbing acceleration is less by four orders of magnitude than the Earth's central gravitational acceleration, neglecting these sources may result in serious orbital errors. Unlike the central attraction force, which is independent of the satellite's mass, area and velocity (as can be inferred from equation 2.2), the non-gravitational forces are mostly dependent on such parameters. Table 3.1 lists the relative percentage magnitude of perturbing accelerations (as compared to the geocentric attraction acceleration, which is approximately equal to 0.57 m/s^2) and the orbital error incurred if a perturbation source is neglected (Dach *et al*, 2007).

Table 3.1 Percentage of perturbing acceleration
with respect to the mean central acceleration
(source: Dach *et al.* 2007)

Perturbing source	Percentage acceleration	Orbital error if source is neglected for 1 day
Earth oblateness	0.0085 %	10 km
Lunar attraction	0.0009 %	3 km
Solar attraction	0.0003 %	800 m
Solid Earth tides	1.6949×10^{-7} %	0.3 m
Solar radiation pressure	1.5254×10^{-5} %	200 m
Y-Bias	8.4746×10^{-8} %	1.5 m

3.1 Oblateness of the Earth

The Keplerian orbit representation is based on the assumption that the Earth is ideally spherical and that no force other than the gravitational attraction is present. According to that assumption, the resulting acceleration is dependent only on the satellite geocentric distance, regardless of its arguments of longitude and latitude. In reality, the Earth bulges at the equator and is flattened at the poles, and the equatorial diameter is greater than the polar diameter by 20 km (Motenbruck and Gill, 2001). The effect of that mass distribution generates an acceleration that is dependent on the geocentric distance as well as the arguments of longitude and latitude. Extra mass at the bulge exerts a force that

attracts the satellite as it passes by the equator. The model that computes the Earth potential V at a particular point of spherical coordinates (r, λ, β) is given by the following harmonic expansion (Hofmann *et. al*, 2001):

$$V(r, \lambda, \beta) = \frac{GM}{r} \left[1 + \sum_{n=2}^{\infty} \sum_{m=0}^n \left(\frac{a_E}{r} \right)^n (J_{n,m} \cos m \lambda + K_{n,m} \sin m \lambda) P_n^{(m)}(\sin \beta) \right] \quad (3.1)$$

where:

G	Universal gravitational constant
M	Earth's mass
a_E	Earth semi-major axis
$J_{n,m}$ and $K_{n,m}$	harmonic coefficients
$P_n^{(m)}$	associated Legendre function
(r, λ, β)	satellites spherical coordinates (geocentric distance, longitude, latitude)

The associated Legendre function $P_n^{(m)}(x)$ is the solution of the differential equation:

$$(1-x^2)y'' - 2xy' + \left(n(n+1) - \frac{m^2}{1-x^2} \right) y = 0 \quad (3.2)$$

The first term on the right hand side of equation (3.1), which is equal to $\frac{GM}{r}$, is the part of Earth potential that is responsible for the main geocentric acceleration. To verify that, one may check the corresponding acceleration by taking the gradient of $\frac{GM}{r}$:

$$\left\| \nabla \left(\frac{GM}{r} \right) \right\| = \frac{GM}{r^2} \approx 0.59 \text{ m/s}^2 \quad (3.3)$$

which is equal to the geocentric acceleration of the Keplerian orbit. Hence, we can conclude that the rest of the right-hand side is the disturbing potential. The greatest harmonic coefficient is $J_{2,0}$, which is equal to 0.00108263 , while subsequent coefficients are less than $J_{2,0}$ by three orders of magnitude (Spilker and Parkinson, 1996). Taking the gradient of disturbing potential, the term that represents the Earth oblateness can be approximated as:

$$a_{oblateness} \approx 3GM \left(\frac{a_E}{r^2} \right)^2 J_{2,0} P_2(\sin \beta) \quad (3.4)$$

Provided the satellite latitude cannot be greater than 55° and $P_2(\sin 55^\circ) = 0.5$, the maximum value of $a_{oblateness}$ will be approximately equal to $5 \cdot 10^{-5} \text{ m/s}^2$, which is less by a factor of 10^4 than the geocentric acceleration. Even though $J_{2,0}$ is higher than other terms by a factor of 10^3 , the number of terms that should be taken into account for accurate orbit determination is 70, as recommended by the Earth Gravitational Model 1996 (EGM96). Some GPS software packages, such as Bernese, permit up to 150 terms; however, the default value is only 12 terms. It can also be noticed that the oblateness acceleration as shown in equation (3.4) increases as β approaches its maximum at 55° and then starts decreasing afterwards. This mathematical relation explains why the satellite speeds up as it approaches the ascending node and slows down otherwise.

3.2 Direct Tidal Effect

Celestial bodies, such as sun, moon and planets, exert tidal attraction forces on satellites. Unlike the effect of the bulge, which pulls the satellite to the Earth, the tidal effect results in acceleration pointing towards the direction of the celestial body. It is found that the

lunar and solar attraction forces have the greatest effect on the satellite, and result in accelerations whose values are $5 \cdot 10^{-6} \text{ m/s}^2$ and $2 \cdot 10^{-6} \text{ m/s}^2$ respectively (Hofmann *et al*, 2001). Other planets such as Mars, Venus and Jupiter are also taken into consideration. Tidal attractions create periodic variations in the argument of perigee and the right ascension of ascending node. Suppose that the geocentric vectors of the satellite and celestial body of mass m_b are $\underline{\rho}^s$ and $\underline{\rho}_b$ respectively (see Figure 3.1). According to Newtonian mechanics the tidal acceleration generated by the celestial body is given by

$$Gm_b \frac{\underline{\rho}_b - \underline{\rho}^s}{\|\underline{\rho}_b - \underline{\rho}^s\|^3}.$$

Figure 3.1 Tidal Effect

Provided that the Earth itself is affected by the celestial body, which generates an acceleration $Gm_b \frac{\underline{\rho}_b}{\|\underline{\rho}_b\|^3}$, the actual perturbing acceleration of the satellite with the Earth taken as reference is expressed as:

$$d\ddot{\rho} = Gm_b \left(\frac{\underline{\rho}_b - \underline{\rho}^s}{\|\underline{\rho}_b - \underline{\rho}^s\|^3} - \frac{\underline{\rho}_b}{\|\underline{\rho}_b\|^3} \right) \quad (3.5)$$

The planetary ephemeris, which are essential for tidal computations, are provided by the Jet Propulsion Laboratory (JPL) in the so-called Development Ephemeris (DE) form which covers the period between 1600 and 2170 (Motenbruck and Gill, 2001).

3.3 Indirect Tidal Effect

Due to the tidal attraction, the Earth experiences crustal and ocean deformations. Since these deformations disturb Earth mass distribution, they may cause an indirect effect on the satellite motion. The disturbing potential V of solid earth is given by (Hofmann *et al*, 2001):

$$V = \frac{1}{2} k G m_b \frac{a_E^5}{(\rho^s \rho_b)^3} (3 \cos^2 z - 1) \quad (3.6)$$

where z is the angle subtended by the satellite and celestial body at the Earth's center, and k is a constant.

Ocean tides, on the other hand, are more difficult to model than that of the solid earth. Furthermore, the geocentric vector of the receiver at the observing station is not constant. For these reasons, tidal charts and ocean loading tables, which describe ocean tides distribution as well as the response of the observing stations to water masses, need to be provided. The perturbing acceleration produced by the indirect tidal effect is in the order of 10^{-9} m/s^2 (Hofmann *et al*, 2001).

3.4 Solar Radiation Pressure (SRP)

The solar radiation pressure arises from the fact that radiation consists of photons that hit the satellite body. The photons energy that is absorbed by the satellite surface within one unit of time is known as the solar flux. The solar pressure is given by the solar flux divided by the satellite surface area. Solar radiation pressure has the greatest effect among the non-gravitational forces. The value of the perturbing acceleration produced by the solar radiation pressure is in the order of 10^{-7} m/s^2 which is exceeded only by the accelerations of oblateness and direct tidal effects. The satellite coordinate system designates the Z axis in the direction of the satellite's antenna, which is pointing towards the Earth center. The X axis is located on the orbital plane and is perpendicular to the Z axis. The Y axis is aligned with solar panel axis which is perpendicular to both X and Z and thus normal to the orbital plane. The basic model of solar radiation perturbation consists of two components; namely $d\ddot{\rho}_X$ in the X direction, referred to as ROCK model, and $d\ddot{\rho}_Y$ in the Y direction, referred to as Y -bias.

3.4.1 ROCK Model:

The first model for the solar radiation pressure for GPS satellites is ROCK4 for Block I satellite and ROCK42 for Block II satellite, which was developed by Rockwell International, the spacecraft company that built Block I and Block II (Dach *et al*, 2007). The ROCK forces consist of two components. The first and main component points to the negative X direction while the second component is on the Y axis. The first component $d\ddot{\rho}_X$ is given by:

$$d\ddot{\underline{\rho}}_x = \gamma K \rho_s^2 \frac{\underline{\rho}^s - \underline{\rho}_s}{\|\underline{\rho}^s - \underline{\rho}_s\|^3} \quad (3.7)$$

where

$\underline{\rho}_s$ is the geocentric vector of the sun

K is a constant that depends on satellites reflectivity and area-to-mass ratio

γ is a positive number between 0 and 1 known as the eclipse factor

The eclipse factor determines whether or not the satellite is in the sunlight. A zero value of the eclipse factor indicates that the satellite is in the Earth's shadow. When the satellite is in the sunlight, the eclipse factor is equal to 1. A value between 0 and 1 indicates that the satellite is passing through the penumbra region.

Accurate modeling of the solar radiation pressures is encountered by several difficulties. One of these difficulties is the complex satellite structure which results in an inaccurate value of K . Another problem is the different satellite surface properties whose reflectivities cannot be represented by only one factor. Additionally, it is hard to assign an accurate value to the eclipse factor as the satellite passes through Earth shadow. The solar panels are oriented towards the Sun by on-board sensors known as attitude control systems (ACS), which are stimulated by sunlight. When the satellite goes into the Earth's shadow, the ACS fail to properly align the solar panels, which results in serious errors. The magnitude of $d\ddot{\underline{\rho}}_x$ is in the order of 10^{-7} m/s^2 .

3.4.2 Yaw-bias:

The ROCK model assumes that the radiation pressure is confined to the X axis only. However, since the satellite's solar panels are not always perpendicular to the X axis,

another component in the y direction is generated. The second component $d\ddot{\rho}_y$ is known as Yaw-bias or Y-bias and is less than $d\ddot{\rho}_x$ by two orders of magnitude. It is found that the magnitude of the Y-bias remains constant for several weeks and therefore it could be taken as an unknown that has to be estimated rather than modeled (Hofmann *et. al*, 2001).

3.4.3 Extended CODE Orbit Model (ECOM):

The ROCK model is insufficient for high precision orbit determination. The shortcoming of the ROCK and Y-bias could be completed by adding additional parameters to the list of unknowns. However, in practice, adding too many parameters may weaken the least square solution. As such, a robust model for solar radiation pressure is necessary for orbit determination. The Extended CODE Orbit Model is a complementary model that is intended to complete the shortcomings of the ROCK model and Y-bias. The ECOM uses a coordinate system that is different from the one adopted by the ROCK model. The X axis coincides with the direction of the satellite-sun vector. The Y axis is the same as that of the ROCK model, which is perpendicular to the orbital plane. The Z axis completes the right-handed system. It is important to note that the X and Z axes of the ROCK model are not aligned to those of ECOM. To avoid confusion, a letter X_1 is used with ECOM to differentiate its X axis from the ROCK X axis. As for the Z axes, there is no need to differentiate between them since ROCK model does not have any component along its Z axis. The total radiation pressure acceleration $\underline{a}_{ECOM}$ is given by (Beutler *et al.*, 1994):

$$\underline{a}_{ECOM} = \underline{a}_{ROCK} + A_x(u)\underline{e}_{X1} + A_y(u)\underline{e}_y + A_z(u)\underline{e}_z \quad (3.8)$$

where

$\underline{a}_{ROCK}$ is the ROCK component

$\underline{e}_x, \underline{e}_y$ and $\underline{e}_z$ are unit vectors in X1, Y and Z directions

u is the argument of latitude

A_x, A_y and A_z are scalar functions of u which are expressed as follows:

$$\begin{aligned} A_x(u) &= A_{x0} + A_{xcu}(u)\cos u + A_{xsu}(u)\sin u \\ A_y(u) &= A_{y0} + A_{ycu}(u)\cos u + A_{ysu}(u)\sin u \\ A_z(u) &= A_{z0} + A_{zcu}(u)\cos u + A_{zsu}(u)\sin u \end{aligned} \quad (3.9)$$

where $A_{x0}, A_{xcu}, A_{xsu}, A_{y0}, A_{ycu}, A_{ysu}, A_{z0}, A_{zcu}$ and A_{zsu} are the 9 parameters of the ECOM model. Provided that variable u is time-dependent, one can conclude that $\underline{a}_{ROCK}$ is implicitly time-varying too. The reason for implementing sine and cosine terms in $\underline{a}_{ROCK}$ is to reflect the periodic nature of the radiation pressure perturbation. Although ECOM is designed specifically to compute the radiation pressure perturbation, it does however, take into consideration the unmodeled perturbations of any other source.

3.5 Earth's Albedo:

The solar radiation that is incident upon the Earth's surface is partially reflected back to space that is known as the Earth's albedo. The Earth's albedo may cause an indirect radiation pressure on the GPS satellite, thus, adding a new acceleration to the list of perturbations. There are two components of albedo: shortwave optical radiation and long wave infrared radiations. In either case, the perturbation decreases with increasing height. For GPS satellites, the magnitude of the albedo perturbation is found to be minimal and can be neglected.

3.6 Air Drag

The drag between the space vehicle and atmospheric particles results in a perturbing force. Air drag is a velocity-dependent force that resists the satellite's motion and acts on a direction opposite to the velocity vector. Since the density of atmospheric particles decreases exponentially as the height increases, it has a negligible effect on GPS satellites.

3.7 Relativistic Effect

During the period between 1907 and 1915, Albert Einstein developed the theory of general relativity which is translated into a set of ten equations known as Einstein Field Equations (Hobson *et al*, 2006). According to this theory, planetary orbits behave in such a way that can not be explained by Newtonian mechanics. Before relativity theory, such phenomenon used to be remedied empirically. In 1915 Einstein developed the model that computes the relativistic effect on Mercury orbit (Hobson, 2006). The relativistic effect on the GPS satellite is mainly attributed to the Earth's gravitational field, which produces a perturbing acceleration a_{rel} , which is approximated by (Beutler, 2005):

$$\underline{a}_{rel} = \frac{GM}{c^2 \rho^3} \left[\left(4 \frac{GM}{\rho} - \dot{\rho}^2 \right) \underline{\rho} + 4(\underline{\rho} \bullet \dot{\underline{\rho}}) \dot{\underline{\rho}} \right] \quad (3.10)$$

The magnitude of that perturbation is in the order of $3 \cdot 10^{-10} \text{ m/s}^2$, and, therefore, can be neglected.

It may be of interest to note that the effect of theory of relativity on GPS orbit should not be confused with its effect on GPS receiver measurements. As a consequence of the latter, a moving clock runs slower than a stationary clock as seen by a stationary

observer. This is known as time dilation which delays the satellite clock by $7 \mu s$. On the contrary, the red shift effect which is another consequence of relativity makes the clock run faster in a weaker gravitational field than it does in a stronger gravitational field. As for the GPS satellite, the clock is $45 \mu s$ ahead. The overall outcome of both effects makes the satellite clock run $38 \mu s$ faster (Nelson, 2002).

4 ARTIFICIAL NEURAL NETWORKS

The main focus of this chapter is to discuss the concept of Artificial Neural Networks with a highlight on how they are structured and developed. Various types of neural network models are described along with their theory of operations. Different training methodologies, learning algorithms and minimization techniques are explained.

4.1 What is the Artificial Neural Network (ANN)?

The Artificial Neural Network (ANN) is an application of the Artificial Intelligence, a renowned branch of science that was developed to emulate human activities such as perception and thinking. ANN is an attempt to approach and simulate the biological brain potentials of learning, analysis, deduction and recognition. The idea of ANN was first brought into existence by Walter McCulloch, a neurobiologist, and Warren Pitts, a mathematician, who wrote the first paper on ANN in 1943 at the University of Chicago (Haykin, 1999). The term Artificial Neural Network was introduced by John McCarthy at the Massachusetts Institute of Technology (MIT) in 1956 (Huang *et al.*, 2004). The ANN acquires knowledge about the data through network training and then applies this knowledge for subsequent purposes of analysis and conclusion. In contrast to digital computers where the data is saved and retrieved for further processing, ANN stores knowledge that can be used for future purposes of modelling and reproduction.

There are several factors that gave ANN a wide popularity in many applications. Those factors include its capability of detecting complicated nonlinearity underlying any data and its ability to generalize and respond to a newly introduced input. Additionally, the

ANN is characterized by its adaptivity to any changes in the surrounding environments (Haykin, 1999). The ANN does not presume any statistical assumption such as normal or linear data distribution. For these reasons, ANN spans a wide spectrum of applications, including robotics, geomatics, aerospace, telecommunication, machine control, and marketing.

4.2 Neural Network Models

The neural network model is defined by its structure and characteristics. A typical network model is designed in layered structure. The three basic units of a neural network are an input, hidden and output layers, as shown in Figure 4.1.

Figure 4.1 The basic units of ANN

It has to be noted that there is no single network structure that can be used in every type of applications. Several network models have been developed over the years to perform

certain tasks in different fields. The hidden layers consist of interconnected nodes known as the *neuron*. The name neuron comes from the analogy with biological neuron as each of them is involved in massive interconnections.

4.2.1 McCulloch-Pitts Model:

The McCulloch-Pitts model is the first network model, which was developed in 1943 by Warren McCulloch and Walter Pitts (Haykin, 1999). This model does not include any learning feature but it has been used as a prototype for later models. It consists of a single summing node and a transfer function. The input data ($i_1, i_2, i_3, \dots, i_n$) is multiplied by a set of weights ($w_1, w_2, w_3, \dots, w_n$) before passing to the node, which adds up the resulted multiplication in addition to a bias (b). The output of the node is modified by a transfer (or activation) function to create nonlinearity and limit output value from going too high or too low, and thus preventing premature network saturation. The model can be represented mathematically by the following equations:

$$v = \sum_{k=1}^n w_k i_k + b \quad (4.1)$$

$$y = f(v) \quad (4.2)$$

Figure 4.2 Basic Structure of a Neuron

There are several types of the activation functions such as:

1. Unipolar Binary Function (Figure 4.3)

$$f(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \quad (4.3)$$

Figure 4.3 Unipolar Binary Function

2. Bipolar Binary Function (Figure 4.4)

$$f(v) = \begin{cases} 1 & v \geq 0 \\ -1 & v < 0 \end{cases} \quad (4.4)$$

Figure 4.4 Bipolar Binary Function

3. Unipolar Sigmoid (Logsigmoid) Function (Figure 4.5)

$$f(v) = \frac{1}{1 + e^{-\lambda v}} \quad (4.5)$$

Figure 4.5 Unipolar Sigmoid Function

4. Bipolar Sigmoid (Tansigmoid) Function (Figure 4.6)

$$f(v) = \frac{1 - e^{-v}}{1 + e^{-v}} \quad (4.6)$$

Figure 4.6 Bipolar Sigmoid Function

5. Unipolar Piecewise Linear Function (Figure 4.7)

$$f(v) = \begin{cases} 1 & v \geq 0.5 \\ v + 0.5 & -0.5 < v < 0.5 \\ 0 & v \leq -0.5 \end{cases} \quad (4.7)$$

Figure 4.7 Unipolar Piecewise Linear Function

6. Bipolar Piecewise Linear Function (Figure 4.8)

$$f(v) = \begin{cases} 1 & v \geq 1 \\ v & -1 < v < 1 \\ 0 & v \leq -1 \end{cases} \quad (4.8)$$

Figure 4.8 Bipolar Piecewise Linear Function

7. Radial Basis Function (Figure 4.9)

$$f(v) = e^{-v^2} \quad (4.9)$$

Figure 4.9 Radial Basis Function

4.2.2 The Perceptron:

This model was designed by Frank Rosenblatt in 1958 based on McCulloch-Pitts model with the added feature of training (Huang *et al*, 2004). Both models consist of a single neuron (Figure 4.2) but the Perceptron model is equipped with adaptable synaptic weights. During the training process, those weights are adjusted such that the output y converges to a desired target d (Rosenblatt, 1958). The activation function adopted by Rosenblatt is the Unipolar Binary Function. In each iteration, the synaptic weights w_i are changed by an amount Δw_i as follows:

$$\Delta w_k = \mu(d - y) i_k \quad (4.9)$$

where

- μ is known as learning rate
- y is the output vector
- d is the target vector
- i_k is the k^{th} input instance
- w_k is the weight associated with i_k

4.2.3 ADALINE and MADALINE Models:

The Adaptive Linear Neuron (ADALINE) was developed by Widrow and his colleagues (Widrow *et al*, 1959). In this model, weights are adjusted by the Widrow-Hoff learning algorithm (Widrow and Hoff, 1960), which is based on minimizing the mean squared error.

Figure 4.10 ADALINE model structure

The Multiple ADALINE (MADALLINE) model is composed of many ADALINEs connected in parallel, as shown in Figure 4.11.

Figure 4.11 MADALINE model structure

4.2.4 Feed-forward Networks:

These are networks with more than one neuron and involve no feedback paths. There are two different types of Feed-forward networks, namely Single-layer Feed-forward Networks (Figure 4.12) and Multi-layer Feed-forward Networks (Figure 4.13).

Figure 4.12 Single-layer Feed-forward Network

In the Single-layer Feed-forward network, there is one input layer and one output layer with no connections back to the input. All the inputs are connected to all neurons but no interconnections exist between the nodes.

Figure 4.13 Multiple-layer Feed-forward Network

Unlike the Single-layer Feed-forward which does not contain hidden layers, the Multiple-layer Feed-forward Network has multiple hidden layers that are neither part of the input nor the output layers. The output of each hidden layer is used as weights to the next layer. The Multi-layer Feed-forward networks are also known as Multi-layer Perceptron (MLP). In 1985, the error Back-propagation Algorithm was developed by Rumelhart and McClelland (Haykin, 1999). Since then, it has been associated with the MLP. The back-propagation is sometimes referred to as Generalized Delta Rule.

4.2.5 Recurrent Networks:

The previous Feed-Forward network has no interconnections between neurons of the same and previous layers. In addition, it does not have memory, in the sense that no feedback is implemented. The Recurrent or Feed-forward Back-propagation remedies these drawbacks by providing feedback connections (Figure 4.14). The output to each neuron depends, not only on its input, but also on the feedback connections.

Figure 4.14 Recurrent Networks Structure

Two types of recurrent networks are commonly used, namely, the Hopfield and Elman Neural Networks. In the Hopfield model, the output of each neuron is delayed and fed back to the other neurons. The number of feedback loops is equal to the number of neurons. No hidden layers are available in the Hopfield Model. The Hopfield model is

used as an associative memory to store certain equilibrium state. The extension of the Hopfield model is the Boltzmann learning machine. On the other hand, The Elman model is a two-layer network with feedback from the first layer's output to the first layer's input. The Elman model has a *tansig* transfer function in its hidden layer and a linear transfer function in its output layer. This combination enables the Elman model to approximate any function with a reasonable accuracy. More hidden neurons can be added as the function increases in complexity.

4.2.6 Radial Basis Function Networks:

The Radial Basis Function (RBF) networks have an advantage of simplicity over MLP networks. The output layer of an RBF network is a linear combination of the hidden layers signals (Figure 4.15).

Figure 4.15 Structure of Radial Bias Function Networks

The activation function of this type of network is the Radial Bias Function (see Figure 4.9). It should be noted that the weights vector is not multiplied by the input vector, as in the case of MLP, but, instead, it is subtracted from the weights vector. The resulted subtraction is modified by the activation function and then multiplied by the bias vector before being added up at the output. The RBF networks are usually used for problems involving approximations. There are two variants of Radial Basis Networks: Generalized Regression Neural Networks (GRNN) and Probabilistic Neural Networks (PNN).

4.2.7 Self-organizing Neural Networks:

The previous network models use the so-called supervised learning process for network training. The supervised learning algorithm requires external input (or teacher) to achieve

optimum training. Self-organizing models use the unsupervised learning process, which does not involve an external effect. The Self-organizing network consists of only one layer with interconnected neurons. There are two types of Self-organizing networks, invented by Kohonen in 1984 (Huang *et al*, 2004): Linear Vector Quantization (LVQ) and Self-organizing Map (SOM).

(a) Linear Vector Quantization (LVQ)

LVQ is used for character recognition, data encoding and data compression. The purpose of LVQ training is to classify the input data into certain patterns according to their geometric relations such as the distance between input points.

(b) Self-Organizing Map (SOM)

SOM is used for image recognition and feature extraction. Instead of using simple geometric relations, as in LVQ, SOM employs a form of spatial arrangement of the output nodes, where the spatial coordinates are significant. The neurons are configured in a two-dimensional matrix and a competitive algorithm is introduced to pick the most possible neuron, which is known as winning neuron. Then, additional training is carried out so that the neuron is moved and brought closer to its classified group.

4.2.8 Modular and Hierarchical Neural Networks:

While local mapping functions networks, such as RBF, have the advantage of fast learning, they are, unfortunately, memory consumptive. On the other hand, global mapping functions, such as MLP, perform better with a limited amount of memory and could be generalized more extensively than RBF. However, MLP suffer from having a

relatively slow learning process. Modular Neural Networks combine the advantages of both RBF and MLP. Modular Neural Networks consist of several smaller modules. The input data is decomposed into smaller segments by the Modular Neural Networks so that each segment can be trained independently by a separate module. The outputs of these modules are combined by an integrating unit to produce the final output (Figure 4.16). As a result, the modular structure has faster learning, better data representation, and a more compact network.

Figure 4.16 Modular Neural Network

The Gating Network consists of k neurons, with each neuron assigned to a particular module. The Gating Network modifies the output of each module by a function g_j before passing it to the final output. This modular structure is known as an Expert Network. Another complex structure is the Hierarchical Mixture of Experts (HME), which is an extension of modular networks, as illustrated in Figure 4.17. The input space is decomposed into nested subspaces, with several Gating Networks at each subspace.

Figure 4.17 Hierarchical Mixtures of Experts (HME)

4.3 Learning Methods

The neural networks need to be trained thoroughly in order to acquire adequate knowledge and collect the required information about the data being modeled and the environment in which the network exists. Several training techniques are adopted to get

the neural network viable for various applications. A few types of learning methods are discussed in the following context.

4.3.1 Supervised Learning (Error Correction Learning):

Supervised learning (also known as learning with a teacher) aims to give the network with sufficient knowledge by presenting a training set of input and desired output data (commonly known as the target). The training is performed over many iterations by adjusting the network parameters. At each iteration, the actual output is compared to the target, and a decision of whether or not to continue the training is taken. The training is terminated when the actual output is sufficiently close to the target. The closeness of the actual output to the target is measured by the mean-squared error, which has to be minimum.

4.3.2 Unsupervised Learning:

This type of learning does not require a teacher, which means that there is no desired output to compare with. Basically, the unsupervised learning is used for pattern classification problems. During the training process, the network groups the input data and assigns them to certain classes. Members of the same class will have the same properties (e.g. brightness, colour and resolution). If the input sample does not match any of existing classes, a new class is created to which it will be assigned.

4.3.3 Reinforced Learning:

In this type of learning, the teacher is present. However, the teacher does not provide a desired response for the network. Instead, the teacher introduces a signal – known as a reinforcement signal – of only "Pass" or "Fail". If a "Fail" signal is received, the network parameters are adjusted and the training is repeated until a "Pass" signal is provided. During the training process, there is no indication that the learning is optimized; therefore, a criterion must be set to terminate the training at the right time.

4.3.4 Competitive Learning (Winner-Takes-All Learning):

In competitive learning, neurons compete with each other and only the winning neuron is selected. Comparing the output nodes to the target, the node that is closest to the target, according to particular criteria, is selected. Only parameters associated with the winning node are adjusted. In other words, the network responds positively only to a specific group of input data.

4.3.5 Hebbian Learning:

In Hebbian Learning, the weights between network neurons are adjusted according to the activation state of any two neurons. The weight between two neurons will increase if the two neurons are activated simultaneously and will decrease otherwise. For example, if two neurons tend to both be positive or negative at the same time, then the weight between them will be strengthened, while the weights of those that have mutually opposite condition are reduced.

4.3.6 Support Vector Learning:

Support Vector Learning (also known as Support Vector Machine) was developed by Vapnik and his group at AT&T BELL Laboratories (Vapnik, 1995; Vapnik and Cortes 1995). The Support Vector Machine (SVM) can be considered a classification technique more than a learning method. The main idea of SVM is to separate the classes with a hyperplane surface by maximizing the margin among them, thus minimizing the generalization error. SVM is used in cases where the data has to be classified into two different classes, such as road feature extraction, face detection in images and handwritten objects.

4.4 Learning Algorithms

The implementation of the previous learning methods in neural networks involves different mathematical procedures known as learning algorithms. A variety of learning algorithms are adopted by network designers to achieve the optimum training by adjusting the network parameters until they reach a convergent solution.

Let w be the weight vector of the neural network. During training, this weight vector needs to be adjusted regularly, according to the following equation:

$$w^{(k+1)} = w^k + \Delta w^k \quad (4.10)$$

where

k refers to the k^{th} iteration (k is not an exponent)

w^k is the weight at the k^{th} iteration

$w^{(k+1)}$ is the weight at the $(k+1)^{th}$ iteration

Δw^k is the weight change

It is important to note that the k is placed as superscript because the subscript is reserved for neurons indices.

In general, the adjustment weight depends on the input vector $\mathbf{x}$, the output vector $\mathbf{y}$ and the desired output $\mathbf{d}$. Hence, it could be represented as follows:

$$\Delta w^k = c r^k x^k \quad (4.11)$$

where

c is the learning constant

r^k is the learning signal which determines the direction of weight adjustment
which is a function of $\mathbf{y}$ and $\mathbf{d}$.

For multi-neuron networks, equation (4.11) will be represented by:

$$\Delta w_{ij}^k = c r^k x_i^k \quad (4.12)$$

w_{ij} is the weight connecting the i^{th} input sample (i.e. x_i) to the j^{th} neuron

4.4.1 Perceptron Learning Algorithms:

This algorithm trains the network according to the supervised learning method where the teacher provides a desired output as shown in Figure 4.18.

Figure 4.18 Perceptron Learning Algorithm in a single-layered multi-neuron network

For this algorithm, the weight adjustment Δw_{ij}^k is written as:

$$\Delta w_{ij}^k = c(d_j - y_j^k)x_i^k \quad (4.13)$$

According to the perceptron convergence theorem (Huang *et al*, 2004), the weight vector $\mathbf{w}$ converges after a finite number of iterations.

4.4.2 Hebbian Learning Algorithm:

This is an unsupervised-based algorithm that is used in self-organized networks. Since there is not target vector, the weight adjustment Δw_{ij}^k could be expressed as follows:

$$\Delta w_{ij}^k = cy_j^k x_i^k \quad (4.14)$$

Note that the repeated multiplication of x_i may produce an exponential increase in Δw_{ij}^k .

To avoid this situation, equation (4.14) is modified by an additional term as:

$$\Delta w_{ij}^k = c y_j^k x_i^k - \alpha y_j^k w_{ij}^k \quad (4.15)$$

where α is a user-defined constant

4.4.3 Widrow-Hoff Learning Algorithm:

This algorithm minimizes the squared error between the desired and the actual output.

The output vector y_j^k of neuron j at the k^{th} iteration is given by:

$$y_j^k = \sum_{i=1}^n w_{ij}^k x_i^k \quad (4.16)$$

and the error vector e^k at the k^{th} iteration between the desired output vector d_j and actual output vector y_j^k is expressed as:

$$e^k = d_j - y_j^k \quad (4.17)$$

the weight adjustment Δw_{ij}^k will be computed by the following equation:

$$\Delta w_{ij}^k = \mu e^k x_i^k \quad (4.18)$$

μ is a user-defined constant that is known as the learning rate. The parameter μ is a significant that greatly affects the network convergence. It is found that the network convergence will be attained if the value of μ satisfies one of the following conditions:

$$1. \quad 0 < \mu < \frac{2}{\lambda_{\max}} \quad (4.19)$$

where $\lambda_{\max}$ is the largest eigenvalue of the autocorrelation matrix $\rho_{xx}(x_i^k)$

$$2. \quad 0 < \mu < \frac{2}{\text{tr}[\rho_{xx}(x_i^k)]} \quad (4.20)$$

where $\text{tr}[\rho_{xx}(x_i^k)]$ is the trace matrix of $\rho_{xx}(x_i^k)$, which is equal to the sum of its diagonal elements.

4.4.4 Delta Learning Rule:

The Delta Learning Rule is supervised-based algorithm, which is commonly applied to the multi-layered feed-forward back-propagation networks. Consider an intermediate layer in a multi-layered network as shown in Figure 4.19

Figure 4.19 An intermediate layer in a multi-layered network

y_i is the output of the previous layer and the input to the intermediate layer

s_j is the input to the activation function

s_j is related to y_i by the following equation:

$$s_j = \sum_{i=1}^n w_{ij} y_i \quad (4.21)$$

f is the activation function

o_j is the output of the j^{th} neuron at the intermediate layer

Define E , the error function as:

$$E = \frac{1}{2} \sum_{j=1}^m (d_j - o_j)^2 \quad (4.22)$$

the weight adjustment Δw_{ij} is expressed as:

$$\Delta w_{ij} = -\mu \frac{\partial E}{\partial w_{ij}} = -\mu \left(\frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}} \right) \quad (4.23)$$

The purpose of expressing the weight adjustment Δw_{ij} in terms of the gradient of E is to guarantee the weight convergence (i.e. $\Delta w_{ij} \rightarrow 0$) as the error function E is approaching its minimum. Now we need to obtain the relation between the gradient of E and the network parameters.

From Figure 4.19 we can find that

$$\frac{\partial o_j}{\partial s_j} = f'(s_j) \quad (4.24)$$

taking partial derivatives of equations (4.21) and (4.22), we can substitute

$$\frac{\partial E}{\partial o_j} = -(d_j - o_j) \quad (4.25)$$

and

$$\frac{\partial s_j}{\partial w_{ij}} = y_i \quad (4.26)$$

Therefore equation (4.23) becomes

$$\Delta w_{ij} = \mu(d_j - o_j)f'(s_j)y_i \quad (4.27)$$

4.5 Minimization Functions

Learning algorithms requires mathematical routines for efficient minimization of the error function. In numerical computations, many algorithms are used for this purpose, such as the Newton Algorithm, Gauss-Newton Algorithm (GNA), Gradient Descent, and Levenberg-Maquardt Algorithm (LMA).

4.5.1 Newton Algorithm:

The Newton Algorithm is based on the fact that the function is minimized when its first derivative is zero and its second derivative is positive.

Using Taylor series, the function $f(u + \delta u)$ can be approximated as:

$$f(u + \delta u) = f(u) + f'(u)\delta u \quad (4.34)$$

$$f(u + \delta u) = f(u) + f'(u)\delta u \quad (4.34)$$

This approximation is valid for small values of the differential amount δu

Taking the derivative of the left hand side and equating the derivative to zero:

$$0 = f'(u) + f''(u)\delta u \quad (4.35)$$

which means that:

$$\delta u = -\frac{f'(u)}{f''(u)} \quad (4.36)$$

substituting δu by $u^{k+1} - u^k$ equation (4.36) can be rewritten as:

$$u^{k+1} = u^k - \frac{f'(u^k)}{f''(u^k)} \quad (4.37)$$

The k and $k+1$ are not exponents. They are only written as superscript to avoid confusion between the iteration index and the variable index, as will be shown shortly.

An initial guess u^0 is substituted in equation (4.37). Then the procedure continues iteratively until $u^{k+1} = u^k$ or $|u^{k+1} - u^k|$ is less than a predefined limit.

In practice, f is a multi-dimensional function that depends on additional variables, i.e. $f(u_1, u_2, u_3, \dots)$.

In this case equation (4.37) can be generalized by replacing $f'(u)$ by the gradient ∇f

and $\frac{1}{f''(u)}$ by the inverse of the Hessian matrix $H(f)$:

$$u^{k+1} = u^k - [H(f)]^{-1} \nabla f \quad (4.38)$$

The Hessian matrix $H(f)$ for a multi-dimensional function $f(u_1, u_2, \dots, u_n)$ is given by:

$$H(f(u_1, u_2, \dots, u_n)) = \begin{vmatrix} \frac{\partial^2 f}{\partial u_1^2} & \frac{\partial^2 f}{\partial u_1 \partial u_2} & \text{L} & \frac{\partial^2 f}{\partial u_1 \partial u_n} \\ \frac{\partial^2 f}{\partial u_2 \partial u_1} & \frac{\partial^2 f}{\partial u_2^2} & \text{L} & \frac{\partial^2 f}{\partial u_2 \partial u_n} \\ \text{M} & \text{M} & \text{M} & \text{M} \\ \frac{\partial^2 f}{\partial u_n \partial u_1} & \frac{\partial^2 f}{\partial u_n \partial u_2} & \text{L} & \frac{\partial^2 f}{\partial u_n^2} \end{vmatrix} \quad (4.39)$$

4.5.2 Gauss-Newton Algorithm:

This algorithm is an extension of Newton Algorithm, which is used to minimize the squared sum of a number of functions.

$$S(x, p) = \sum_{i=1}^m [f_i(x, p)]^2 \quad (4.40)$$

x and p refer to the input and parameters vector for neural networks respectively; f_i refers to the vector of $(f_1, f_2, \dots, f_m)$.

Applying the Newton algorithm for the function $S(x, p)$ yields:

$$p^{k+1} = p^k - [H(S)]^{-1} \nabla S \quad (4.41)$$

which can be simplified further as follows:

$$p^{k+1} = p^k - \left(J_f^T J_f + \sum_{i=1}^m f_i(x, p) H(f_i) \right)^{-1} J_f^T f_i(x, p^k) \quad (4.42)$$

where J_f is the Jacobian of f which is given by:

$$J_f = \begin{vmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & L & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & L & \frac{\partial f_2}{\partial u_m} \\ M & M & M & M \\ \frac{\partial f_m}{\partial u_1} & \frac{\partial f_m}{\partial u_2} & L & \frac{\partial f_m}{\partial u_m} \end{vmatrix} \quad (4.43)$$

An initial guess vector p^0 is first applied, after which subsequent parameters are computed recurrently until the network is trained.

4.5.3 Gradient Descent:

The Gradient Descent is based on approaching the local minimum of a function by moving in the direction of the negative gradient. For the sake of explanation, let us take an example of one-dimensional function $f(u)$, as shown in Figure 4.20.

Figure 4.20 Minimization of a one-dimensional function using Gradient descent

The algorithm starts by moving from point a to point b according to the following iterative equation:

$$u^{k+1} = u^k - \lambda f'(u^k) \quad (4.44)$$

λ is a small positive number, known as the *damping factor*

Since $f'(u^k)$ is negative and λ is positive, it follows that $u^{k+1} > u^k$ and, consequently, $f(u^{k+1}) < f(u^k)$, that is, moving towards the minimum.

For a multi-dimensional function, the recurrent equation becomes:

$$u^{k+1} = u^k - \lambda \nabla f \quad (4.45)$$

4.5.4 Levenberg-Marquardt Algorithm (LMA):

The Levenberg-Marquardt Algorithm (LMA) interpolates between Gauss-Newton and Gradient Descent algorithms. LMA has an advantage over GNA in finding a solution even if the initial estimates are very far from the final minimum (Gill *et al*, 1978). For MATLAB, this algorithm appears to be the fastest method for training medium-sized networks (up to a few hundred weights); therefore, it is the one that is used for minimizing the global error during the network training in this research.

The recurrent equation of LMA can be expressed as follows:

$$p^{k+1} = p^k - (J_f^T J_f + \lambda I)^{-1} J_f^T f_i(x, p^k) \quad (4.46)$$

where:

λ is the damping factor

I is the identity matrix

J_f is the Jacobian of f

5 ANALYSIS AND MODELING METHODOLOGY

This chapter explains the methods and algorithms used to achieve the objectives of the research. A description of how the input data is handled and processed is overviewed followed by analysis and discussion of the results. Advantages and limitations of different methods are also provided. Lastly, a few suggestions are proposed for the sake of further improvement in future work.

5.1 Assessment of Several Interpolation Methods for Precise GPS Satellite Orbit

In this section, GPS precise ephemeris that are spaced at equal intervals of 15 minutes is interpolated to a higher rate of 5 minutes using four interpolation methods, namely, Lagrange, Newton Divided Difference, Cubic Spline and Trigonometric Methods. A comparison between the results of each method is conducted. A new approach of interpolation using broadcast ephemeris is also described.

5.1.1 Lagrange Interpolation Method

The Lagrange formula is an algebraic expression that can be used to fit a particular data set to a certain polynomial (whose degree is equal to the number of data points) such that it returns the exact value of the function at each data point. Let $f_0, f_1, f_2, \dots, f_n$ be the values of a given data at times $t_0, t_1, t_2, \dots, t_n$, respectively. The approximated value of f , denoted by $p(t)$, at any time t is given by (Spiegel, 1999):

$$p(t) = a_0 f_0 + a_1 f_1 + a_2 f_2 + \dots + a_n f_n = \sum_{i=0}^n a_i f_i \quad (5.1)$$

where

$$a_i = \frac{(t-t_0)(t-t_1)L(t-t_{i-1})(t-t_{i+1})L(t-t_n)}{(t_i-t_0)(t_i-t_1)L(t_i-t_{i-1})(t_i-t_{i+1})L(t_i-t_n)} \quad (5.2)$$

Since a_i coefficient is a function of t , it can also be referred to as $L_i(t)$ which is called the Lagrange operator. Now, substituting t by $t_0, t_1, t_2, \dots, t_n$ in equation (2) we get:

$$a_i = L_i(t) = \begin{cases} 1 & \text{for } t = t_i \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

Going back to equation (1) and substituting again t by $t_0, t_1, t_2, \dots, t_n$, we get:

$$p(t_0) = f_0, p(t_1) = f_1, p(t_2) = f_2, \dots, p(t_n) = f_n \quad (5.4)$$

5.1.2 Newton Divided Difference

The Newton Divided Difference formula is another polynomial expression of a given data set. Similar to Lagrange, Newton Divided Difference returns the exact value of the function at every data point. This formula is termed *divided difference* because its coefficients involve division of differences. The Newton Divided Difference formula is given, for a certain data set (t_i, f_i) , by (Spiegel, 1999):

$$p(t) = a_0 + a_1(t-t_0) + a_2(t-t_0)(t-t_1) + a_3(t-t_0)(t-t_1)(t-t_2) + \dots \\ + a_n(t-t_0)(t-t_1)\dots(t-t_{n-1}) \quad (5.5)$$

where:

$a_0, a_1, a_2, \dots, a_n$ are coefficients; and $p(t)$ is the approximated value at any time t . The coefficients $a_0, a_1, a_2, \dots, a_n$ can be determined recursively by substituting $t_0, t_1, t_2, \dots, t_n$ as follows:

$$a_0 = p(t_0) \quad (5.6)$$

$$a_1 = \frac{p(t_1) - a_0}{(t_1 - t_0)} \quad (5.7)$$

$$a_2 = \frac{p(t_2) - a_0 - a_1(t_2 - t_0)}{(t_2 - t_0)(t_2 - t_1)} \quad (5.8)$$

$$a_n = \frac{p(t_n) - a_0 - a_1(t_n - t_0) - a_2(t_n - t_0)(t_n - t_1) - \dots - a_{n-1}(t_n - t_0)(t_n - t_1)\dots(t_n - t_{n-1})}{(t_n - t_0)(t_n - t_1)(t_n - t_2)\dots(t_n - t_{n-1})} \quad (5.9)$$

5.1.3 Cubic Spline Interpolation:

In this method, the data set is represented with a 3rd degree polynomial piecewisely (i.e. the data is divided into sections and each section is represented by a cubic polynomial). For example if we have a data set (t_i, f_i) , then the Cubic Spline can be defined as follows (Press et al., 2002):

$$S(t) = \left\{ \begin{array}{ll} s_0(t) & t_0 < t < t_1 \\ s_1(t) & t_1 < t < t_2 \\ s_2(t) & t_2 < t < t_3 \\ & \text{M} \\ s_i(t) & t_i < t < t_{i+1} \\ & \text{M} \\ s_{n-1}(t) & t_{n-1} < t < t_n \end{array} \right\} \quad (5.10)$$

where:

$$s_i(t) = a_i t^3 + b_i t^2 + c_i t + d_i \quad (5.11)$$

Each section of data has different coefficients a_i , b_i , c_i and d_i . These four coefficients can be determined as follows:

Substitute $t = t_i$ and $t = t_{i+1}$

$$s_i(t_i) = a_i t_i^3 + b_i t_i^2 + c_i t_i + d_i \quad (5.12)$$

$$s_i(t_{i+1}) = a_i t_{i+1}^3 + b_i t_{i+1}^2 + c_i t_{i+1} + d_i \quad (5.13)$$

Differentiating $s_i(t)$ and substituting $t = t_i$ and $t = t_{i+1}$

$$s'_i(t_i) = 3a_i t_i^2 + 2b_i t_i + c_i \quad (5.14)$$

$$s'_i(t_{i+1}) = 3a_i t_{i+1}^2 + 2b_i t_{i+1} + c_i \quad (5.15)$$

Now using the numerical differentiation formula:

$$s'_i(t_i) = \frac{f_{i+1} - f_i}{t_{i+1} - t_i} \quad (5.16)$$

and

$$s'_i(t_{i+1}) = \frac{f_{i+2} - f_{i+1}}{t_{i+2} - t_{i+1}} . \quad (5.17)$$

Equations (12) to (17) can be solved simultaneously to determine the coefficients a_i , b_i , c_i and d_i of each section.

5.1.4 Trigonometric Interpolation:

The trigonometric formula is a harmonic representation of a given data set. This representation is mainly convenient for the data that has a periodic nature, which is the case for the GPS orbit. The trigonometric formula for a given data set (t_i, f_i) , is given by (Spiegel, 1999):

$$p(t) = \frac{a_0}{2} + \sum_{n=1}^M [a_n \cos(\frac{2\pi n}{T}t) + b_n \sin(\frac{2\pi n}{T}t)] \quad (5.18)$$

where:

$p(t)$ is the approximated value at time t

a_n, b_n are coefficients

T is the period of f

N is the number of collocation points

M is the number of truncated terms (ideally $M = \infty$)

To determine a_n multiply both sides by $\cos(\frac{2\pi m}{T}t)$ and take the integration over one period:

$$\begin{aligned} \int_0^T p(t) \cos(\frac{2\pi m}{T}t) dt &= \int_0^T \frac{a_0}{2} \cos(\frac{2\pi m}{T}t) dt \\ &+ \sum_{n=1}^M \int_0^T a_n \cos(\frac{2\pi n}{T}t) \cos(\frac{2\pi m}{T}t) dt + \sum_{n=1}^M \int_0^T a_n \sin(\frac{2\pi n}{T}t) \cos(\frac{2\pi m}{T}t) dt \end{aligned} \quad (5.19)$$

The first integration on the right hand side is equal to 0. Moreover according to the orthogonality theorem:

$$\int_0^T \sin(\frac{2\pi n}{T}t) \cos(\frac{2\pi m}{T}t) dt = 0 \quad (5.20)$$

$$\int_0^T \cos(\frac{2\pi n}{T}t) \cos(\frac{2\pi m}{T}t) dt = \begin{cases} 0 & m \neq n \\ \frac{T}{2} & m = n \end{cases} \quad (5.21)$$

Making use of equations (19) and (20) we get:

$$a_n = \frac{2}{T} \int_0^T p(t) \cos(\frac{2\pi n}{T}t) dt \quad (5.22)$$

Similarly:

$$b_n = \frac{2}{T} \int_0^T p(t) \sin(\frac{2\pi n}{T}t) dt \quad (5.23)$$

Equations (22) and (23) can be reformulated into the following numerical form:

$$a_n = \frac{2}{N} \sum_{k=1}^N f_k \cos\left(\frac{2\pi k}{T} t_k\right) \quad (5.24)$$

$$b_n = \frac{2}{N} \sum_{k=1}^N f_k \sin\left(\frac{2\pi k}{T} t_k\right) \quad (5.25)$$

5.1.5 Results and Discussion

To assess the performance of the various interpolation methods, a test data set was acquired from NOAA/NGS. The data set consists of the following files:

- a. IN_15MIN.200 (A 24-hour precise orbit file spaced at 15 min in the inertial reference frame);
- b. ECF_15MIN.200 (same as “a” but the orbital coordinate refer to the ECEF reference frame);
- c. IN_5MIN.200 (same as “a” but the data interval is 5 minutes);
- d. ECF_5MIN.200 (same as “b” but the data interval is 5 minutes).

The four files (“a” to “d”) were generated from the IGS sp3 rapid ephemeris (igr11472.sp3) for January 1, 2002. The first two files (i.e., “a” and “b”) were used as input for each interpolation function, while the third and fourth files (i.e., “c” and “d”) were used as sources to which the interpolation results were compared.

Upon examination of the interpolation results, it was found that the following properties are shared by all interpolation methods:

- Having too few collocation points produces an unreliable interpolation output. On the other hand, having relatively many points is expected to improve the result but to some limit.
- The accuracy degrades noticeably near the end points and tends to improve as the interpolator moves towards the center.

(a) Lagrange Interpolation Results

As mentioned previously, the interpolation accuracy enhances as the number of points increases. The accuracy also assumes its maximum at a specific number of terms, after which it continues to degrade and eventually become unstable beyond a certain limit. Trying to investigate the optimum number of terms that gives the best accuracy, we started interpolation by taking the whole number of points for the 24-hour trajectory, which is 96 points spaced at 15 minutes intervals. We found that 96 points are too many to give an acceptable result, as shown in Figure 5.1.

Figure 5.1 Resulting error when taking the whole 24-hour data to build 96 Lagrange interpolating terms.

Taking smaller portions of data with different number of collocation points (8, 9, 10 and 11 points), we found that the best accuracy was achieved when $n = 9$, as shown in Figures 5.2 and 5.3 (Notice the difference in vertical scales of both figures). The accuracy can also be seen to degrade slightly near the boundaries for all figures. To overcome this problem, we segmented the 24-hour orbit into 23 overlapping portions; each has 9 terms, as shown in Figure 5.4

Figure 5.2 Resulting error when taking 11 Lagrange terms

Figure 5.3 Resulting error when taking 9 Lagrange terms.

Figure 5.4 Segmentation scheme: 24-hour data is segmented into 23 overlapping segments of 9 terms each.

The segmentation scheme is designed in such a manner that each segment starts at the midpoint of the previous segment. The last segment, however, starts at point 88 (instead of point 89) up to point 96 in order to complete the desired 9 terms. Otherwise, we will be left with only 8 terms for the last segment, which leads to a relatively lower accuracy. The idea behind having overlapped segments is to avoid producing spikes near the endpoints, as shown in Figures 5.2 and 5.3. Since the accuracy of interpolation assumes its best at the middle, values near the end points will be replaced by those of the mid adjacent segments. A comparison between partitioned and overlapped segments is illustrated in Table 5.1 and Figures 5.5 and 5.6. From the table and figures, we can easily see the accuracy enhancement in the overlapping case.

Table 5.1 Comparison between interpolation errors generated by partitioned vs. overlapping segmentation.

Partitioned vs. Overlapped	MEAN (cm)			Standard Deviation (cm)		
	dx	dy	dz	dx	dy	dz
Partitioned Segments	0.0030	0.0139	0.0051	0.1064	0.1720	0.1366
Overlapped Segments	0.0025	0.0067	0.0037	0.0451	0.0756	0.0405

Figure 5.5 Interpolation error when the GPS orbit is partitioned into smaller segments without overlapping

Figure 5.6 Interpolation error when the GPS orbit is segmented into overlapping segments

Applying Lagrange interpolation using the previous scheme for both inertial and ECEF coordinates, Figures 5.7 and 5.8 show the obtained interpolation error for PRN01 (Notice the difference in vertical scales of both figures)

Figure 5.7 Lagrange Interpolation error for Inertial GPS Orbit.

Figure 5.8 Lagrange Interpolation error for ECEF GPS Orbit.

The interpolation accuracy of inertial orbit is better than that of ECEF, as shown in Figures 5.7 and 5.8. The statistics in Table 5.2 also depict this fact numerically.

Table 5.2 Comparison between Inertial vs. ECEF Lagrange interpolation error.

INERTIAL vs. ECEF	MEAN (cm)			STD (cm)			MAX (cm)		
	dx	dy	dz	dx	dy	dz	dx	dy	dz
INERTIAL	0.0025	0.0067	0.0037	0.0451	0.0756	0.0405	0.4127	0.6374	0.2233
ECEF	0.0016	0.0120	0.0034	0.1548	0.2501	0.0623	1.5216	3.3753	0.4276

(b) Newton Divided Difference Interpolation Results

Using the previously discussed algorithm, which can be applied to any other interpolation function, we found that the Newton interpolation result is completely identical to that of Lagrange for both Inertial and ECEF orbits, as shown in Figures 5.9 and 5.10.

Figure 5.9 Newton Divided Difference Interpolation error for Inertial GPS Orbit.

Figure 5.10 Newton Divided Difference Interpolation error for ECEF GPS orbit

(c) Cubic Spline Interpolation Results

The Cubic Spline interpolation resulted in a very low accuracy as seen in Figure 5.11. Perhaps, the reason is due to the fact that the Cubic Spline is a polynomial of order 3, which is not suitable to represent the periodic nature of the GPS orbit.

Figure 5.11 Cubic Spline Interpolation error for GPS Inertial Orbit.

(d) Trigonometric Interpolation Results

The periodic nature of the GPS orbit in the inertial reference frame makes it reasonable to employ the periodic Trigonometric interpolation. The Trigonometric interpolation method for GPS orbit was adopted – among other methods – by Schenewerk (2003) and later by Feng *et al* (2005). Using different values of M in Equation (18) to determine the number of terms that can give the best accuracy, it was found that $M = 9$ terms is the one that satisfied this objective. The statistics shown in Table 5.3 justify the choice of $M = 9$.

Table 5.3 Statistics of Trigonometric Interpolation using different number of terms

Number of Terms	MEAN (cm)			STD (cm)			MAX (cm)		
	dx	dy	dz	dx	dy	dz	dx	dy	dz
5	0.0024	0.0210	0.0115	0.2083	0.4020	0.4018	1.0	1.3	1.7
7	0.0017	0.0028	0.0024	0.046	0.0794	0.0606	0.4	0.3	0.4
9	0.0010	0.0035	0.0007	0.0499	0.0841	0.0654	0.3	0.5	0.4
11	0.00	0.0050	0.0030	0.0621	0.0973	0.0827	0.6	0.9	0.7
13	0.0017	0.0084	0.0038	0.1186	0.1465	0.1016	1.3	1.9	1.3

The results of interpolating INERTIAL and ECEF orbit using 9 terms are shown in figures 5.12 and 5.13 respectively. Again, the transformation error adds up to the interpolation error in the case of ECEF orbit.

Figure 5.12 Trigonometric Interpolation error for GPS INERTIAL orbit

Figure 5.13 Trigonometric Interpolation error for GPS ECEF orbit

5.2 Interpolation Using Broadcast Ephemeris

Previous discussion focused on the interpolation of the actual precise orbit using four different interpolation methods. However, as a result of the high GPS altitude, interpolating the actual orbit is expected to produce a relatively large interpolation error.

To overcome this problem, the residuals between the broadcast and precise orbits are formed and interpolated. The following steps summarize the interpolation procedure using broadcast ephemeris (Yousif and El-Rabbany, 2007):

- (i) The broadcast navigation file that corresponds to the test data set for January 1, 2002 was downloaded from the Crustal Dynamics Data Information System server.
- (ii) The orbital parameters of PRN01 were extracted and used to calculate the broadcast ephemeris at 15 minute intervals.
- (iii) The residuals between the broadcast ephemeris - calculated in the step (ii) - and the precise rapid ephemeris of January 1, 2002 were obtained.
- (iv) Then those residuals were interpolated at 5 minutes intervals using Lagrange method. It is worthy of mention that the residuals range from -4 m to 4 m, which makes it easier to obtain millimetre accuracy of interpolation.
- (v) Again, the orbital parameters of PRN01 were used to compute the broadcast ephemeris, but this time at 5 minute intervals.
- (vi) The broadcast ephemeris at 5 minute intervals were added to the residuals of step (iii) to give the ECEF coordinates at 5 minute intervals.
- (vii) The output was then compared to the reference ECEF coordinates and the errors for the first 2 hours were plotted in Figure 5.14.

As can be seen in Figure 5.14, the maximum interpolation error is about -1.6 mm, which shows that this approach produces better results than the previous approaches. With the improvement in the GPS broadcast ephemeris as part of the modernization program, it is expected that the use of more recent navigation files would further reduce the interpolation error. It should be mentioned, however, that as the ephemeris record is only

valid for two hours, a larger interpolation error would be expected whenever a new record is used.

Figure 5.14. Interpolation error using the broadcast ephemeris

5.3 Precise Orbit Prediction Using Artificial Neural Networks

Most of the GPS real-time and near real time applications require the availability of predicted orbit ephemeris that is more accurate than the IGS ultra-rapid ephemeris product. This research takes advantage of the Artificial Neural Networks to produce more precise predicted ephemeris than the IGS products. The prediction process in this research is performed through three different steps. The first two steps employ the Bernese 5.0 software. An initial 6-hour predicted orbit is first generated, using 5-day IGS rapid precise products (ephemeris, clock corrections, Earth rotation parameters (*ERP*) files) preceding the predicted part. The initially predicted orbit is then improved, using 24-hour GPS observations from a network of well-distributed global reference stations,

corresponding to the day preceding the predicted part. The third step utilizes the power of Artificial Neural Networks to improve the Bernese-generated predicted orbit.

5.3.1 Initial Prediction using Bernese Software

The GPS satellite orbit prediction by Bernese can be summarized into two main steps, as follows (Dach *et al*, 2007):

a. Preparation of orbit fitting and partial derivatives with respect to orbital parameters

- The procedures start with importing IGS rapid precise ephemeris for five days prior to the day to be predicted.
- The IGS Earth orientation parameters corresponding to the same period are also imported and converted from IERS format to Bernese format. The ERP file is necessary for the transformation from the Earth-Centered Earth-Fixed (ECEF) reference system to the inertial frame.
- Each of the precise ephemeris is converted to Bernese tabular format (*.TAB). The resulted tabular orbit is the satellites' coordinates, expressed in the J2000 inertial frame. This transformation simplifies curve fitting and numerical integration.
- The satellites' clock correction coefficients are then extracted and saved to a (*.CLK) file, which is used later for receiver clock synchronization. The clock corrections are fitted to a second-degree polynomial with 12-hour intervals.

- The five precise ephemeris files are concatenated and fitted to a curve of 10^{th} degree polynomial. To achieve a good fit, Bernese does not treat the entire orbit with the same polynomial coefficients. Instead, it assigns a different set of polynomial coefficients for every hour segment. Based on the 10^{th} degree polynomial fitting, the concatenated curve is then extrapolated for 6 hours to generate an initial prediction, as shown in Figure 5.15. The final panel of concatenation includes the start and end time where the 6 hours could be added.

Figure 5.15 Residuals of polynomial fitting before orbit improvement

- The tabular orbit file of the day right before the predicted part is used to generate the so-called standard orbit (Bernese-internal in binary format) and the radiation pressure files. The coefficients of the ephemeris polynomial are stored in the standard (*.STD) binary file while the coefficients of the radiation pressure are stored in (*.RPR) binary file. The radiation pressure

coefficients are 9 parameters according to the empirical CODE Extended Radiation Pressure Model (Beutler *et al*, 1994).

- The partial derivatives of the orbit coordinates with respect to the radiation pressure coefficients and the six Keplerian parameters are obtained and stored in the (*.RPR) file. The six Keplerian parameters refer to the ones corresponding to the precise ephemeris. These parameters can easily be computed if the satellite's position $\underline{\rho}$ and velocity $\underline{\dot{\rho}}$, at a given epoch, are known:

$$\underline{\rho} = f(a, e, M, \omega, \Omega, i)$$

$$\underline{\dot{\rho}} = g(a, e, M, \omega, \Omega, i)$$

Since we have three components for each vector, the above vector equations can be decomposed into six scalar equations, which are sufficient for computing the six parameters. Bernese uses a subroutine, called *XYZELE.f*, to compute the osculating parameters from one set of position and velocity.

b. Generation of improved orbit using RINEX observation files

- RINEX observation files from 35 well-distributed global stations, for the day preceding the predicted part, are imported into Bernese. Figure 5.16 shows the distribution of these stations.

Figure 5.16 Distribution of selected IGS stations

- Imported code and carrier-phase information are stored separately in code header files (*.CZH), code observation files (*.CZO), carrier-phase header files (*.PZH) and carrier-phase observation files (*.PZO).
- The zero-difference code observations are used for the receiver clock synchronization with respect to the GPS time. Using the previously extracted satellite clock corrections (*.CLK) and other observables, the receiver clock corrections are computed through the least-square estimation of the unknown parameters. In Bernese, the code observations are used only for clock synchronization.
- Station baselines are then created, and the double difference linear combinations are performed. Bernese automatically selects a set of baselines with the maximum number of observations. In our case, the number of baselines was 34.

- The carrier-phase observables are preprocessed for cycle slip detection and repair. If the cycle slips can not be repaired a new ambiguity parameter is established.
- Data screening is performed to mark and exclude bad observations, such as those at low satellite elevations or epochs with unpaired data (*L1* without *L2* or vice versa).
- A float solution is then carried out, followed by ambiguity fixing for each of the 34 baselines individually.
- The ambiguity resolution results from the previous step are used for the final solution.
- The most important output of the final solution is the improved orbital parameters file (*.ELE) and the Earth orientation parameters file (*.ERP).
- The improved orbital elements file (*.ELE) and the Earth orientation parameters file (*.ERP) are used to improve the previously generated initial orbit. After updating the orbit, the residuals exhibit some improvement, as shown in Figure 5.17.

Figure 5.17 Residuals of the polynomial fitting after improvement

5.3.2 Prediction Enhancement using Artificial Neural Networks

- The residuals between the five rapid precise and fitted ephemeris are calculated.
- For generality, three GPS satellites from different generations were used to test the prediction in this research, namely, PRN04 (Block IIA), PRN13 (Block IIR) and PRN17 (Block IIR-M).
- The residuals are used to train a feed-forward back-propagation neural network. Each of the three Cartesian coordinates is trained independently to minimize the load on the network.
- Different network structures are tried by varying the network parameters such as the transfer function, training algorithm, goal and number of hidden layers. Eventually the network was settled to the following configuration:
 - The transfer function selected was the *Hyperbolic Tangent Sigmoid* (*TANSIG*), which is mathematically equivalent to $\tanh(x)$. The *TANSIG* runs faster than *LOGSIG* in MATLAB.

- The best training algorithm for this research was found to be *Levenberg-Marquardt algorithm* (LMA).
 - The goal was set to 10^{-5} .
 - The number of iterations was 3000.
 - Even though the above parameters were common for all data sets, the number of hidden layers was not kept constant. Each data set had to be treated with different number of hidden layers, which varied between 15 and 25 layers. The reason for this is the various lengths of the training data.
 - The data sets had to be trained individually because running the network, with all the data as one ensemble, would produce relatively low accuracy, and at the same time, slow down the network.
- After training the network, the residuals for six hours of the new day are predicted.
 - The ANN-predicted residuals are added to the improved 6-hour arc produced by Bernese.

5.3.3 Prediction Results

As indicated earlier, we used a 5-day IGS rapid ephemeris to generate an initial 6-hour predicted orbit. Making numerous trials, we found that the 5-day ephemeris produce satisfactory results. In this research, we used the IGS rapid ephemeris for the period February 5, 2007 to February 9, 2007. It is important to note that the rapid ephemeris of February 9, 2007 will be delayed by 17 hours and, as such, it will not be acceptable to use it for prediction. However, the observed half of February 9, 2007 is delayed by only 3

hours, which makes it reasonable to use it instead of rapid ephemeris. Another option is to produce the rapid ephemeris using Bernese. The precise ephemeris were concatenated and extrapolated to generate an initially predicted 6-hour orbit of February 10, 2007 as described in 3.1 above. Figure 5.15 shows the residuals of the fitted orbit compared with the corresponding precise rapid orbit before any improvement is applied. As shown in Figure 5.15, the initially predicted part is far from the desired accuracy and, therefore, has to be improved by a subsequent process. The RINEX observation files of a network of 35 IGS stations were downloaded from the *Cadastral Dynamics Data Information System* (CDDIS) ftp server, corresponding to the 24-hour period of February 9, 2007. We found that the number of stations should neither be too low nor too high. Performing many trials, the number of stations was eventually set at 35 global stations (see Figure 5.16). The RINEX observations were processed to produce an osculating orbital element file, which was used to improve the fitted orbit in the final step. Figure 5.17 shows the residuals of the fitted orbit after improvement.

The Bernese-produced improved predicted orbit, although better than the initial prediction, does not fulfill the requirement of many GPS applications. As such, the Bernese-produced improved predicted orbit should be subjected to further enhancement. Almost all orbit prediction strategies apply sophisticated procedures to model the stochastic parameters of orbit ephemeris. ANN, on the other hand, does not presume any normal or linear data distribution (Haykin, 1999). In addition, ANN does not require the data to be preprocessed or smoothed before the analysis. Smoothing the data beforehand may truncate a genuine part of the data. Those were the reasons that ANN was resorted to in this research where the residuals of five preceding days were used to train the ANN.

The MATLAB default training algorithm was too slow. Therefore, it was replaced by another MATLAB-based tool (Neural Network System Identification Toolbox), which was developed by Magnus Nørsgaard at the Technical University of Denmark.

The ANN consumes computer resources when the training data is relatively large. Therefore attempts were made to reduce the size of data and, at the same time, maintain a good quality of output. To achieve this, each of the three Cartesian components was trained separately and the training data was limited to only three days. Then, after training the ANN, residuals of February 10th, 2007 were lastly produced. The predicted residuals were added to the improved orbit to obtain the predicted orbit for the first six hours of February 10th, 2007. To find the accuracy of the predicted orbit, we compared it with the final IGS orbit. Figures 5.18, 5.19, and 5.20 show the residuals (i.e., final IGS orbit – predicted orbit) and the standard deviations values for three randomly selected satellites from Blocks IIA (PRN04), IIR (PRN13) and IIR-M (PRN17), respectively.

Figure 5.18 Prediction Errors for PRN04 (Block IIA)

Figure 5.19 Prediction Errors for PRN13 (Block IIR)

Figure 5.20 Prediction Errors for PRN17 (Block IIR-M)

Comparing the standard deviation values of the obtained results, it can be seen from Figures 5.18 through 5.20 that the accuracy of the ANN-based orbital prediction, for the selected satellites, is better than that of the IGS ultra-rapid orbit. Table 5.4 shows the total prediction errors for the above-mentioned satellites and the percentage of improvement when the newly developed ANN-based orbital prediction model is used. It can be seen that an improvement of up to 47% is obtained with the ANN-based orbital prediction model. This makes the ANN-based prediction method more suitable for high-precision near real-time applications.

Table 5.4 Prediction Error (cm) and ANN Percentage improvement

Satellite ID	PRN 04		PRN 13		PRN 17	
Prediction method	IGS-U	ANN	IGS-U	ANN	IGS-U	ANN
x	2.1	1.0	0.9	0.9	1.1	1.1
y	1.5	1.2	3.4	2.7	2.2	1.5
z	2.6	1.1	1.7	1.1	2.5	1.1
total	3.6	1.9	3.9	3.0	3.5	2.2
Percentage improvement	47%		22%		37%	

6 CONCLUSIONS AND FUTURE RECOMMENDATIONS

6.1 Conclusions

The first part of this research examined the performance of four different interpolation methods, namely, Lagrange, Newton Divided Difference, Cubic Spline and Trigonometric methods. Apart from Cubic Spline method, the other three methods produced a relatively good accuracy. The structure of the Cubic Spline, which is essentially a third degree polynomial, made it incapable of representing the GPS orbit. Lagrange and Newton Divided Difference methods demonstrated completely identical results in terms of interpolation error. The Trigonometric method gave the best results due to the periodic nature of the GPS orbit. However, as far as boundaries are concerned, the Trigonometric method is less accurate than Lagrange near the boundaries. In fact the trigonometric method gives better results if the boundaries are not considered. For this reason it is advisable to use Lagrange in near real-time applications where the boundaries can never be disregarded. For non real-time applications, it is advisable to locate the orbital data that needs to be interpolated at the centre in order to avoid boundary interpolation failure. The interpolation via the broadcast ephemeris has produced even better accuracy, particularly at the boundaries where the error is 0.15 cm, which is less than those of other methods.

In the second part of this research, a three-stage neural network-based orbital prediction model has been developed. An initial 6-hour predicted orbit is first obtained by

extrapolating a 5-day IGS rapid orbit. The initial orbit is then improved by using GPS observations from a network of 35 globally distributed IGS reference stations. Lastly, a modular, three-layer feed-forward back-propagation neural network is used to enhance the predicted orbit. It has been shown that the obtained accuracy of ANN-based orbital prediction for some satellites is better than the corresponding predicted part of the IGS ultra-rapid orbit. In comparison with the IGS ultra-rapid orbit, the newly developed neural network-based model improved the orbit prediction by 47%, 23%, and 37% for three randomly-selected satellites from Blocks IIA, IIR and IIR-M, respectively. This makes the ANN-based prediction method more suitable for near real-time applications requiring high-precision orbit. Trigonometric and Lagrange interpolation methods can be applied to predicted orbit to produce precise ephemeris at high rates.

6.2 Future Recommendations

The accuracy of, not only the IGS ultra-rapid orbit, but also the satellite clock prediction accuracy needs to be improved for near real time applications. The current IGS ultra-rapid accuracy of the clock prediction in the predicted half is in the order of 5 ns, while those of the rapid and final products are 0.1 ns and less than 0.1 ns, respectively. Since the clock prediction is 50 times worse than the rapid and final products, it is recommended that a better prediction model be developed in the future to enhance the satellite clock prediction. Also, investigation of the correlation between the clock error and orbital prediction is recommended, which would enhance the obtained positioning solution and its covariance matrix. At present, the correlation between satellite

coordinates and clock correction is recorded as part of the SP3 precise ephemeris file (Hilla, 2002). This correlation may well be detected if ANN is involved.

The orbital prediction in true real time using a physical model is yet another field to be explored.

REFERENCES

- Bates, D. M. and Watts, D. G. (1988). "Nonlinear Regression and Its Applications", New York: Wiley.
- Beutler G., (2005). "Methods of Celestial Mechanics". Springer.
- Beutler G., Kouba J., and Springer T. (1995). "Combining the Orbits of the IGS Analysis Centers", *Bullitica Geoditica*, Vol. 69, No. 4, pp. 200-222.
- Beutler G., Brockmann E., Hugentobler U., Mervart L., Rothacher M. and Weber R. (1994). "Extended Orbit Modeling Techniques at the CODE Processing Center of the International GPS Service: Theory and Initial Results", *Manuscripta Geodetica* 19, pp. 367-386.
- Bishop C.M. (1995). "Neural Networks for Pattern Recognition", Oxford: Clarendon Press.
- Crustal Dynamics Data Information Service (CDDIS). RINEX navigation file accessed at (<ftp://cddis.gsfc.nasa.gov/pub/gps/gpsdata/brdc/>) on January 5th, 2006.
- Crustal Dynamics Data Information System (CDDIS). RINEX observation files accessed at (<ftp://cddis.gsfc.nasa.gov/gps/data/daily/2007>) on April 2nd , 2007.
- Dach R., Hugentobler U., Fridez P. and Meindl M. (2007). "Bernese GPS Software Version 5.0 Manual, Astronomical Institute", University of Bern.
- Dousa J. (2003). "GPS Orbit Determination For Meteorology Applications", Research Institute of Geodesy, Topography and Cartography.
- El-Rabbany A. (2006). "Introduction to GPS: The Global Positioning System", Artech House Inc.

- El-Rabbany A., Auda G. and Abdelazim S. (2002). "Predicting Sea Ice Conditions Using Neural Networks", *The Journal of Navigation* 55, 137-143.
- El-Rabbany A. and El-Diasty M. (2003), "A New Approach to Sequential Tidal Prediction", *Journal of Navigation* 56, 305-314.
- Feng, Y. and Yi Zheng (2005). "Efficient Interpolation to GPS Orbits for Precise Wide Area Applications" *GPS Solutions*, Vol. 9, No. 4, pp. 273-282.
- Geomatics for Informed Decisions (2006). "Next-generation Algorithms for Navigation, Geodesy and Earth Science under Modernized GNSS", (<http://gge.unb.ca/Research/GRL/GNSS/QuebecTW2006.htm>), Accessed on November 20th, 2007.
- Gurtner, W., (2000). "RINEX: The Receiver-Independent Exchange Format", *GPS World*, Vol. 5, No. 7, pp. 48-52.
- Haykin S., (1999). "Neural Networks: A Comprehensive Foundation". Second Edition. Prentice Hall.
- Hilla, S., (2002). "Extending the Standard Product 3 (SP3) Orbit Format", *Proceedings of the International GPS Service Network, Data, and Analysis Center Workshop*, Ottawa, Canada, April 8-11, 2002.
- Hobson M. P., G. P. Efstathiou and A. N. Lasenby (2006). "General Relativity: An Introduction for Physics", Cambridge University Press.
- Huang S., Tan K. K. and Tang K. Z., (2004). "Neural Network Control: Theory and Applications", Research Studies Press Ltd, England.
- Gill, P. E. and Murray W. (1978). "Algorithms for The Solution of The Nonlinear Least-squares Problem", *SIAM J. Numerical. Analysis* 15 [5] 977-992.

- IGS Analysis Coordinator (GFZ), Potsdam, Germany (2007), http://www.gfz-potsdam.de/pb1/igsacc/index_igsacc.html, Accessed on September 19, 2007.
- International GNSS Service (2007). "GPS precise products", (http://igsb.jpl.nasa.gov/components/prods_cb.html), Accessed on April 2nd, 2007.
- International GNSS Service (2007). "Accuracy of IGS precise products", (<http://igsb.jpl.nasa.gov/igsb/resource/igssheet.pdf>), Accessed on October 15th, 2007.
- International GNSS Service (2007). "IGSMail 2717, JPL ultra-rapid orbit", (<http://igsb.jpl.nasa.gov/mail/igsmail/2000/msg00057.html>), Accessed on May 20th, 2007.
- International GNSS Service (2007). "NRCan ultra-rapid clock estimation", (<http://igsb.jpl.nasa.gov/mail/igsmail/2002/msg00478.html>) Accessed on November 20th, 2007.
- John A. Tierney (1989). "Differential Equations", Wm. C. Brown. Publishers, Dubuque, Iowa.
- Kouba J. (2003). "A Guide to using IGS Products, Natural Resource Canada", (<http://igsb.jpl.nasa.gov/igsb/resource/pubs/GuidetoUsingIGSProducts.pdf>), Accessed on September 19, 2007.
- Kavzoglu T. and Saka M. H. (2005). "Modelling local GPS/Levelling Geoid Undulations using Artificial Neural Networks", Journal of Geodesy, Vol. 78 No. 9, pp. 520-527.
- Miima JB, Niemiemeier W. and Kraus B (2001). "A Neural Network Approach to Modelling Geodetic Deformations". Proceedings of the 2nd Symposium on Geodesy

- for Geotechnical and Structural Engineering. Berlin, Germany, May 21-24, pp. 186-194.
- McCulloch, W. S. and W. Pitts, (1943). "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133.
- Motenbruck O. and E. Gill, (2001). "Satellite Orbits: Models, Methods and Applications", Springer-Verlag Berlin Heidelberg New York.
- Nelson Robert, (2002). "The Global Positioning System: A detailed look at the miracle of modern navigation", Satellite Engineering Research Corporation, www.atcourses.com/global_positioning_system.htm
- NOAA/NGS/NOS (2006), "GPS Toolbox", <http://www.ngs.noaa.gov/gps-toolbox/sp3intrap>, Accessed on March 15th, 2006.
- Nørgaard M. (2000). "Neural Network Based System Identification Toolbox", Tech. Report. 00-E-891, Department of Automation, Technical University of Denmark.
- Press, W.H., S.A. Teukolosky, W.T. Vetterling, B.P. Flannery (2002). *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press.
- Remondi, B. (1985). "Distribution of Global Positioning System Ephemeris by the National Geodetic Survey", Presented at the First Conference on Civil Applications of GPS - Institute of Navigation, September 12, 1985, 8 pp.
- Remondi, B. (1989). "Extending the National Geodetic Survey Geodetic Orbit Formats", NOAA Technical Report 133 NGS 46, 85 pp.
- Remondi, B. (1991). "NGS Second Generation ASCII and Binary Orbit Formats and Associated Interpolation Studies", Poster presentation at the 20th General Assembly of the IUGG, Viena, 11 – 24 August.

- Ripley B.D. (1996), "Pattern Recognition and Neural Networks", Cambridge University Press.
- Romero I., Garcia C., Dow J. and Zandbergen R. (2001). "Moving Precise Orbit Determination Towards Real-Time", Proceedings of GNSS 2001, Seville, Spain, May 2001, pp. 1-4.
- Rumelhart D.E., Hinton J.E and Williams R.J. (1986), "Learning Representations of Back-propagation Error", Nature (London), Vol. 323, pp. 533-536.
- Schenewerk, M. (2003). "A Brief Review Of Basic GPS Orbit Interpolation Strategies." GPS Solutions, Vol. 6, No. 4, pp. 265-267.
- Schuh H. Ulrich M., Egger D., Müller J. Schwegmann W. (2002). "Prediction of Earth Orientation Parameters by Artificial Neural Networks", Journal of Geodesy Vol. 76, No. 5, pp. 247-258.
- Schueler T. (1998). "On the interpolation of SP3 orbit files." IfEN Technical IFEN-TropAC-TN – 002 – 01. Institute of Geodesy and Navigation, University of Federal
- Spofford, P. R. and Remondi, B., (1994). "The National Geodetic Survey Standard GPS Format SP3. (SP3-a format)", available from the IGS website: http://igsjb.jpl.nasa.gov/igsjb/data/format/sp3_docu.txt
- Spiegel, M. R. (1999). Mathematical Handbook of Formulas and Tables. McGraw Hill. Armed Forces, Munich.
- Spilker, J., and Bradford W. Parkinson (1996). "Overview of GPS Operation and Design, Global Positioning System: Theory and Applications", Volume 1, Chap 2, Progress in Astronautics and Aeronautics, Volume 163, American Institute of Astronautics and Aeronautics Inc.

- USNO (2007), "GPS Operational Satellites (BLOCK II/IIA/IIR/IIR-M)",
<ftp://tycho.usno.navy.mil/pub/gps/gpsb2.txt>, Accessed on October 15th, 2007.
- Vapnik V. N., (1995). "The Nature of Statistical Learning Theory", New York, Springer-Verlag.
- Vapnik V. N. and C. Cortes, (1995). "Support Vector Machine, Machine Learning", Vol. 20, pp. 273 - 297.
- Widrow B., M. C. Yovitz. G. T. Jacobi and G. D. Goldstein. (1959). "Generalization of Information Storage in Networks of ADALINE Neurons", Spartan Books, Washington D. C.
- Widrow, B. and Hoff, (1960). "Adaptive Switching Circuits", IRE WESCON Convention Record, pages 96 - 104.
- Yousif H. and A. El-Rabbany (2007). "Assessment of Several Interpolation Methods for Precise GPS Orbit", Journal of Navigation Vol. 60, pages 443 - 455.

APPENDIX I: SUPPLEMENTARY INTERPOLATION RESULTS

I.1 Lagrange Interpolation Error for PRN 03

Figure I.1 Lagrange Interpolation Error for PRN 03

Table I.1 Statistics of Lagrange Interpolation Error for PRN 03

MEAN (cm)			Standard Deviation (cm)		
dx	dy	dz	dx	dy	dz
0.0004	0.0020	0.0010	0.035	0.05	0.04

I.2 Lagrange Interpolation Error for PRN 10

Figure I.2 Lagrange Interpolation Error for PRN 10

Table I.2 Statistics of Lagrange Interpolation Error for PRN 10

MEAN (cm)			Standard Deviation (cm)		
dx	dy	dz	dx	dy	dz
0.001	0.006	0.004	0.04	0.05	0.09

I.3 Lagrange Interpolation Error for PRN 20

Figure I.3 Lagrange Interpolation Error for PRN 20

Table I.3 Statistics of Lagrange Interpolation Error for PRN 20

MEAN (cm)			Standard Deviation (cm)		
dx	dy	dz	dx	dy	dz
0.001	0.002	0.003	0.047	0.047	0.045

I.4 Lagrange Interpolation Error for PRN 29

Figure I.4 Lagrange Interpolation Error for PRN 29

Table I.4 Statistics of Lagrange Interpolation Error for PRN 29

MEAN (cm)			Standard Deviation (cm)		
dx	dy	dz	dx	dy	dz
0.0004	0.0019	0.0050	0.044	0.072	0.057

APPENDIX II: PREDICTION COMPARISON BETWEEN IGS, CODE AND ANN

II.1 6-hour prediction for PRN04

Figure II.1 Prediction Error for PRN 04

Figure II.2 STD Error for PRN 04

II.2 6-hour prediction for PRN13

Figure II.3 Prediction Error for PRN 13

Figure II.4 STD Error for PRN 13

II.3 6-hour prediction for PRN17

Figure II.5 Prediction Error for PRN 17

Figure II.6 STD Error for PRN 17