

NONLINEAR BAYESIAN FILTERING WITH APPLICATIONS TO
ESTIMATION AND NAVIGATION

A Dissertation

by

DEOK-JIN LEE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2005

Major Subject: Aerospace Engineering

NONLINEAR BAYESIAN FILTERING WITH APPLICATIONS TO
ESTIMATION AND NAVIGATION

A Dissertation

by

DEOK-JIN LEE

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Kyle T. Alfriend
(Chair of Committee)

John L. Junkins
(Member)

Srinivas R. Vadali
(Member)

Alexander Parlos
(Member)

Helen L. Reed
(Head of Department)

May 2005

Major Subject: Aerospace Engineering

ABSTRACT

Nonlinear Bayesian Filtering with Applications to
Estimation and Navigation. (May 2005)

Deok-Jin Lee,

B.S., Chonbuk National University;

M.S., Texas A&M University

Chair of Advisory Committee: Dr. Kyle T. Alfriend

In principle, general approaches to optimal nonlinear filtering can be described in a unified way from the recursive Bayesian approach. The central idea to this recursive Bayesian estimation is to determine the probability density function of the state vector of the nonlinear systems conditioned on the available measurements. However, the optimal exact solution to this Bayesian filtering problem is intractable since it requires an infinite dimensional process. For practical nonlinear filtering applications approximate solutions are required. Recently efficient and accurate approximate nonlinear filters as alternatives to the extended Kalman filter are proposed for recursive nonlinear estimation of the states and parameters of dynamical systems. First, as sampling-based nonlinear filters, the sigma point filters, the unscented Kalman filter and the divided difference filter are investigated. Secondly, a direct numerical nonlinear filter is introduced where the state conditional probability density is calculated by applying fast numerical solvers to the Fokker-Planck equation in continuous-discrete system models. As simulation-based nonlinear filters, a universally effective algorithm, called the sequential Monte Carlo filter, that recursively utilizes a set of weighted samples to approximate the distributions of the state variables or parameters, is investigated for dealing with nonlinear and non-Gaussian systems. Recent

particle filtering algorithms, which are developed independently in various engineering fields, are investigated in a unified way. Furthermore, a new type of particle filter is proposed by integrating the divided difference filter with a particle filtering framework, leading to the divided difference particle filter. Sub-optimality of the approximate nonlinear filters due to unknown system uncertainties can be compensated by using an adaptive filtering method that estimates both the state and system error statistics. For accurate identification of the time-varying parameters of dynamic systems, new adaptive nonlinear filters that integrate the presented nonlinear filtering algorithms with noise estimation algorithms are derived.

For qualitative and quantitative performance analysis among the proposed nonlinear filters, systematic methods for measuring the nonlinearities, biasness, and optimality of the proposed nonlinear filters are introduced. The proposed nonlinear optimal and sub-optimal filtering algorithms with applications to spacecraft orbit estimation and autonomous navigation are investigated. Simulation results indicate that the advantages of the proposed nonlinear filters make these attractive alternatives to the extended Kalman filter.

To My parents and lovely wife

ACKNOWLEDGMENTS

I would like to express my deepest thanks to my dissertation advisor, Professor Kyle T. Alfriend. I thank him for his continuous motivation, encouragement, and guidance with which I could finish my dissertation.

I would also like to thank professors John L. Junkins, Srinivas R. Vadali, and Alexander Parlos for their help and for being on my dissertation committee.

Most of all, I'd like to thank my family to whom I dedicated this work, especially my wife Yun Sun for all the love she brought into my life and my parents for their graceful support.

TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION	1
	1.1 Overview	1
	1.2 Recursive Bayesian Estimation	4
	1.3 Review to Nonlinear Filtering	8
	1.4 Contributions of the Dissertation	12
	1.5 Dissertation Outline	14
II	OPTIMAL LINEAR FILTERING	16
	2.1 Batch Least-Squares Estimation	16
	2.2 Weighted Least-Squares Estimation	17
	2.2.1 Relationship to Maximum Likelihood Estimation	19
	2.2.2 Relationship to Bayesian Estimation	19
	2.2.3 Unbiased Estimator	21
	2.3 Linear Sequential Estimation	22
	2.4 Kalman Filter	25
	2.4.1 Introduction	25
	2.4.2 Kalman Filtering Algorithm	27
III	SUBOPTIMAL NONLINEAR FILTERING	32
	3.1 Nonlinear Least-Squares Estimation	32
	3.2 Extended Kalman Filter	36
	3.3 Unscented Kalman Filter	39
	3.3.1 Unscented Transformation	41
	3.3.2 Scaled Unscented Transformation	43
	3.3.3 Unscented Kalman Filter	45
	3.4 Divided Difference Filter	49
	3.4.1 Linearization	50
	3.4.2 First-Order Approximation	52
	3.4.3 Second-Order Approximation	53
	3.4.4 First-Order Divided Difference Filter	55
	3.4.5 Second-Order Divided Difference Filter	59
	3.5 Finite Difference Filter	62
	3.5.1 Exact Nonlinear Filter	64

CHAPTER	Page
3.5.1.1 Fokker-Planck Equation	64
3.5.1.2 Measurement Update	65
3.5.2 Approximate Nonlinear Filter	66
3.5.2.1 Prediction Equation	68
3.5.2.2 Measurement Update	72
3.5.2.3 Finite Difference Nonlinear Filter	73
IV PARTICLE FILTERING	74
4.1 Introduction	74
4.2 Optimal Recursive Bayesian Estimation	75
4.3 Particle Filtering	77
4.3.1 Sequential Importance Sampling	79
4.3.2 Resampling	83
4.3.2.1 Effective Sample Size	83
4.3.2.2 Resampling Algorithms	84
4.3.3 Generic Particle Filter Algorithm	87
4.4 SIR Particle Filtering	87
4.5 Improving Particle Filters	90
4.6 Local Linearization Particle Filter	93
4.6.1 Extended Kalman Particle Filter	94
4.6.2 Unscented Particle Filter	96
4.6.3 Divided Difference Particle Filter	98
4.7 Regularized Particle Filter	101
4.7.1 Density Estimation	101
4.7.2 Kernel Methods	103
4.7.3 Regularized Particle Filter	104
4.8 Markov Chain Monte Carlo Method	106
4.8.1 Metropolis-Hastings Algorithm	108
4.8.1.1 Proposal Choice	109
4.8.2 Gibbs Sampler	110
4.9 Rao-Blackwellized Particle Filter	111
4.10 Cramér-Rao Bounds for Nonlinear Filtering	117
4.10.1 Cramér-Rao Lower Bounds	118
4.10.2 Posterior Cramér-Rao Lower Bounds	120
4.10.3 Posterior Cramér-Rao Lower Bound for Gaussian Noise	121
4.10.3.1 Nonlinear Gaussian Filtering Problem	121
4.10.3.2 Linear Gaussian Filtering Problem	122
V ADAPTIVE FILTERING	123

CHAPTER	Page
5.1 Introduction	123
5.2 Optimality Conditions of Kalman Filter	126
5.3 Maybeck's Estimator	128
5.4 Automated Adaptive Nonlinear Estimator	129
5.4.1 Process Noise Covariance Estimation	129
5.4.2 Automated Calculation of Scale Factor	131
5.5 Adaptive Unscented Kalman Filter	133
5.5.1 Unscented Kalman Filter Algorithms	134
5.5.2 Adaptive Unscented Kalman Filtering	138
5.6 Adaptive Divided Difference Filter	139
5.6.1 Divided Difference Filter Algorithms	139
5.6.2 Adaptive Divided Difference Filtering	142
VI ADAPTIVE MODEL COMPENSATION	143
6.1 Dynamic Model Compensation	143
6.1.1 Stochastic Acceleration	143
6.1.2 Filtering Model	146
6.1.2.1 State Transition Matrix	146
6.1.2.2 Process Noise Matrix	150
6.2 Reduced Dynamic Tracking	156
6.2.1 Modified Stochastic Acceleration	156
6.2.2 Filtering Model	159
6.2.2.1 State Transition Matrix	159
6.2.2.2 Process Noise Matrix	161
VII APPLICATIONS TO AEROSPACE SYSTEMS	163
7.1 Orbit and Parameter Estimation	163
7.1.1 Introduction	163
7.1.2 Equations of Orbital Motion	165
7.1.2.1 Gravitational Perturbation	167
7.1.2.2 Drag Perturbation	168
7.1.2.3 Stochastic Drag Model	170
7.1.3 Observation and Tracking Model	172
7.1.3.1 Radar Tracking	172
7.1.3.2 Laser Tracking	177
7.1.4 Initial Orbit Determination	178
7.1.4.1 Herrick-Gibbs Method	179
7.1.5 Orbit and Parameter Estimation	182

CHAPTER	Page
7.1.5.1 Satellite Visibility Analysis	183
7.1.5.2 Gauss Least-Squares Differential Correction . . .	183
7.1.5.3 Nonlinear Filtering Algorithm	188
7.1.6 Error Sources and Compensation	192
7.1.6.1 Measurement Error	192
7.1.6.2 Mathematical Models	193
7.2 Autonomous GPS Navigation	193
7.2.1 GPS Navigation Systems	194
7.2.1.1 GPS Systems	194
7.2.1.2 GPS Satellite Constellation	195
7.2.1.3 GPS Measurement Models	196
7.2.1.4 GPS Positioning Accuracy	200
7.2.1.5 Coordinate Transformation	203
7.2.2 Navigation Solution	203
7.2.3 Kinematic Navigation Solutions	205
7.2.3.1 Geometric State Solution	205
7.2.3.2 Least-Squares Solution	208
7.2.4 Dynamic Navigation Solution	209
7.2.4.1 State Noise Compensation Method	210
7.2.4.2 Dynamic Model Compensation Method	216
VIII SIMULATION RESULTS	221
8.1 Performance Criteria	221
8.1.1 Posterior Cramér-Rao Lower Bound	222
8.1.2 Optimality Analysis	223
8.1.3 Whiteness Test	225
8.2 Orbit and Parameter Estimation	226
8.2.1 Generation of True Trajectory	226
8.2.2 Generation of Observations	227
8.2.3 Filter Dynamic Model	228
8.2.4 Simulation Results and Performance Analysis	229
8.2.4.1 Nonlinear Estimation	229
8.2.4.2 Adaptive Nonlinear Filtering	238
8.3 Autonomous GPS Navigation	251
8.3.1 Generation of User True Trajectories	251
8.3.2 Generating of GPS Satellite Trajectories	252
8.3.3 Filter Dynamic Model and Error Sources	253
8.3.4 Simulation Results and Performance Analysis	255

CHAPTER	Page
8.3.5 Conclusion	262
8.4 Autonomous Robot Navigation	263
8.4.1 Robot Dynamic System	264
8.4.2 Sensor Measurements	265
8.4.2.1 Range Measurement	266
8.4.2.2 Speed Measurement	266
8.4.2.3 Measurement Equations	266
8.4.3 Simulation and Performance Analysis	267
8.4.4 Conclusion	270
IX CONCLUSION AND SUMMARY	274
REFERENCES	277
APPENDIX A	293
APPENDIX B	299
APPENDIX C	300
VITA	302

LIST OF TABLES

TABLE	Page
2.1 Sequential Least-Squares (SLS) Algorithm	25
2.2 Kalman Filter (KF) Algorithm	31
3.1 Extended Kalman Filter (EKF) Algorithm	40
3.2 Unscented Kalman Filter (UKF) Algorithm	48
3.3 First-Order Divided Difference Filter (DDF) Algorithm	60
4.1 Systematic Resampling (SR) Algorithm	86
4.2 Generic Particle Filter (PF) Algorithm	88
4.3 SIR Particle Filter Algorithm	91
4.4 Local Linearization Particle Filter (LLPF)	95
4.5 Extended Kalman Particle Filter (EKPF) Algorithm	97
4.6 Unscented Particle Filter (UPF) Algorithm	99
4.7 Divided Difference Particle Filter (DDPF) Algorithm	100
4.8 Regularized Particle Filter (RPF) Algorithm	107
4.9 Metropolis-Hastings Algorithm	109
4.10 Gibbs Sampling Algorithm	111
4.11 Rao-Blackwellized Particle Filter (RBPF) Algorithm	116
8.1 Classical Orbital Elements	226
8.2 User Satellite Orbital Elements	252
8.3 GPS Constellation Elements	253

TABLE	Page
8.4 Simulation Error Sources	255
8.5 RMS Errors from Geometric Solution (Static Solution)	258
8.6 RMS Errors from Dynamic Model Compensation Estimation	262

LIST OF FIGURES

FIGURE	Page
1.1 Prediction and Update Stages for the Recursive Bayesian Estimation . .	6
1.2 Prediction Step for the Recursive Bayesian Estimation with Non-Gaussian Probability Densities	7
1.3 Measurement Update Step for the Recursive Bayesian Estimation with Non-Gaussian Probability Densities	8
2.1 Diagram of Predictor-Corrector Form of the Kalman Filter	30
3.1 Nonlinear Least-Squares Differential Correction	33
3.2 Diagram of Predictor-Corrector Form of the Extended Kalman Filter . .	39
3.3 Schematic Diagram for Mean and Covariance Propagation	43
3.4 Diagram of Predictor-Corrector Form of the Unscented Kalman Filter .	47
3.5 Diagram of Predictor-Corrector Form of the Divided Difference Filter . .	58
3.6 Diagram of Finite-Difference Nonlinear Filtering	73
4.1 Conceptual Description of Sampling Difference between Recursive Filtering Algorithms; EKF (Top), UKF (Middle), and PF (Bottom) . . .	78
4.2 Diagram for Particle Resampling	84
4.3 Systematic Diagram for Generic Particle Filtering	89
4.4 Concept for Moving Samples to Regions of High Likelihood	93
5.1 Downhill Simplex Algorithm for Automated Parameter Tuning	132
5.2 Structure of Adaptive Sigma Point Filters	134
5.3 Diagram for Adaptive Unscented Kalman Filter Algorithm	137

FIGURE	Page
5.4 Diagram for Adaptive Divided Difference Filter Algorithm	141
7.1 Description of the Geometry of Ground-Based Sensor	166
7.2 Description of Radar Site for Satellite Observation	176
7.3 Herrick-Gibbs Method for Initial State and Covariance Information . . .	181
7.4 Diagram for Satellite Visibility Check	184
7.5 Diagram for GLSDC Orbit Estimation	189
7.6 Diagram for the GPS Segments	195
7.7 The GPS Satellite Constellation	196
8.1 Diagram for Orbit Estimation Strategy	230
8.2 Averaged Position and Velocity Errors with Small Initial Errors	232
8.3 Optimality Index for the State and Covariance Estimates	233
8.4 Consistency Analysis for the State and Covariance Prediction	233
8.5 Averaged Position and Velocity Errors for Large Initial Errors	234
8.6 Averaged Position and Velocity Errors After 12-Hour Propagation	236
8.7 Averaged Position and Velocity Errors After 24-Hour Propagation	236
8.8 Absolute Position Estimation Errors in First Track	239
8.9 Absolute Velocity Estimation Errors in First Track	240
8.10 Drag Coefficient Estimation Error Ratio in First Track	240
8.11 Consistency Test with Optimality Index	242
8.12 Measurement Innovation Errors with 2-sigma Bound	242
8.13 Absolute Position Estimation Errors in Second Track	245
8.14 Absolute Velocity Estimation Errors in Second Track	245

FIGURE	Page
8.15 Drag Coefficient Estimation Error Ratio in Second Track	246
8.16 Adaptive Covariance Estimation with Q Adaptation	247
8.17 Consistency Test with Optimality Index	247
8.18 Range Innovation Errors with 2-Sigma Bound	248
8.19 Azimuth Innovation Errors with 2-Sigma Bound	248
8.20 Elevation Innovation Errors with 2-Sigma Bound	249
8.21 Description of the Geometry of GPS Satellite Orbit	254
8.22 Position Errors for Geometric Solution	257
8.23 Velocity Errors for Geometric Solution	258
8.24 Clock Bias Errors for Geometric Solution	259
8.25 Position Errors for DMC Solution	260
8.26 Velocity Errors for DMC Solution	260
8.27 Clock Bias Errors for DMC Solution	261
8.28 Clock Drift Errors for DMC Solution	261
8.29 Diagram for Autonomous Robot Navigation System	264
8.30 Initial Position Particles at $k = 0$	267
8.31 True Paths of Planar Robot Motion	269
8.32 Planar Position Estimation Errors	270
8.33 Speed Estimation Errors	271
8.34 Heading-Angle Estimation Errors	271
8.35 Distance Estimation Errors	272
8.36 Final Updated Position Particles	272

FIGURE	Page
8.37 Evolution of Probability Density Function for x State	273
8.38 Evolution of Probability Density Function for y State	273

CHAPTER I

INTRODUCTION

1.1 Overview

The nonlinear filtering problem consists of estimating the states of a nonlinear stochastic dynamical system. The class of systems considered is broad and includes orbit/attitude estimation, integrated navigation, and radar or sonar surveillance systems.¹ Because most of these systems are nonlinear and/or non-Gaussian, a significant challenge to engineers and scientists is to find efficient methods for on-line, real-time estimation and prediction of the dynamical systems and error statistics from the sequential observations. In a broad sense, general approaches to optimal nonlinear filtering can be described by a unified way using the recursive Bayesian approach.²⁻⁴ The central idea of this recursive Bayesian estimation is to determine the probability density function of the state vector of the nonlinear systems conditioned on the available measurements. This *a posteriori* density function provides the most complete description of an estimate of the systems. In linear systems with Gaussian process and measurement noises, an optimal closed-form solution is the well-known Kalman filter.^{2,5} In nonlinear systems the optimal exact solution to the recursive Bayesian filtering problem is intractable since it requires infinite dimensional processes.⁶ Therefore, approximate nonlinear filters have been proposed. These approximate nonlinear filters can be categorized into five types: (1) analytical approximations, (2) direct numerical approximations, (3) sampling-based approaches, (4) Gaussian mixture filters, and (5) simulation-based filters. The most widely used approximate nonlinear filter is the extended Kalman filter, which is the representative analytical approximate non-

The journal model is *AIAA Journal of Guidance, Control and Dynamics*.

linear filter. However, it has the disadvantage that the covariance propagation and update are analytically linearized up to the first-order in the Taylor series expansion, and this suggests that the region of stability may be small since nonlinearities in the system dynamics are not fully accounted for.⁷ Thus, the purpose of this research is to investigate new and more sophisticated nonlinear estimation algorithms, develop new nonlinear filters, and demonstrate their applications in accurate spacecraft orbit estimation and navigation.

The work presented here involves the investigation of system identification and nonlinear filtering algorithms that are compatible with the general goals of precise estimation and autonomous navigation. In this dissertation, efficient alternatives to the extended Kalman filter (EKF) are suggested for the recursive nonlinear estimation of the states and parameters of aerospace vehicles. First, approximate (suboptimal) nonlinear filtering algorithms, called sigma point filters (SPFs) that include the unscented Kalman filter (UKF),^{8,9} and the divided difference filter (DDF),¹⁰⁻¹² are reviewed. The unscented Kalman filter, which belongs to a type of sampling-based filters, is based on the nonlinear transformation called the unscented transformation in which a set of sampled sigma points are used to parameterize the mean and covariance of a probability distribution efficiently. The divided difference filter, which falls into the sampling-based polynomial filters, adopts an alternative linearization method called a central difference approximation in which derivatives are replaced by functional evaluations, leading to an easy expansion of the nonlinear functions to higher-order terms. Secondly, a direct numerical nonlinear filter called the finite difference filter (FDF) is introduced where the state conditional probability density is calculated by applying fast numerical solvers to the Fokker-Planck equation in continuous-discrete system models.^{13,14}

However, most of the presented nonlinear filtering methods (EKF, UKF, and

DDF), which are based on local linearization of the nonlinear system equations or local approximation of the probability density of the state variables, have yet been universally effective algorithms for dealing with both nonlinear and non-Gaussian system. For these nonlinear and/or non-Gaussian filtering problems, the sequential Monte Carlo method is investigated.^{15,16} The sequential Monte Carlo filter can be loosely defined as a simulation-based method that uses a Monte Carlo simulation scheme in order to solve on-line estimation and prediction problems.¹⁷ The sequential Monte Carlo approach is known as the bootstrap filtering,¹⁸ the condensation algorithm,¹⁹ and the particle filtering.²⁰ The flexible nature of the Monte Carlo simulations results in these methods often being more adaptive to some features of the complex systems.²¹ There have been many recent modifications and improvements on the particle filter.²² However, some of the problems, which are related to choice of proposal distribution, optimal sampling from the distribution, and computational complexity, still remain. This work investigates a number of improvements for particle filters that are developed independently in various engineering fields. Furthermore, a new type of particle filter is proposed by integrating the divided difference filter with a particle filtering framework, leading to the divided difference particle filter.

The performance of the proposed nonlinear filters is degraded when the first and second moment statistics of the observational and system noise are not correctly specified.^{23,24} Sub-optimality of the approximate nonlinear filters due to unknown system uncertainties and/or noise statistics can be compensated by using an adaptive filtering method that estimates both the state and system error statistics.²⁵ For accurate estimation of the time-varying parameters of dynamical systems, a new adaptive nonlinear filter, called the adaptive sigma point filter (ASPF),²⁶ that integrates the presented nonlinear filtering algorithms with a noise estimation algorithm is derived.

For qualitative and quantitative performance analysis among the proposed non-

linear filters, systematic methods for measuring the nonlinearities and optimality of the proposed filters are introduced. The applications of the proposed nonlinear filters to the spacecraft orbit estimation and autonomous navigation problems are considered.

1.2 Recursive Bayesian Estimation

The probabilistic Bayesian approach to estimation of the state \mathbf{x}_k with the set of all measurements $\mathbf{Y}_k = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$ is to calculate the posterior distribution for \mathbf{x}_k conditional on all the measurements \mathbf{Y}_k . Then, the Bayesian solution is to calculate the density $p(\mathbf{x}_k|\mathbf{Y}_k)$. This density will encapsulate all the information about the state vector \mathbf{x}_k which is contained in the measurement \mathbf{Y}_k and the prior distribution of \mathbf{x}_{k-1} . If the density $p(\mathbf{x}_k|\mathbf{Y}_k)$ is known, then optimal estimates of the state can be obtained. In other words, the estimate of \mathbf{x}_k with the measurements \mathbf{Y}_k is given by the conditional expectation of \mathbf{x}_k with \mathbf{Y}_k

$$\hat{\mathbf{x}}_k = E(\mathbf{x}_k|\mathbf{Y}_k) = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{Y}_k) d\mathbf{x}_k \quad (1.1)$$

This can be extended to estimating functions of the state instead of the state itself. Therefore, calculating the conditional probability density function $p(\mathbf{x}_k|\mathbf{Y}_k)$ plays an important role in filtering theory.

The key to calculating the posterior density for the state is *Bayes theorem*,² which states that

$$p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{x}) p(\mathbf{y}|\mathbf{x}) \quad (1.2)$$

In other words, the posterior density for \mathbf{x} with given observations \mathbf{y} is proportional to the prior value of \mathbf{x} multiplied by the likelihood of the observation.

One of the difficulties for obtaining $p(\mathbf{x}_k|\mathbf{Y}_k)$ lies in the high-dimensional inte-

gration, which means that the computational burden will also grow. This can be avoided by using a sequential scheme. It can easily be seen that

$$p(\mathbf{x}_1, \dots, \mathbf{x}_k | \mathbf{Y}_k) \propto p(\mathbf{x}_1, \dots, \mathbf{x}_k | \mathbf{Y}_{k-1}) p(\mathbf{y}_k | \mathbf{x}_k) \quad (1.3)$$

Therefore, marginally integrating out $\mathbf{x}_1, \dots, \mathbf{x}_{k-1}$ gives

$$p(\mathbf{x}_k | \mathbf{Y}_k) \propto p(\mathbf{x}_k | \mathbf{Y}_{k-1}) p(\mathbf{y}_k | \mathbf{x}_k) \quad (1.4)$$

Applying the Markov structure of the system equation leads to

$$p(\mathbf{x}_k, \mathbf{x}_{k-1} | \mathbf{Y}_{k-1}) = p(\mathbf{x}_{k-1} | \mathbf{Y}_{k-1}) p(\mathbf{x}_k | \mathbf{x}_{k-1}) \quad (1.5)$$

where \mathbf{x}_{k-1} can be integrated out to give an equation for $p(\mathbf{x}_k | \mathbf{Y}_{k-1})$ in terms of $p(\mathbf{x}_{k-1} | \mathbf{Y}_{k-1})$. Therefore, the densities of interest can be updated recursively to either take account of a new observation or to consider an estimate of a future state of the system. The preceding argument yields the following equations. First, the probability *prediction* equation⁴ or the Chapman-Kolmogorov (CK) equation is introduced

$$p(\mathbf{x}_k | \mathbf{Y}_{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{Y}_{k-1}) d\mathbf{x}_{k-1} \quad (1.6)$$

and the *update* equation⁴ is obtained by incorporating the latest noisy measurement in terms of the observation likelihood

$$p(\mathbf{x}_k | \mathbf{Y}_k) = c_k p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Y}_{k-1}) \quad (1.7)$$

where c_k is the normalizing factor given by

$$c_k = \left(\int p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Y}_{k-1}) d\mathbf{x}_k \right)^{-1} \quad (1.8)$$

These prediction and correction equations formulate a *recursive Bayesian* estimation

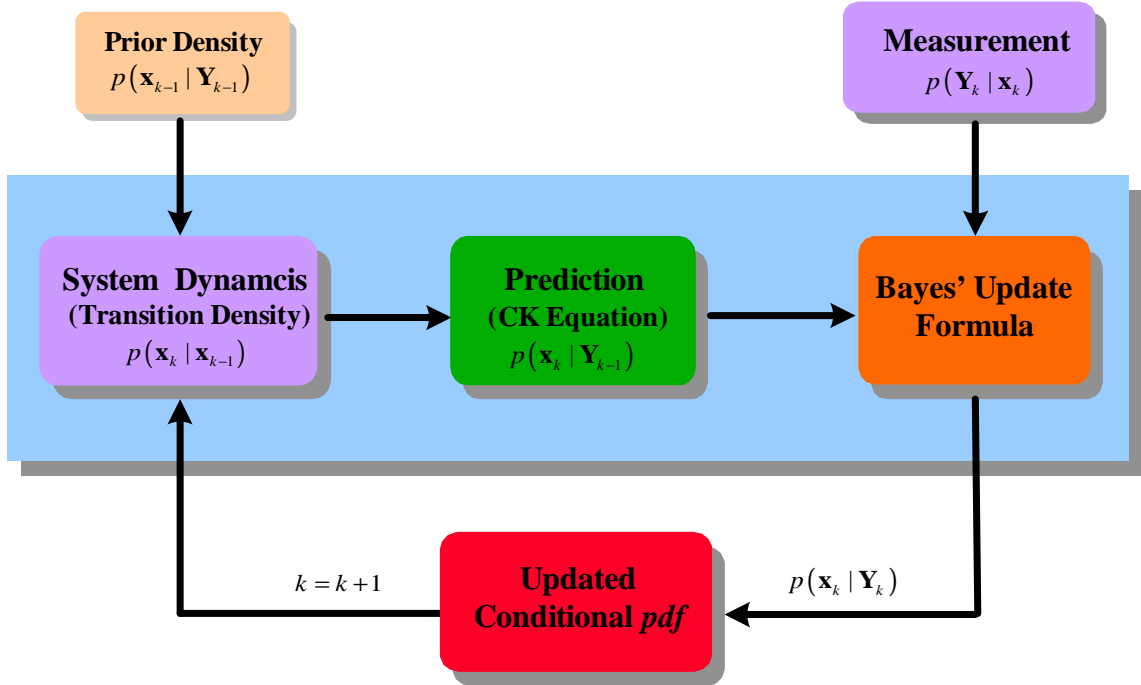


Fig. 1.1 Prediction and Update Stages for the Recursive Bayesian Estimation

algorithm for the filtering problem as illustrated in Fig. 1.1. Note that the update equations (1.7) and (1.8) can be integrated into the following

$$p(\mathbf{x}_k | \mathbf{Y}_k) = \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Y}_{k-1})}{p(\mathbf{y}_k | \mathbf{Y}_{k-1})} \quad (1.9)$$

For intuitive illustration of the recursive Bayesian relations of the predictor-corrector step, a one-dimensional example of the convolution of two non-Gaussian probability densities is represented. The prediction shown in Fig. 1.2 is carried out by making the convolution integral of the *a priori density* and the dynamic transition density. In this step the mean is shifted by the state transition density and the covariance grows with the effect of the process noise. In the correction step in Fig. 1.3, the update is obtained by combining the predicted and actual observations. As can be seen from this example the conditional probability densities in the predicted and up-

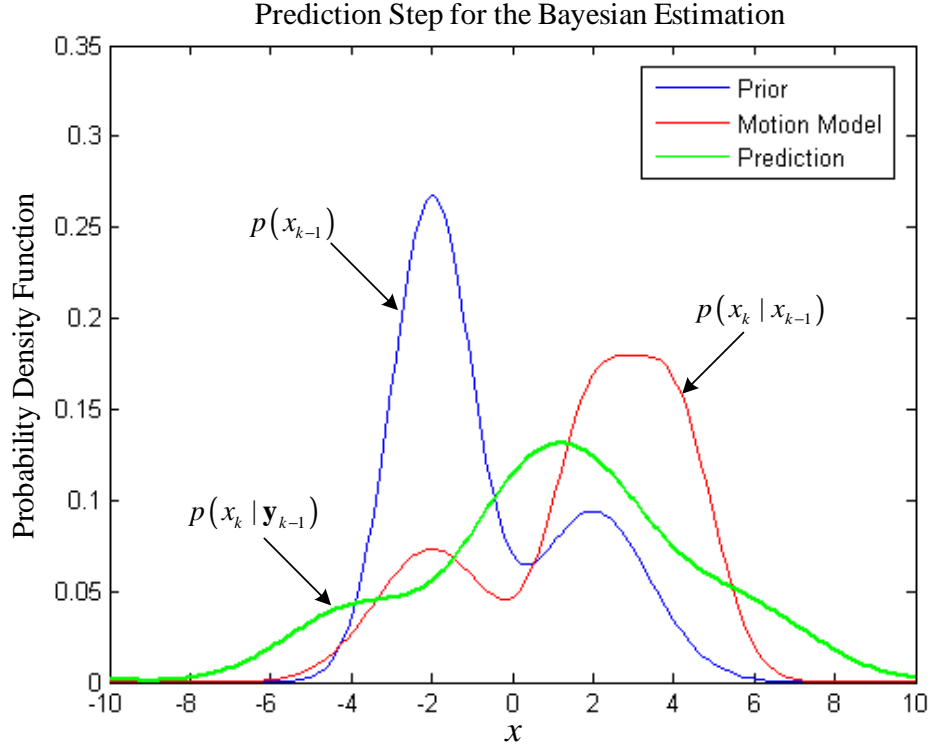


Fig. 1.2 Prediction Step for the Recursive Bayesian Estimation with Non-Gaussian Probability Densities

dated processes become non-Gaussian, which makes the optimal recursive estimation intractable in general.

The purpose of the optimal nonlinear filtering problem is to determine the *a posteriori* density function $p(\mathbf{x}_k | \mathbf{Y}_k)$. This density function provides the most complete description of the system information. However, the equations (1.6)~(1.7) for recursive filtering can only be solved exactly for a small class of problems such as linear and Gaussian cases because general nonlinear and/or non-Gaussian problems as shown in the above get involved in intractable infinite integrals. Therefore, the main emphasis of this dissertation in recursive nonlinear filtering theory is on proposing efficient and more accurate approximate nonlinear filters.

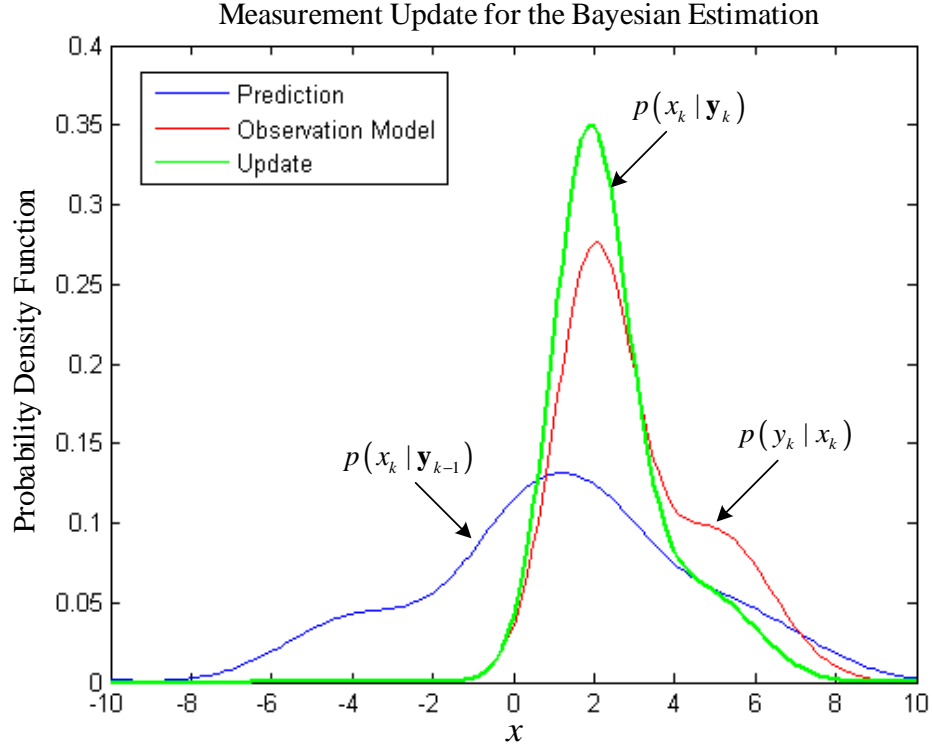


Fig. 1.3 Measurement Update Step for the Recursive Bayesian Estimation with Non-Gaussian Probability Densities

1.3 Review to Nonlinear Filtering

The recursive Bayesian relations were derived in the previous subsection. The solution to the recursive Bayesian estimation consists of a set of three integral equations, which are required to be solved each time step. There are classes of problems for which these equations are tractable. The most important of these classes is the set of problems where the state and observation equations are linear, and the distributions of the *prior*, and observation and system noise are Gaussian. In this case equations (1.6)~(1.7) can be solved to produce the Kalman filter.^{1,5} For other models the recursive relations are generally difficult to solve either in closed-form or numerically. Thus, approximate filtering solutions need to be obtained.

Historically the first of these approximate nonlinear filters was the extended Kalman filter (EKF),¹ which linearizes the system and observation equations about a single sample point with the assumption that the *a priori* distribution is Gaussian, and uses the Kalman filter to obtain estimates for the state and covariance of these estimates. The single sample point is chosen as the best estimate, that is, the approximation of the conditional mean.

There are two sources of inaccuracy in the EKF.²⁷ The first comes from the linearization of the nonlinear dynamic and/or measurement equations at each time step. The second results from the fact that the Gaussian assumption of the *priori* or *posterior* with the estimated mean and covariance will be false due to this linearization. The nonlinearities in the system model will result in non-Gaussian posterior and prior distributions at each time step, and the calculated mean and covariance matrix will be approximations to the true quantities. These problems have been considered and possible improvements were suggested by many scientists and engineers.^{28,29} In the truncated second-order nonlinear filter¹ the linearization errors can be reduced by retaining up to the second-order terms in the expansion of the system equations, whereas third and higher order central moments are neglected. Commonly, the assumption is also made that the *a posteriori* density is still Gaussian with the presence of the second-order terms. A similar procedure is used to derive the modified Gaussian second-order nonlinear filter,^{1,28} where nonlinearities are carried out up to fourth order, whereas fifth and higher central moments are neglected by assuming the Gaussian probability distribution (fourth moment is considered). However, it was shown that using van der Pol's equation the truncated second-order filter for this second-order system with linear observations turned out to be unstable. Henriksen³⁰ rederived the truncated second-order nonlinear filter and verified that the modified second-order filter is the correct form of what has been termed the truncated second-order filter,

provided a small correction is made in the discrete-time series.

Recently, there have been interesting developments in derivative-free nonlinear state estimation techniques as efficient alternatives to the extended Kalman filter.^{8,11,12} These include the unscented Kalman Filter (UKF), the central difference filter (CDF), and the divided difference filter (DDF). These are called sigma point filters (SPFs) and belong to the simulation based nonlinear filters.³¹ The UKF⁸ works on the principle that a set of discretely sampled sigma points can be used to parameterize the mean and covariance of the Gaussian random variables, and the posterior mean and covariance are propagated through the true nonlinear function without the linearization steps. The UKF has advantages over the EKF in that 1) it can lead to a more accurate, stable estimate of both the state and covariance, 2) the new filter can estimate with discontinuous functions, 3) no explicit derivation of the Jacobian or Hessian matrix is necessary, and 4) the new filter is suitable for parallel processing. The CDF¹¹ and the DDF¹² adopt an alternative linearization method called a central difference approximation¹⁰ in which derivatives are replaced by functional evaluations, and an easy expansion of the nonlinear functions to higher-order terms is possible. This accommodates easy and efficient implementation of the filters in nonlinear estimation applications.

A direct numerical approximation to the optimal nonlinear filter is to calculate the state conditional probability density by applying fast solvers to the Kushner-Stratonovich equation in the case of continuous-continuous system models³²⁻³⁴ or to the Fokker-Planck equation in continuous-discrete system models.³⁵⁻³⁸ Then, quantities of interest such as the minimum mean square error (MMSE), the covariance, or the maximum likelihood state estimate can then be constructed from the approximated posterior density. In continuous-discrete system models, the Fokker-Planck equation can be solved accurately and efficiently using finite difference schemes.^{13,14}

For discrete time measurement updates Bayes' formula can be applied recursively to combine measurement likelihoods derived from physical sensor models with the target state probability density.

To date, however, most of the presented nonlinear filtering methods (EKF, UKF, and DDF), which are based on local linearization of the nonlinear system equations or local approximation of the probability density of the state variables with the sampled sigma points, have yet been universally effective algorithms for dealing with both nonlinear and non-Gaussian system. As computing power increased, more computationally expensive filters were suggested. For nonlinear and/or non-Gaussian filtering problem, the Gaussian sum filter (GSF) which approximates the *a posteriori* density function by a weighted sum of Gaussian densities was proposed.^{39,40} It is also an extension of the EKF and copes with non-Gaussian noise in either or both of the system and observation equations. The key idea of the Gaussian sum filter is based on the principle that certain *a posteriori* densities can be approximated by a linear combination of Gaussian densities. This type of approach is quite reasonable when the posterior functions are multimodal densities. This mixture distribution method, however, suffers from the fact that for an accurate approximation of a non-Gaussian density a large number of densities may be required, which leads to heavy computational load in general.⁴¹

Similarly, a second general approach to nonlinear and non-Gaussian filtering is to evaluate the posterior density at a series of prescribed grid points in the sample space. Bucy⁴² suggested to use the error covariance matrix to establish the region and the grid. The grid point is centered at the mean value and the grid along each principal axis was chosen to extend over a distance sufficient to insure that the true state is contained in the grid region. The point masses that are the evaluated density values at each grid point are used to approximate the posterior density. Alternatively,

these grid points can be the basis for an approximation of the posterior by splines^{43,44} or by step functions.⁴⁵ The advantage of these approximations are that they simplify the integration involved in the recursive Bayesian solution. However, the number of grid points has to increase exponentially as the dimension increases, and the calculations at each grid point are non-trivial.⁴⁶

More recently, scientists and engineers have begun to pay attention to a new class of nonlinear/non-Gaussian filtering methods based on the sequential Monte Carlo approach since the appearance of the method, the bootstrap filter.^{15,16,22} The sequential Monte Carlo can be loosely defined as a simulation-based method that uses the Monte Carlo simulation method in order to solve on-line estimation and prediction problems. The sequential Monte Carlo approach is known as the bootstrap filtering,¹⁵ the condensation algorithm,¹⁹ and the particle filtering.²⁰ The flexible nature of the Monte Carlo simulations results in these methods often being more adaptive to some features of the target system.²¹ There have also been many recent modifications and improvements on the method.⁴⁷ This work investigates a number of improvements for particle filters, which universally effective algorithms for dealing with nonlinear and non-Gaussian system.

1.4 Contributions of the Dissertation

1. Unified Approaches to Nonlinear Filtering

- Various nonlinear filtering algorithms from the extended Kalman filter to novel, accurate, and theoretically better motivated algorithms such as the unscented filter, divided difference filter, finite difference filter, and particle filters are reviewed and interpreted in a unified way using the recursive Bayesian estimation. The examined nonlinear and/or non-Gaussian filter-

ing methods are applied to challenging engineering problems, and enhanced performance benefits are demonstrated with the applications.

2. Local Linearized Particle Filtering

- A new local linearized particle filtering algorithm called the divided difference particle filter (DDPF) is formulated. This method mitigates the sampling degeneracy phenomenon where most of particles have negligible weights after a few iterations.
 - Divided Difference Particle Filter

3. Adaptive Sigma Point Filters

- Practical new adaptive nonlinear filters for recursive estimation of the state and parameters of nonlinear systems with unknown noise statistics are presented. The adaptive nonlinear filters combine adaptive estimation techniques for system noise statistics with the nonlinear filters that include the unscented Kalman filter and divided difference filter. The purpose of the integrated filters is to not only compensate for the nonlinearity effects neglected from linearization by utilizing nonlinear filters, but also to take into account the system modeling errors by adaptively estimating the noise statistics and unknown parameters.
 - Adaptive Unscented Kalman Filter
 - Adaptive Divided Difference Filter

4. Applications

- For qualitative and quantitative performance analysis among the proposed nonlinear filters, systematic methods for measuring the nonlinearities and

optimality of the proposed filters are introduced. The proposed nonlinear optimal and sub-optimal filtering algorithms with applications to spacecraft orbit estimation, autonomous GPS navigation, and robot navigation are investigated.

1.5 Dissertation Outline

Chapter 2 reviews the current literature on linear filtering from the least-squares method to the Kalman filter. The linear filters are interpreted by using the Bayesian recursive structure.

Chapter 3 discusses the approximate nonlinear filters from the point of view of the Bayesian relations. The approximate nonlinear filters include the extended Kalman filter, the unscented Kalman filter, the divided difference filter, and the finite difference filter. The advantages and drawbacks of each filter will be discussed.

In Chapter 4, first the particle filter, known as the SIR or bootstrap filter, is introduced for nonlinear and/or non-Gaussian filtering problems. Many modifications and improvements on the particle filter are also investigated. These include the locally linearized particle filters and regularized particle filters that improve the performance of the standard particle filter and mitigate the sample degeneracy phenomenon and sample impoverishment, respectively.

In Chapter 5 the existing linear adaptive filters are reviewed, and new adaptive nonlinear filtering algorithms are derived by integrating the presented nonlinear filtering with an innovation based adaptive process noise compensation.

In Chapter 6, the application of dynamic model compensation (DMC) is discussed as an adaptive filtering scheme. DMC improves the performance of the filter, but it requires tuning in order to determine the optimal values for the parameters

that determine the characteristics of the stochastic acceleration function.

Chapter 7 explains the procedures and mathematical algorithms for the orbit estimation and navigation applications. Guidelines for the construction of the discrete-time process covariance matrix that is necessary for the numerical simulation of the discrete-time system is derived.

In chapter 8, simulation results and future works are discussed. The proposed nonlinear optimal and sub-optimal filtering algorithms with applications to spacecraft orbit estimation, autonomous GPS navigation, and robot navigation are investigated. A number of different nonlinear filtering approaches for each simulation example are demonstrated.

Finally, chapter 9 presents some conclusions about the proposed nonlinear/adaptive filters and directions for future research are indicated.

CHAPTER II

OPTIMAL LINEAR FILTERING

This chapter presents the optimal linear estimation algorithms from the batch filter to the Kalman filter. Both batch least-squares and statistical Bayesian recursive estimation methods are illustrated.

2.1 Batch Least-Squares Estimation

The least-squares (LS) is based on a set of linear measurements of unknown constant parameters. The purpose of the LS is to estimate the state vector \mathbf{x} modeled as an unknown vector from the noisy observations

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (2.1)$$

where the measurement noise $\mathbf{v} \in \Re^{m \times 1}$ has zero-mean and covariance $\mathbf{R} \in \Re^{m \times m}$, and $\mathbf{H} \in \Re^{m \times n}$ is the linear mapping between the measurement $\mathbf{y} \in \Re^{m \times 1}$ and state vector $\mathbf{x} \in \Re^{n \times 1}$.

When the measurement set contains redundant information, that is, there are more independent equations than unknowns ($m > n$), according to the *principle of least-squares*,⁷ the optimal estimate $\hat{\mathbf{x}}$ of the unknown parameter \mathbf{x} is obtained by minimizing the objective function J of the sum of the squares of the residual errors \mathbf{e} , which is the difference between the true \mathbf{y} and estimated $\hat{\mathbf{y}}$ observations

$$J = \frac{1}{2} \sum_{j=1}^m e_j^2 \quad (2.2)$$

The residual error e_j is defined by

$$e_j \equiv y_j - \hat{y}_j = y_j - \sum_{i=1}^n h_i(t_j) \hat{x}_i \quad (2.3)$$

The objective function J can be rewritten in a vector/matrix form

$$J = \frac{1}{2} \mathbf{e}^T \mathbf{e} = \frac{1}{2} [\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}]^T [\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}] \quad (2.4)$$

The LS estimator that minimizes the cost function is obtained by setting its gradient with respect to $\hat{\mathbf{x}}$ to zero

Necessary Condition

$$\nabla_{\hat{\mathbf{x}}} J = \frac{\partial J}{\partial \hat{\mathbf{x}}} = \mathbf{H}^T \mathbf{H} \hat{\mathbf{x}} - \mathbf{H}^T \mathbf{y} = 0 \quad (2.5)$$

Sufficient Condition

$$\nabla_{\hat{\mathbf{x}}}^2 J \equiv \frac{\partial^2 J}{\partial \hat{\mathbf{x}} \partial \hat{\mathbf{x}}^T} = \mathbf{H}^T \mathbf{H} > 0 \quad (2.6)$$

where $\nabla_{\hat{\mathbf{x}}} J$ is the *Jacobian* and $\nabla_{\hat{\mathbf{x}}}^2 J$ is the *Hessian*. From the necessary condition, the *normal* equation is obtained

$$(\mathbf{H}^T \mathbf{H}) \hat{\mathbf{x}} = \mathbf{H}^T \mathbf{y} \quad (2.7)$$

If the required inverse of $\mathbf{H}^T \mathbf{H}$ exists, then the optimal estimate of the unknown parameter is obtained by

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y} \quad (2.8)$$

It is seen that if the Hessian of Eq. (2.6) is positive definite (p.d.) with respect to $\hat{\mathbf{x}}$ the extremum of the optimal estimate is a minimum.

2.2 Weighted Least-Squares Estimation

The least-squares (LS) estimator in Eq. (2.8) places equal weight on each measurement \mathbf{y} . In real applications, however, each measurement has different precision, and possibly, different units. In this situation a weighted least-squares (WLS) is appro-

priate for the estimation of an unknown constant parameter. Now, the objective function is represented in terms of a vector/matrix form⁷

$$J = \frac{1}{2} \mathbf{e}^T \mathbf{W} \mathbf{e} = \frac{1}{2} [\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}]^T \mathbf{W} [\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}] \quad (2.9)$$

where the weight matrix \mathbf{W} is selected to be the inverse of the covariance matrix of the measurement noise \mathbf{R}

$$\mathbf{W} = \mathbf{R}^{-1} \quad (2.10)$$

and

$$\mathbf{R} = \begin{bmatrix} R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_k \end{bmatrix} \quad (2.11)$$

The extremum conditions are obtained by setting its gradient with respect to $\hat{\mathbf{x}}$ to zero

Necessary Condition

$$\nabla_{\hat{\mathbf{x}}} J = \frac{\partial J}{\partial \hat{\mathbf{x}}} = \mathbf{H}^T \mathbf{W} \mathbf{H} \hat{\mathbf{x}} - \mathbf{H}^T \mathbf{W} \mathbf{y} = 0 \quad (2.12)$$

Sufficient Condition

$$\nabla_{\hat{\mathbf{x}}}^2 J \equiv \frac{\partial^2 J}{\partial \hat{\mathbf{x}} \partial \hat{\mathbf{x}}^T} = \mathbf{H}^T \mathbf{W} \mathbf{H} > 0 \quad (2.13)$$

From the necessary condition, the optimal estimate is obtained by

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbf{y} \quad (2.14)$$

It is seen from Eq. (2.13) that the weight matrix \mathbf{W} should be positive definite.

2.2.1 Relationship to Maximum Likelihood Estimation

The least-squares (LS) estimator does not make any probabilistic interpretation, but it is derived from a deterministic point of view. Consequently, the LS estimator may be preferred to other estimators when there is no information for the probability density functions of \mathbf{x}_k and \mathbf{y}_k . Alternatively, if the measurement errors v_i are independent Gaussian random variables with mean zero and covariance R_{ii} , then minimizing the LS criterion in Eq. (2.9) is equivalent to maximizing the likelihood (ML) function^{7,27}

$$\Lambda_k(\hat{\mathbf{x}}) = p(\mathbf{y}_k|\hat{\mathbf{x}}) = \prod_{i=1}^k p[y_i|\hat{\mathbf{x}}] \quad (2.15)$$

$$= c \exp \left\{ -\frac{1}{2} \sum_{i=1}^k [y_i - \mathbf{h}_i \hat{\mathbf{x}}]^T R_i^{-1} [y_i - \mathbf{h}_i \hat{\mathbf{x}}] \right\} \quad (2.16)$$

where $c > 0$ is a constant parameter, and \mathbf{y}_k is the stacked vector of measurement up to the current time k

$$\mathbf{y}_k = [y_1, y_2, \dots, y_k]^T \quad (2.17)$$

In order to maximize the likelihood function $p(\mathbf{y}_k|\hat{\mathbf{x}})$, the component in the bracket of the exponent should be minimized. This is equivalent to minimizing the cost function in Eq. (2.9). Therefore, the least-squares and maximum likelihood estimators coincide in this case where the ML estimator maximizes the likelihood function under the Gaussian assumption with zero mean and covariance R_i of the measurement errors.

2.2.2 Relationship to Bayesian Estimation

When the statistical models for the probability density functions of \mathbf{x} and \mathbf{y} are available, the Bayesian estimation can be another approach. Then, the estimation problem is changed to seeking the *a posteriori* conditional density function $p(\mathbf{x}|\mathbf{y})$, which contains all the statistical information for the mean and covariance values. The

conditional density $p(\mathbf{x}|\mathbf{y})$ is evaluated by employing the Bayes's theorem given in Eq. (1.9)

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \quad (2.18)$$

where $p(\mathbf{x})$ is the *a priori* probability density function of \mathbf{x} , $p(\mathbf{y})$ is the probability density function of the measurements, and $p(\mathbf{y}|\mathbf{x})$ is the likelihood function.

The estimate $\hat{\mathbf{x}}$ can be computed from $p(\mathbf{x}|\mathbf{y})$, depending on the criteria of the optimality. In other words, the Bayesian estimation is based on the minimization of the risk function \mathcal{J} which consists of a cost function $J(\Delta\mathbf{x})$ ²⁷

$$\mathcal{J}(\hat{\mathbf{x}}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J(\Delta\mathbf{x})p(\mathbf{x}, \mathbf{y}) d\mathbf{x}d\mathbf{y} \quad (2.19)$$

where $p(\mathbf{x}, \mathbf{y})$ is the joint probability density function of the random variables \mathbf{x} and \mathbf{y} , and the cost function $J(\Delta\mathbf{x})$ is a function of the estimation error $\Delta\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}}$. Different approaches for solving this problem depend on the choice of the cost function $J(\Delta\mathbf{x})$

The minimum variance estimate minimizes the risk function with the cost function

$$J(\Delta\mathbf{x}) = \Delta\mathbf{x}^T \mathbf{W} \Delta\mathbf{x} \quad (2.20)$$

where \mathbf{W} is a positive, symmetric matrix. The minimum of the risk function \mathcal{J} is found for the value of $\hat{\mathbf{x}}_{mv}$

$$\hat{\mathbf{x}}_{mv} = E\{\mathbf{x}|\mathbf{y}\} \quad (2.21)$$

Then, the conditional mean estimate $\hat{\mathbf{x}}$ is evaluated by

$$\hat{\mathbf{x}} = E\{\mathbf{x}|\mathbf{y}\} = \int \mathbf{x}p(\mathbf{x}|\mathbf{y})d\mathbf{x} \quad (2.22)$$

Assuming the distributions for \mathbf{x} and \mathbf{v} are Gaussian, the conditional mean value $\hat{\mathbf{x}}$

is evaluated by⁷

$$\hat{\mathbf{x}} = (\mathbf{P}_0^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{y} \quad (2.23)$$

where \mathbf{P}_0 is the *a priori* covariance matrix of \mathbf{x} .

Note that if there is no *a priori* information, the above equation reduces to the weighted least-squares solution in Eq. (2.14). If all measurement errors are uncorrelated (i.e., \mathbf{R} is a diagonal matrix) and they have equal variance (i.e., $\mathbf{R} = \sigma^2 \mathbf{I}$) Eq. (2.23) becomes the LS solution in Eq. (2.8).⁷

2.2.3 Unbiased Estimator

If the measurement errors v_i are uncorrelated, zero-mean random variables with covariance R_i , then the LS estimator is *unbiased*

$$E \{ \hat{\mathbf{x}} \} = [\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{R}^{-1} E \{ \mathbf{H} \mathbf{x} + \mathbf{v} \} = \mathbf{x} \quad (2.24)$$

The estimation error $\Delta \mathbf{x}$ becomes

$$\Delta \mathbf{x} = \mathbf{x} - \hat{\mathbf{x}} = - [\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{v}) \quad (2.25)$$

The covariance matrix of the LS estimator is computed by

$$\begin{aligned} \mathbf{P} &\equiv E \left\{ [\hat{\mathbf{x}} - \mathbf{x}] [\hat{\mathbf{x}} - \mathbf{x}]^T \right\} \\ &= E \left\{ \Delta \mathbf{x} \Delta \mathbf{x}^T \right\} \\ &= [\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{R}) \mathbf{R}^{-1} \mathbf{H} [\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \end{aligned} \quad (2.26)$$

After manipulations of Eq. (2.26), the covariance matrix becomes

$$\mathbf{P} = [\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \quad (2.27)$$

2.3 Linear Sequential Estimation

In a batch least-squares (BLS) estimator the measurements are available in simultaneous processing. In this section, however, it is assumed that the measurements are taken in a sequential way such that new estimates are computed by utilizing all previous measurements including the current data.²⁷

Consider two subsets of observations

$$\mathbf{y}_1 = \mathbf{H}_1 \mathbf{x} + \mathbf{v}_1 \quad (2.28)$$

$$\mathbf{y}_2 = \mathbf{H}_2 \mathbf{x} + \mathbf{v}_2 \quad (2.29)$$

where the measurement vectors are

$$\mathbf{y}_1 = [y_{11}, y_{12}, \dots, y_{1m_1}]^T \in \Re^{m_1 \times 1} \quad (2.30)$$

$$\mathbf{y}_2 = [y_{21}, y_{22}, \dots, y_{2m_2}]^T \in \Re^{m_2 \times 1} \quad (2.31)$$

and the linear mappings are $\mathbf{H}_1 \in \Re^{m_1 \times n}$ and $\mathbf{H}_2 \in \Re^{m_2 \times n}$, respectively.

The least-squares estimates $\hat{\mathbf{x}}$ of the unknown parameter \mathbf{x} based on the first measurement subset is obtained by

$$\hat{\mathbf{x}}_1 = (\mathbf{H}_1^T \mathbf{W}_1 \mathbf{H}_1)^{-1} \mathbf{H}_1^T \mathbf{W}_1 \mathbf{y}_1 \quad (2.32)$$

where \mathbf{W}_1 is an $m_1 \times m_1$ symmetric, positive definite matrix associated with measurement noise \mathbf{v}_1 . Now, consider both measurement subsets \mathbf{y}_1 and \mathbf{y}_2 simultaneously in partitioned forms

$$\mathbf{y} = \mathbf{H} \mathbf{x} + \mathbf{v} \quad (2.33)$$

where

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$$

and the stacked weight matrix is given in block diagonal structure

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 \end{bmatrix}$$

The optimal estimate based on the first two measurements subsets are obtained by

$$\hat{\mathbf{x}}_2 = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbf{y} \quad (2.34)$$

The optimal estimate in Eq. (2.34) can be expanded by using the block diagonal of the weight matrix \mathbf{W}

$$\hat{\mathbf{x}}_2 = [\mathbf{H}_1^T \mathbf{W}_1 \mathbf{H}_1 + \mathbf{H}_2^T \mathbf{W}_2 \mathbf{H}_2]^{-1} (\mathbf{H}_1^T \mathbf{W}_1 \mathbf{y}_1 + \mathbf{H}_2^T \mathbf{W}_2 \mathbf{y}_2) \quad (2.35)$$

For further compact formulation, the following variables are defined

$$\mathbf{P}_1 \equiv [\mathbf{H}_1^T \mathbf{W}_1 \mathbf{H}_1]^{-1} \quad (2.36)$$

$$\mathbf{P}_2 \equiv [\mathbf{H}_1^T \mathbf{W}_1 \mathbf{H}_1 + \mathbf{H}_2^T \mathbf{W}_2 \mathbf{H}_2]^{-1} \quad (2.37)$$

Then, the covariance \mathbf{P}_2 has the following relationship with \mathbf{P}_1

$$\mathbf{P}_2^{-1} = \mathbf{P}_1^{-1} + [\mathbf{H}_2^T \mathbf{W}_2 \mathbf{H}_2]^{-1} \quad (2.38)$$

Finally, after manipulation, the optimal estimate $\hat{\mathbf{x}}_2$ based on the previous estimate $\hat{\mathbf{x}}_1$ is calculated by

$$\hat{\mathbf{x}}_2 = \hat{\mathbf{x}}_1 + \mathcal{K}_2(\mathbf{y}_2 - \mathbf{H}_2 \hat{\mathbf{x}}_1) \quad (2.39)$$

where

$$\mathcal{K}_2 \equiv \mathbf{P}_2 \mathbf{H}_2^T \mathbf{W}_2 \quad (2.40)$$

Now, general recursive least-squares estimation that uses the k th estimate to

determine the estimate at $k + 1$ leads to the sequential formula⁴⁸

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + \mathcal{K}_{k+1} (\mathbf{y}_{k+1} - \mathbf{H}_{k+1} \hat{\mathbf{x}}_k) \quad (2.41)$$

where

$$\mathcal{K}_{k+1} = \mathbf{P}_{k+1} \mathbf{H}_{k+1}^T \mathbf{W}_{k+1} \quad (2.42)$$

$$\mathbf{P}_{k+1}^{-1} = \mathbf{P}_k^{-1} + [\mathbf{H}_{k+1}^T \mathbf{W}_{k+1} \mathbf{H}_{k+1}]^{-1} \quad (2.43)$$

Eq. (2.41) updates the previous estimate $\hat{\mathbf{x}}_k$ by utilizing the current measurement information \mathbf{y}_{k+1} , which is known as *Kalman update* process. The parameter value \mathcal{K}_{k+1} is the *Kalman gain* matrix.

The inverse covariance matrix \mathbf{P}_{k+1}^{-1} known as the Fisher information matrix can be rewritten by using the matrix *matrix inversion lemma*

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{P}_k \mathbf{H}_{k+1}^T [\mathbf{H}_{k+1} \mathbf{P}_k \mathbf{H}_{k+1}^T + \mathbf{W}_{k+1}^{-1}]^{-1} \mathbf{H}_{k+1} \mathbf{P}_k \quad (2.44)$$

The update equation can be also rearranged in alternative forms. First, the *Kalman gain* equation can be rewritten by substituting Eq. (2.44)

$$\mathcal{K}_{k+1} = \mathbf{P}_k \mathbf{H}_{k+1}^T [\mathbf{H}_{k+1} \mathbf{P}_k \mathbf{H}_{k+1}^T + \mathbf{W}_{k+1}^{-1}]^{-1} \quad (2.45)$$

Now, the covariance update equation can be expressed in terms of the *Kalman gain* matrix

$$\mathbf{P}_{k+1} = [\mathbf{I} - \mathcal{K}_{k+1} \mathbf{H}_{k+1}] \mathbf{P}_k \quad (2.46)$$

Let's denote the covariance of the residual as

$$\mathbf{S}_{k+1} \equiv \mathbf{H}_{k+1} \mathbf{P}_k \mathbf{H}_{k+1}^T + \mathbf{W}_{k+1}^{-1} \quad (2.47)$$

Table 2.1 Sequential Least-Squares (SLS) Algorithm

Initialization:

$$\hat{\mathbf{x}}_k = (\mathbf{H}_1^T \mathbf{W}_1 \mathbf{H}_1)^{-1} \mathbf{H}_1^T \mathbf{W}_1 \mathbf{y}_1$$

$$\mathbf{P}_k = [\mathbf{H}_1^T \mathbf{W}_1 \mathbf{H}_1]^{-1}$$

Innovation Covariance:

$$\mathbf{S}_{k+1} = \mathbf{H}_{k+1} \mathbf{P}_k \mathbf{H}_{k+1}^T + \mathbf{W}_{k+1}^{-1}$$

Update:

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + \mathcal{K}_{k+1} (\mathbf{y}_{k+1} - \mathbf{H}_{k+1} \hat{\mathbf{x}}_k)$$

$$\mathcal{K}_{k+1} = \mathbf{P}_k \mathbf{H}_{k+1}^T \mathbf{S}_{k+1}^{-1}$$

$$\mathbf{P}_{k+1} = [\mathbf{I} - \mathcal{K}_{k+1} \mathbf{H}_{k+1}] \mathbf{P}_k$$

where \mathcal{K}_{k+1} = Kalman gain matrix, \mathbf{W}_{k+1} = Measurement error matrix

which leads to the compact form of the *Kalman gain* equation

$$\mathcal{K}_{k+1} = \mathbf{P}_k \mathbf{H}_{k+1}^T \mathbf{S}_{k+1}^{-1} \quad (2.48)$$

Finally, the algorithms of the linear sequential estimation are summarized in Table 2.1.

2.4 Kalman Filter

2.4.1 Introduction

The Kalman filter has been the subject of extensive research and application, particularly in the area of orbit determination and autonomous GPS navigation, since the

publication of Kalman's famous paper⁵ describing a recursive solution to the filtering problem.

The Kalman filter is a set of mathematical equations that provides an efficient recursive solution of the least-squares method. It provides estimates of the past, present, and also future states, and it can do so when the precise nature of the modeled system is unknown, for example, when the modeling errors of the system model are not known well.

A major characteristic of the batch least-squares estimator is that the estimated state estimate and covariance matrix are based on processing a batch of data spread over some time interval. A second characteristic is that the estimate is involved with a particular epoch. Therefore, we expect the state vector and the covariance matrix to be predicted from the epoch time to a new time. During this processing, one specific problem is how to propagate the state and covariance matrix over the time interval to provide statistical information at the new epoch.

The Kalman filter is a technique for computing the best estimate of the state of a time varying process with imperfect observations and an uncertain dynamic model. In other words, it provides the minimum variance estimate of the state based on statistical information about the dynamic model and the observations. It differs from the least-squares technique in three very important and essential approaches.⁷ First, it continuously updates the epoch time, thus estimating the state at each epoch successive observation time. Second, it carries all the information concerning past measurements in its current state and covariance estimates, and, therefore, doesn't need to reprocess all of the past measurement information at each time step. Finally, the Kalman filter can cope with more realistic noises, whereas usually least-squares methods can not deal with noises that depend on the mathematical models of the dynamic systems.

In this section, the procedures for the Kalman filter algorithms are described in terms of the recursive Bayesian approach.

2.4.2 Kalman Filtering Algorithm

The following linear discrete-time system equations are considered

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k \quad (2.49)$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (2.50)$$

where $\mathbf{F} \in \Re^{n \times n}$ is a dynamic model, $\mathbf{x}_k \in \Re^{n \times 1}$ is the state vector, $\mathbf{H} \in \Re^{m \times n}$ is a measurement output model, and $\mathbf{y}_k \in \Re^{m \times 1}$ is the observation vector. It is assumed that the noise vectors are stationary, white Gaussian processes with the zero-mean and covariance

$$E[\mathbf{w}_k \mathbf{w}_j^T] = \delta_{kj} \mathbf{Q}_k, \quad E[\mathbf{v}_k \mathbf{v}_j^T] = \delta_{kj} \mathbf{R}_k, \quad E[\mathbf{v}_k \mathbf{w}_j^T] = 0, \quad \forall k, j \quad (2.51)$$

The system and observational errors are assumed to be uncorrelated. The Kalman filtering algorithm is represented by a two-step recursive process, *prediction* and *update*.¹

From the Bayesian approach in Eqs. (1.6)~(1.7), the recursive structure is described by the current state density as a function of the previous density and the most recent measurements. The dynamic and measurement models play a role in determining the state transition probability $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and measurement likelihood function $p(\mathbf{y}_k|\mathbf{x}_k)$. Specifically, the state transition density is computed by the state space model in Eq. (2.49), and the additive Gaussian process noise $p(\mathbf{w}_k) = \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$. Thus, the state transition probability $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is obtained by

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1}, \mathbf{Q}_k) \quad (2.52)$$

Similarly, the likelihood function $p(\mathbf{y}_k|\mathbf{x}_k)$ is determined by the observation model, and the measurement noise density $p(\mathbf{v}_k) = \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$

$$p(\mathbf{y}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{y}_k; \hat{\mathbf{y}}_k, \mathbf{R}_k) \quad (2.53)$$

The recursive relationships in Eqs. (1.6)~(1.7) are given by³

$$p(\mathbf{x}_k|\mathbf{Y}_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k, \mathbf{P}_k) \quad (2.54)$$

$$p(\mathbf{x}_{k+1}|\mathbf{Y}_k) = \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1}^-, \mathbf{P}_k^-) \quad (2.55)$$

$$p(\mathbf{x}_{k+1}|\mathbf{Y}_{k+1}) = \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1}, \mathbf{P}_{k+1}) \quad (2.56)$$

where $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{P})$ denotes a Gaussian density with argument \mathbf{x} , mean \mathbf{m} , and covariance \mathbf{P} expressed by

$$\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{P}) \equiv |2\pi\mathbf{P}|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m}) \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m})^T \right\} \quad (2.57)$$

If it is assumed that all densities remain Gaussian, then the Bayesian recursion can be simplified in terms of only the conditional mean $\hat{\mathbf{x}}_k = E\{\mathbf{x}_k|\mathbf{Y}^k\}$ and covariance $\mathbf{P}_k = E\{\Delta\mathbf{x}_k\Delta\mathbf{x}_k^T|\mathbf{Y}^k\}$. More details are found in Ref. 3. The optimal components in the recursion estimation are given by

$$\hat{\mathbf{x}}_{k+1}^- = E\{\mathbf{F}\mathbf{x}_k + \mathbf{w}_k|\mathbf{Y}^k\} \quad (2.58)$$

$$\hat{\mathbf{y}}_{k+1}^- = E\{\mathbf{H}\hat{\mathbf{x}}_{k+1}^- + \mathbf{v}_{k+1}|\mathbf{Y}^k\} \quad (2.59)$$

The state prediction in Eq. (2.58) can be represented by

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{F}\hat{\mathbf{x}}_k \quad (2.60)$$

where $\hat{\mathbf{x}}_{k+1}^-$ is the model prediction and $\hat{\mathbf{x}}$ is the currently estimated state. The estimate $\hat{\mathbf{x}}_{k+1}^+$ of the true state \mathbf{x}_{k+1} is obtained by combining the observations \mathbf{y}_{k+1}

and the model predictions $\hat{\mathbf{x}}_{k+1}^-$

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathcal{K}_{k+1} \mathbf{v}_{k+1} \quad (2.61)$$

where \mathbf{v}_{k+1} is the innovation vector, which is equal to the difference between the actual and the predicted observations

$$\mathbf{v}_{k+1} \equiv \mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^- = \mathbf{y}_{k+1} - \mathbf{H} \hat{\mathbf{x}}_{k+1}^- \quad (2.62)$$

The predicted and updated equations for the state covariance matrix are computed by

$$\mathbf{P}_{k+1}^- = \mathbf{F} \mathbf{P}_k \mathbf{F}^T + \mathbf{Q}_k \quad (2.63)$$

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathcal{K}_{k+1} \mathbf{P}_{k+1}^{vv} \mathcal{K}_{k+1}^T \quad (2.64)$$

where the covariance of the innovation vector is given by

$$\mathbf{P}_{k+1}^{vv} = \mathbf{H} \mathbf{P}_{k+1}^- \mathbf{H}^T + \mathbf{R}_{k+1} \quad (2.65)$$

The *Kalman gain* \mathcal{K}_{k+1} is computed by

$$\mathcal{K}_{k+1} = \mathbf{P}_{k+1}^{xy} (\mathbf{P}_{k+1}^{vv})^{-1} \quad (2.66)$$

where \mathbf{P}_{k+1}^{xy} is the predicted cross-correlation matrix between $\hat{\mathbf{x}}_{k+1}^-$ and $\hat{\mathbf{y}}_{k+1}^-$

$$\mathbf{P}_{k+1}^{xy} = \mathbf{P}_{k+1}^- \mathbf{H}^T \quad (2.67)$$

The Bayesian relations of the predictor-corrector structure for the Kalman filtering algorithm can be represented by the block diagram as shown in Fig. 2.1 and the specific algorithms are summarized in Table 2.2 with detail.

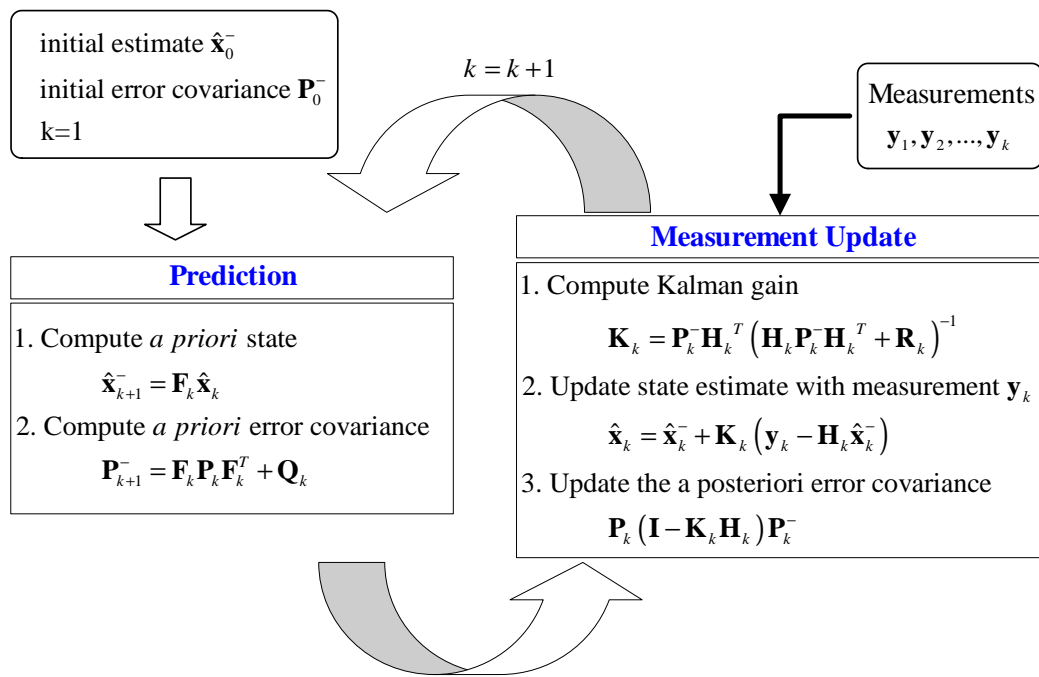


Fig. 2.1 Diagram of Predictor-Corrector Form of the Kalman Filter

Table 2.2 Kalman Filter (KF) Algorithm

Initialization:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$$

$$\mathbf{P}_0 = E \left[(\mathbf{x}_0 - \hat{\mathbf{x}}_0) (\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T \right]$$

State Propagation:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{F}_k \hat{\mathbf{x}}_k$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_k$$

Observation Propagation:

$$\hat{\mathbf{y}}_{k+1}^- = \mathbf{H}_{k+1} \hat{\mathbf{x}}_{k+1}^-$$

$$\mathbf{P}_{k+1}^{vv} = \mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1}$$

$$\mathbf{P}_{k+1}^{xy} = \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T$$

Update:

$$\mathcal{K}_{k+1} = \mathbf{P}_{k+1}^{xy} (\mathbf{P}_{k+1}^{vv})^{-1}$$

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathcal{K}_{k+1} \mathbf{P}_{k+1}^{vv} \mathcal{K}_{k+1}^T$$

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathcal{K}_{k+1} \mathbf{v}_{k+1}$$

where $(-)$ denotes a “propagated” value, $(+)$ denotes a “updated” value

\mathbf{Q}_k = System Model Error Matrix, \mathbf{R}_k = Measurement Error Matrix

CHAPTER III

SUBOPTIMAL NONLINEAR FILTERING

In linear systems an optimal, closed form solution exists,^{3,5} but when the systems are nonlinear there are restrictions on the analytical forms of the optimal solutions,⁷ i.e., multi-dimensional integration. Therefore, approximate nonlinear filters have been proposed.^{1,4,7} These approximate nonlinear filters can be categorized into four broad types: (1) analytical approximations, (2) direct numerical approximations, (3) sampling-based approaches, and (4) Gaussian sum filters (or multiple model filters). The most widely used analytical approximate nonlinear filter is the extended Kalman filter. In this section, three types of approximate nonlinear filters, analytical, sampling-based, and direct numerical approximate filters, are investigated along with the nonlinear least-squares estimation.

3.1 Nonlinear Least-Squares Estimation

The purpose of the least-squares process is to minimize a cost function that depends on the actual and computed observations that depend on the dynamic model and the initial state. For this problem we assume a continuous system model with errors modeled by white-Gaussian noise $\mathbf{w}(t)$, and discrete time measurements corrupted by white-Gaussian noise \mathbf{v}_k . The continuous-discrete nonlinear equations are given by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, t) + \mathbf{w}(t) \quad (3.1)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k \quad (3.2)$$

where $\mathbf{x}_k \in \Re^n$ is the $n \times 1$ state vector, $\mathbf{y}_k \in \Re^m$ is the $m \times 1$ observation vector. $\mathbf{w}(t) \in \Re^q$ is the $q \times 1$ state noise process vector and $\mathbf{v}_k \in \Re^s$ is the $s \times 1$ additive mea-

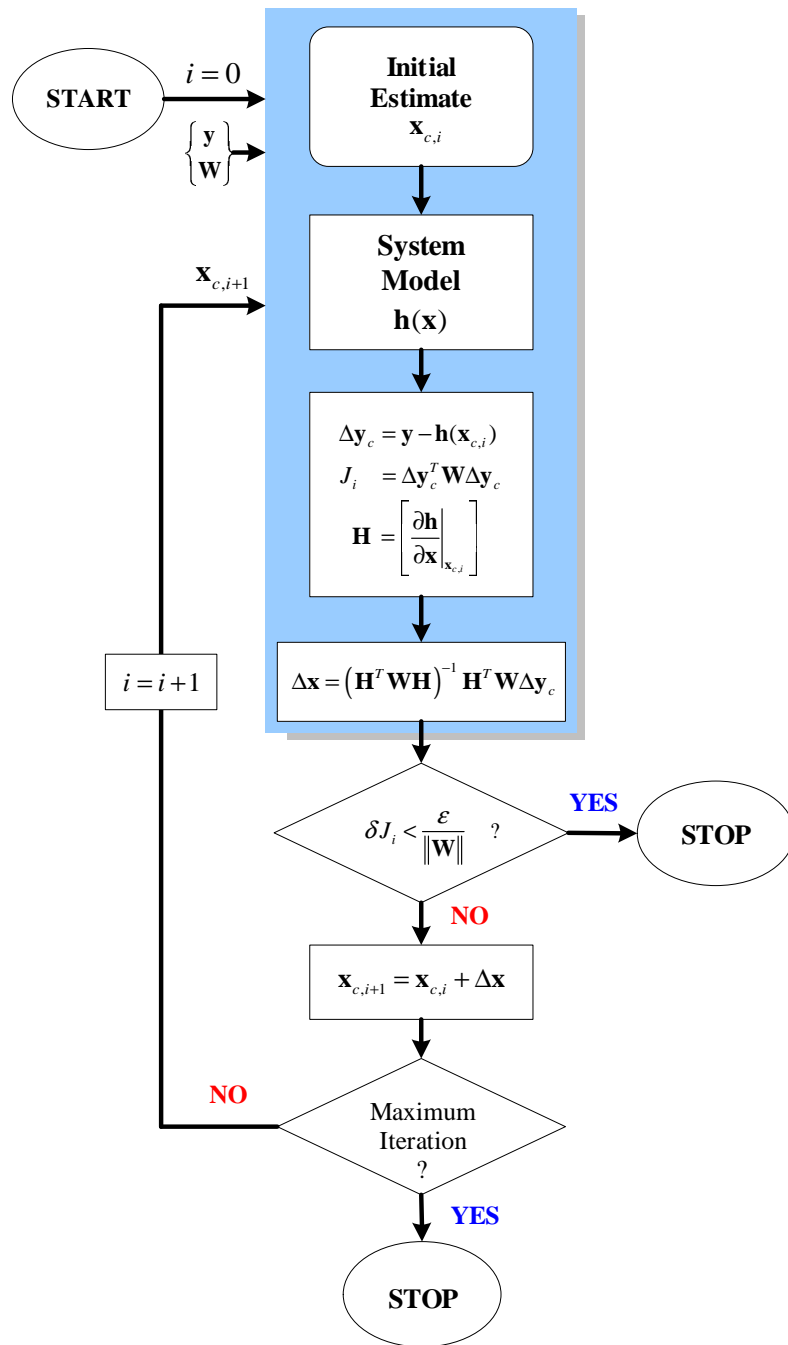


Fig. 3.1 Nonlinear Least-Squares Differential Correction

surement noise vector. It is assumed that the noise vectors are zero-mean Gaussian processes

$$E [\mathbf{w}(t)\mathbf{w}^T(s)] = \delta(t-s)\mathbf{Q}(t), \quad E [\mathbf{v}_k\mathbf{v}_j^T] = \delta_{kj}\mathbf{R}_k \quad (3.3)$$

The basic principle of the method of least-squares is that the best estimate of the state is the estimate which minimizes the sum of the squares of the residuals. The residual is defined as

$$\mathbf{v}_k = \mathbf{y}_k - \mathbf{h}(\mathbf{x}_k, k) \quad (3.4)$$

where \mathbf{y}_k are the actual observations or true observations. If the nominal trajectory of the system is $\hat{\mathbf{x}}_k$, then the measurement function $\mathbf{h}(\mathbf{x}_k, k)$ can be approximated by using the Taylor-series expansion

$$\mathbf{h}(\mathbf{x}_k) = \mathbf{h}(\hat{\mathbf{x}}_k) + \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_k} (\mathbf{x}_k - \hat{\mathbf{x}}_k) + \text{H.O.T} \quad (3.5)$$

where $\mathbf{h}(\hat{\mathbf{x}}_k)$ is the estimated value of the observation at the value $\hat{\mathbf{x}}_k$, and the gradient matrix \mathbf{H}_k , also known as the Jacobian matrix, is defined as

$$\mathbf{H}_k \equiv \left[\left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k} \right] \quad (3.6)$$

Assume that the current estimates of the state \mathbf{x}_c are denoted by

$$\mathbf{x}_c = [x_{1c}, x_{2c}, \dots, x_{nc}]^T \quad (3.7)$$

and they are related to the estimates $\hat{\mathbf{x}}_k$ by an unknown set of corrections $\Delta \mathbf{x}$

$$\hat{\mathbf{x}}_k = \mathbf{x}_c + \Delta \mathbf{x} \quad (3.8)$$

If the components of the corrections $\Delta \mathbf{x}$ are sufficiently small, it may be possible to solve for an approximation to the corrections and update \mathbf{x}_c with an improved estimate of $\hat{\mathbf{x}}_k$ using Eq. (3.8). With this assumption, the function $\mathbf{h}(\hat{\mathbf{x}}_k, k)$ can be

linearized about \mathbf{x}_c

$$\mathbf{h}(\hat{\mathbf{x}}_k) \approx \mathbf{h}(\mathbf{x}_c) + \mathbf{H}\Delta\mathbf{x} \quad (3.9)$$

The measurement residual after the correction can be linearly approximated by

$$\Delta\mathbf{y} \equiv \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k) \approx \mathbf{y}_k - \mathbf{h}(\mathbf{x}_c) - \mathbf{H}\Delta\mathbf{x} \quad (3.10)$$

where the differential correction is $\Delta\mathbf{x} = \hat{\mathbf{x}}_k - \mathbf{x}_c$, and the residual before the correction is defined by

$$\Delta\mathbf{y}_c \equiv \mathbf{y}_k - \mathbf{h}(\mathbf{x}_c) \quad (3.11)$$

The objective of the weighted Least-Squares estimation is to minimize the weighted sum of the squares of the measurement residuals given by the cost function J ⁴⁸

$$J = \frac{1}{2} \Delta\mathbf{y}^T \mathbf{W} \Delta\mathbf{y} = \frac{1}{2} [\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k)]^T \mathbf{W} [\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k)] \quad (3.12)$$

where \mathbf{W} is an $m \times m$ symmetric weighting matrix used to weight the relative importance of each measurement. The local strategy for determining the differential corrections $\Delta\mathbf{x}$ is to select the particular corrections that lead to the minimum sum of squares of the linearly predicted residuals J_p . The measurement residual can be approximated in terms of $\Delta\mathbf{y}_c$ by using Eq. (3.10), and the cost function is rewritten by

$$J_p \equiv \frac{1}{2} [\Delta\mathbf{y}_c - \mathbf{H}\Delta\mathbf{x}]^T \mathbf{W} [\Delta\mathbf{y}_c - \mathbf{H}\Delta\mathbf{x}] \quad (3.13)$$

Note that the minimization of J_p in Eq. (3.13) is equivalent to the minimization of J in Eq. (3.12). For the minimization of J_p , the following conditions should be satisfied

Necessary Condition

$$\nabla_{\Delta\mathbf{x}} J_p = \frac{\partial J_p}{\partial \Delta\mathbf{x}} = \mathbf{H}^T \mathbf{W} \mathbf{H} \Delta\mathbf{x} - \mathbf{H}^T \mathbf{W} \Delta\mathbf{y}_c = 0 \quad (3.14)$$

Sufficient Condition

$$\nabla_{\Delta \mathbf{x}}^2 J_p \equiv \frac{\partial^2 J_p}{\partial \Delta x_i \partial \Delta x_j} = \mathbf{H}^T \mathbf{W} \mathbf{H} > 0, \quad (\text{p. d.}) \quad (3.15)$$

From the necessary conditions, the normal equation can be obtained

$$\mathbf{H}^T \mathbf{W} \mathbf{H} \Delta \mathbf{x} = \mathbf{H}^T \mathbf{W} \Delta \mathbf{y}_c \quad (3.16)$$

Finally, the solution for solving the weighted least-squares problem applies to solving for $\Delta \mathbf{x}$ and the explicit solution is computed by

$$\Delta \mathbf{x} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \Delta \mathbf{y}_c \quad (3.17)$$

Note that because of the nonlinearities in the nonlinear function this process must be iterated until the solution converges, i.e., $\Delta \mathbf{x}$ approaches zero.

The complete nonlinear least-squares algorithm is summarized in Fig. 3.1. An initial guess \mathbf{x}_c of the current estimates is required to begin the algorithm. A stopping condition with an accuracy dependent tolerance for the minimization of J is given by

$$\delta J \equiv \frac{|J_i - J_{i-1}|}{J_i} < \frac{\varepsilon}{\|\mathbf{W}\|} \quad (3.18)$$

where i is the iteration number and ε is a prescribed small value. If the judgment criterion in Eq. (3.18) is not satisfied, the update procedure is iterated with the new estimate as the current estimate until the process converges.

3.2 Extended Kalman Filter

The extended Kalman filter provides the minimum variance estimate of the state based on statistical information about the dynamic model and observations. In this section the EKF is reviewed from the concept of the Bayesian approach, and derived

using an *innovations concept*⁴⁹ for the discrete-time nonlinear equations given by

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{w}_k, k) \quad (3.19)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k \quad (3.20)$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the $n \times 1$ state vector, $\mathbf{y}_k \in \mathbb{R}^m$ is the $m \times 1$ observation vector. $\mathbf{w}_k \in \mathbb{R}^q$ is the $q \times 1$ state noise process vector and $\mathbf{v}_k \in \mathbb{R}^s$ is the $s \times 1$ additive measurement noise vector. It is assumed that the noise vectors are zero-mean Gaussian processes and

$$E[\mathbf{w}_k \mathbf{w}_j^T] = \delta_{kj} \mathbf{Q}_k, \quad E[\mathbf{v}_k \mathbf{v}_j^T] = \delta_{kj} \mathbf{R}_k, \quad E[\mathbf{v}_k \mathbf{w}_j^T] = 0, \quad \forall k, j \quad (3.21)$$

Given a system model and initial state and covariance values, the extended Kalman filter propagates the first two moments of the distribution of \mathbf{x}_k recursively. Then, along with imperfect measurements, the EKF updates the estimates of the state vector and the covariance. The update is accomplished through the Kalman gain matrix, \mathcal{K} , which comes from minimizing the weighted scalar sum of the diagonal elements of the covariance matrix. Thus, the EKF algorithm has a distinctive *predictor-corrector* structure, which is equivalent to the recursive Bayesian relationships in Eqs. (1.6)~(1.7).

The EKF is based on the analytical Taylor series expansion of the nonlinear systems and observation equations about the current estimated value $\hat{\mathbf{x}}_k$. Thus, for nonlinear models the predicted state estimate and covariance are approximated by^{3,27}

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k, k) \quad (3.22)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_k \quad (3.23)$$

where \mathbf{F}_k is the Jacobian matrix of \mathbf{f} evaluated about $\hat{\mathbf{x}}_k$. The update equations are

written as

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathcal{K}_{k+1} \mathbf{v}_{k+1} \quad (3.24)$$

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathcal{K}_{k+1} \mathbf{P}_{k+1}^{vv} \mathcal{K}_{k+1}^T \quad (3.25)$$

where \mathbf{v}_{k+1} is the innovative vector, which is equal to the difference between the actual and the predicted observations, and is given by

$$\mathbf{v}_{k+1} = \mathbf{y} - \hat{\mathbf{y}}_{k+1}^- = \mathbf{y} - \mathbf{h}(\hat{\mathbf{x}}_{k+1}^-, k+1) \quad (3.26)$$

The covariance of the innovation vector is obtained by

$$\mathbf{P}_{k+1}^{vv} = \mathbf{P}_{k+1}^{yy} + \mathbf{R}_{k+1} \quad (3.27)$$

where \mathbf{P}_{k+1}^{yy} is the output covariance. The *Kalman gain* \mathcal{K}_{k+1} is computed by

$$\mathcal{K}_{k+1} = \mathbf{P}_{k+1}^{xy} (\mathbf{P}_{k+1}^{vv})^{-1} \quad (3.28)$$

where \mathbf{P}_{k+1}^{xy} is the predicted cross-correlation matrix between $\hat{\mathbf{x}}_{k+1}^-$ and $\hat{\mathbf{y}}_{k+1}^-$.

The probability densities in the Bayesian recursion are related to the optimal terms in the EKF algorithms by²¹

$$p(\mathbf{x}_k | \mathbf{Y}_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k, \mathbf{P}_k) \quad (3.29)$$

$$p(\mathbf{x}_{k+1} | \mathbf{Y}_k) = \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1}^-, \mathbf{P}_k^-) \quad (3.30)$$

$$\approx \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{f}(\hat{\mathbf{x}}_k), \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_k) \quad (3.31)$$

$$p(\mathbf{x}_{k+1} | \mathbf{Y}_{k+1}) = \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1}, \mathbf{P}_{k+1}) \quad (3.32)$$

$$\approx \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1}^- + \mathcal{K}_{k+1} \mathbf{v}_{k+1}, \mathbf{P}_{k+1}^- - \mathcal{K}_{k+1} \mathbf{P}_{k+1}^{vv} \mathcal{K}_{k+1}^T) \quad (3.33)$$

where $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{P})$ denotes a Gaussian density with argument \mathbf{x} , mean \mathbf{m} , and covariance \mathbf{P} . The recursive Bayesian relations of the predictor-corrector structure for the

extended Kalman filter can be represented by the block diagram as shown in Fig. 3.2, and the specific algorithm of the EKF is formulated in terms of the innovation vector and covariance terms and summarized in Table 3.1.

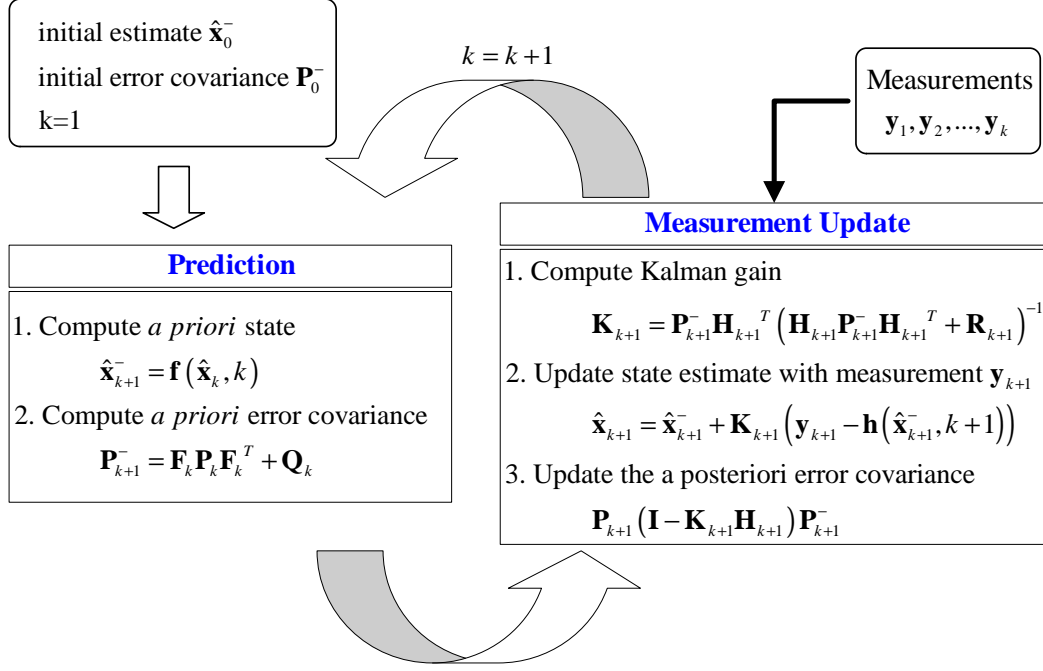


Fig. 3.2 Diagram of Predictor-Corrector Form of the Extended Kalman Filter

Note that in the EKF algorithm the state distribution is approximated by a Gaussian random variable, which is propagated through the first-order linearization of the nonlinear functions. These approximations, however, can introduce large errors in the true mean and covariance. In the next section, a new nonlinear transformation for the mean and covariance will be introduced to handle the linearization issue.

3.3 Unscented Kalman Filter

The basic difference between the EKF and unscented Kalman filter results from the manner in which the state distribution of the nonlinear model is approximated. The

Table 3.1 Extended Kalman Filter (EKF) Algorithm

Initialization:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$$

$$\mathbf{P}_0 = E \left[(\mathbf{x}_0 - \hat{\mathbf{x}}_0) (\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T \right]$$

State Propagation:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k, k)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_k$$

Observation Propagation:

$$\hat{\mathbf{y}}_{k+1}^- = \mathbf{h}(\hat{\mathbf{x}}_{k+1}^-, k+1)$$

$$\mathbf{P}_{k+1}^{vv} = \mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1}$$

$$\mathbf{P}_{k+1}^{xy} = \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T$$

Update:

$$\mathcal{K}_{k+1} = \mathbf{P}_{k+1}^{xy} (\mathbf{P}_{k+1}^{vv})^{-1}$$

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathcal{K}_{k+1} \mathbf{P}_{k+1}^{vv} \mathcal{K}_{k+1}^T$$

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathcal{K}_{k+1} \mathbf{v}_{k+1}$$

where $(-)$ denotes a “propagated” value, $(+)$ denotes a “updated” value

\mathbf{Q}_k = System Model Error Matrix, \mathbf{R}_k = Measurement Error Matrix

unscented Kalman filter introduced by Julier⁸ utilizes a nonlinear transformation, called the unscented transformation, in which the state probability distribution is represented by a minimal set of sampled sigma points, which are used to parameterize the true mean and covariance of the state distribution. Thus, the unscented Kalman filter belongs to the simulation-based local nonlinear filters and can be described as a sigma point filter (SPF) in a unified way. When they are propagated through the true nonlinear system, the posterior mean and covariance is obtained up to the 2nd order for any nonlinearity. First, the unscented transformation is described.

3.3.1 Unscented Transformation

The unscented transformation (UT) utilizes a deterministic sampling approach to calculate the mean and covariance, and works on the principle that it is easier to approximate a probability distribution than an arbitrary nonlinear function.⁹ Consider the propagation of a random variable $\mathbf{x} \in \Re^n$ with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{xx}}$ through a nonlinear function $\mathbf{y} = \mathbf{h}(\mathbf{x})$. To calculate the statistics of $\mathbf{y} \in \Re^m$, a new matrix \mathcal{X} that consists of $(2n + 1)$ weighted sigma vectors \mathcal{X}_i is formed according to the following

$$\begin{aligned} \mathcal{X}_0 &= \bar{\mathbf{x}} & W_0 &= \kappa/(n + \kappa) \quad i = 0 \\ \mathcal{X}_i &= \bar{\mathbf{x}} + \left(\sqrt{(n + \kappa)\mathbf{P}_{\mathbf{xx}}} \right)_i & W_i &= 1/2(n + \kappa) \quad i = 1, \dots, n \\ \mathcal{X}_i &= \bar{\mathbf{x}} - \left(\sqrt{(n + \kappa)\mathbf{P}_{\mathbf{xx}}} \right)_{i-n} & W_i &= 1/2(n + \kappa) \quad i = n + 1, \dots, 2n \end{aligned} \quad (3.34)$$

where $\kappa \in \Re$ is a scaling parameter that designates the scaling direction of sigma points and $\left(\sqrt{(n + \kappa)\mathbf{P}_{\mathbf{xx}}} \right)_i$ is the i th row or column of the matrix square root. W_i is the weight which is associated with the i th point. These sigma point vectors are

propagated through the true nonlinear transformation without linearization by

$$\mathcal{Y}_i = \mathbf{h}(\mathcal{X}_i) \quad i = 0, \dots, 2n \quad (3.35)$$

Then, the estimated mean and covariance of \mathbf{y} are approximated using a weighted sample mean and covariance of the posterior sigma points respectively

$$\bar{\mathbf{y}} = \sum_{i=0}^{2n} W_i \mathcal{Y}_i \quad (3.36)$$

$$\mathbf{P}_{yy} = \sum_{i=0}^{2n} W_i \{\mathcal{Y}_i - \bar{\mathbf{y}}\} \{\mathcal{Y}_i - \bar{\mathbf{y}}\}^T \quad (3.37)$$

These estimates of the mean and covariance are captured accurately up to the second order (third order for Gaussian) of the Taylor series expansion for an arbitrary nonlinear function.⁵⁰ A simple illustration of the approach is summarized in Fig. 3.3 for a 2-dimensional unscented transformation. The lower plot shows the mean and covariance obtained from the sampled sigma-point approach through the unscented transformation, whereas the upper plot depicts the results from a linearization approach about a single point, the first moment of a Gaussian prior. The UT approach results in third order accuracy for Gaussian inputs, which leads to estimates of the mean and covariance closer to the truth. In contrast, the linearized method in the EKF results in only first order accuracy and large errors. In the transformation κ provides an extra degree of freedom to fine tune the higher order moments of the approximation. If \mathbf{x} is a Gaussian distribution, then $\kappa = 3 - n$ is used for multi-dimensional systems.⁵⁰ The distance of the sigma point from $\bar{\mathbf{x}}$, $|\mathcal{X}_i - \bar{\mathbf{x}}|$ is proportional to $\sqrt{n + \kappa}$. When $\kappa = 0$ the distance is proportional to \sqrt{n} , when $\kappa > 0$ the points are scaled further from $\bar{\mathbf{x}}$, and when $\kappa < 0$ the points are scaled toward $\bar{\mathbf{x}}$. Although κ can be positive or negative, a negative choice can result in the calculated covariance being negative semi-definite. A scaled unscented transformation (SUT) was developed to handle this

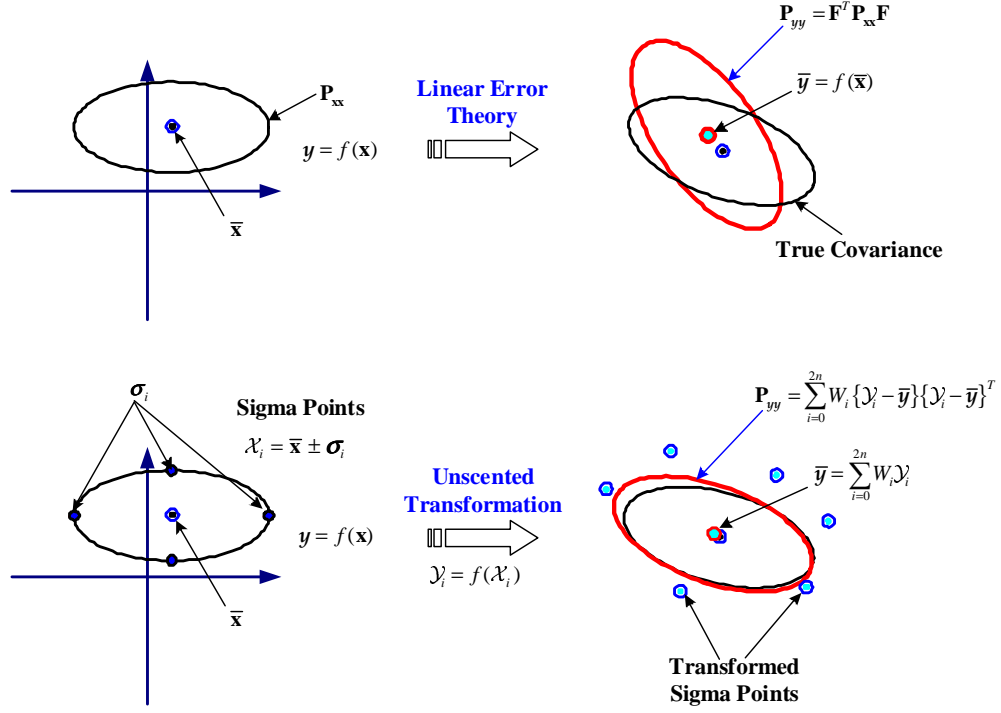


Fig. 3.3 Schematic Diagram for Mean and Covariance Propagation

issue.⁵¹

3.3.2 Scaled Unscented Transformation

This section introduces a general framework for scaling sigma points to deal with the non-positive covariance matrix.⁵¹ The scaled unscented transformation (SUT) replaces the original set of sigma points with a transformed set that could obey the conditions

$$\mathcal{X}'_i = \mathcal{X}_i + \alpha(\mathcal{X}_i - \mathcal{X}_0) \quad (3.38)$$

where α is the positive parameter that scales the spread of the sigma point and can be set sufficiently small such that the higher order terms do not affect the filter. A

sigma point set $\mathcal{S} = \{W, \mathcal{X}\}$ that is obtained using equation (3.34) is transformed into the scaled set $\mathcal{S}' = \{W', \mathcal{X}'\}$ given by

$$W'_i = \begin{cases} W_0/\alpha^2 + (1 - 1/\alpha^2) & i = 0 \\ W_i/\alpha^2 & i \neq 0 \end{cases} \quad (3.39)$$

Then, the statistics of the estimated mean and covariance of \mathbf{y} are formulated by

$$\mathcal{Y}'_i = \mathbf{h}(\mathcal{X}'_i) \quad (3.40)$$

$$\bar{\mathbf{y}}' = \sum_{i=0}^{2n} W'_i \mathcal{Y}'_i \quad (3.41)$$

$$\mathbf{P}'_{yy} = \sum_{i=0}^{2n} W'_i \{\mathcal{Y}'_i - \bar{\mathbf{y}}'\} \{\mathcal{Y}'_i - \bar{\mathbf{y}}'\}^T + (1 + \beta - \alpha^2) \{\mathcal{Y}'_i - \bar{\mathbf{y}}'\} \{\mathcal{Y}'_i - \bar{\mathbf{y}}'\}^T \quad (3.42)$$

where β is the third parameter that incorporates further higher order effects by adding the weighting of the zeroth sigma point of the calculation of the covariance, and $\beta = 2$ is optimal for Gaussian distributions. α controls the size of the sigma point distribution and should be a small number ($0 \leq \alpha \leq 1$).^{51,52} The sigma point selection formula and scaling effect parameter α can be combined into a new single parameter by setting

$$\lambda = \alpha^2(n + \kappa) - n \quad (3.43)$$

Then, the selection of the scaled sigma points is reformulated in a simple form as follows

$$\begin{aligned} \mathcal{X}_0 &= \bar{\mathbf{x}} & i &= 0 \\ \mathcal{X}_i &= \bar{\mathbf{x}} + \left(\sqrt{(n + \lambda) \mathbf{P}_{\mathbf{xx}}} \right)_i & i &= 1, \dots, n \\ \mathcal{X}_i &= \bar{\mathbf{x}} - \left(\sqrt{(n + \lambda) \mathbf{P}_{\mathbf{xx}}} \right)_{i-n} & i &= n + 1, \dots, 2n \end{aligned} \quad (3.44)$$

$$W_i^{(m)} = \begin{cases} \lambda/n + \lambda & i = 0 \\ 1/\{2(n + \lambda)\} & i = 1, \dots, 2n \end{cases} \quad (3.45)$$

$$W_i^{(c)} = \begin{cases} \lambda/(n + \lambda) + (1 - \alpha^2 + \beta) & i = 0 \\ 1/\{2(n + \lambda)\} & i = 1, \dots, 2n \end{cases} \quad (3.46)$$

where $W_i^{(m)}$ is the weight for the mean and $W_i^{(c)}$ is the weight for the covariance. Even though the scaled unscented transformation provides the same performance as the truncated second order filter with the same order of calculations as an extended Kalman filter, in real-time applications, however, it is critical that both the computational costs and storage requirements should be minimized. Julier⁵³ introduced reduced sigma points, that can minimize the number of regression points from $2n + 1$ to $n + 1$ for an n dimensional state space. The reduced sigma points are called *simplex sigma points*, and result in a computationally more efficient solution. See Ref. 53 for details.

3.3.3 Unscented Kalman Filter

In the previous section a series of transformation methods for parameterizing the mean and covariance were introduced. In this section the scaled unscented transformation is applied to the nonlinear filtering problem. The UKF is a straightforward extension of the UT to the recursive estimation for both dynamic and measurement models. The UKF is derived for discrete-time nonlinear equations presented in Eqs. (3.19) and (3.20). Note that the process noise is not simply additive but the measurement noise is assumed to be additive. Assume that $\mathbf{x}_k \in \mathbb{R}^n$ is the $n \times 1$ state vector and $\mathbf{y}_k \in \mathbb{R}^m$ is the $m \times 1$ measurement vector at time k . $\mathbf{w}_k \in \mathbb{R}^q$ is the $q \times 1$ process noise vector and $\mathbf{v}_k \in \mathbb{R}^r$ is the $r \times 1$ additive measurement noise vector, and they are assumed to be zero-mean Gaussian noise processes with covariances given by \mathbf{Q}_k

and \mathbf{R}_k , respectively. The original state vector is redefined as an augmented state vector along with process noise variables and an augmented covariance matrix on the diagonal is reconstructed as

$$\mathbf{x}_k^a = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{w}_k \end{bmatrix}, \quad \mathbf{P}_k^a = \begin{bmatrix} \mathbf{P}_k & \mathbf{P}_k^{xw} \\ \mathbf{P}_k^{xw} & \mathbf{Q}_k \end{bmatrix} \quad (3.47)$$

where \mathbf{P}_k^{xw} is the correlation between the process noise and the error in the state estimate. The set of $\{2(n+q)+1\}$ sigma points of the augmented state is constructed by implementing the nonlinear transformation in Eq. (3.44) to the augmented system given by Eq. (3.47). The unscented Kalman filter can be formulated by taking the *predictor-corrector* step made in the EKF algorithm in Table 3.1. As for the state propagation, the predicted state vector $\hat{\mathbf{x}}_{k+1}^-$ and its predicted covariance \mathbf{P}_{k+1}^- are computed by applying one of the proposed unscented transformations

$$\mathcal{X}_{i,k+1} = \mathbf{f}(\mathcal{X}_{i,k}^a, k) \quad (3.48)$$

$$\hat{\mathbf{x}}_{k+1}^- = \sum_{i=0}^{2(n+q)} W_i^{(m)} \mathcal{X}_{i,k+1} \quad (3.49)$$

$$\mathbf{P}_{k+1}^- = \sum_{i=0}^{2(n+q)} W_i^{(c)} \{\mathcal{X}_{i,k+1} - \hat{\mathbf{x}}_{k+1}^-\} \{\mathcal{X}_{i,k+1} - \hat{\mathbf{x}}_{k+1}^-\}^T \quad (3.50)$$

Similarly, the predicted observation vector $\hat{\mathbf{y}}_{k+1}^-$ and its predicted covariance \mathbf{P}_{k+1}^{yy} are also calculated as

$$\mathcal{Y}_{i,k+1} = \mathbf{h}(\mathcal{X}_{i,k+1}, k+1) \quad (3.51)$$

$$\hat{\mathbf{y}}_{k+1}^- = \sum_{i=0}^{2(n+q)} W_i^{(m)} \mathcal{Y}_{i,k+1} \quad (3.52)$$

$$\mathbf{P}_{k+1}^{yy} = \sum_{i=0}^{2(n+q)} W_i^{(c)} \{\mathcal{Y}_{i,k+1} - \hat{\mathbf{y}}_{k+1}^-\} \{\mathcal{Y}_{i,k+1} - \hat{\mathbf{y}}_{k+1}^-\}^T \quad (3.53)$$

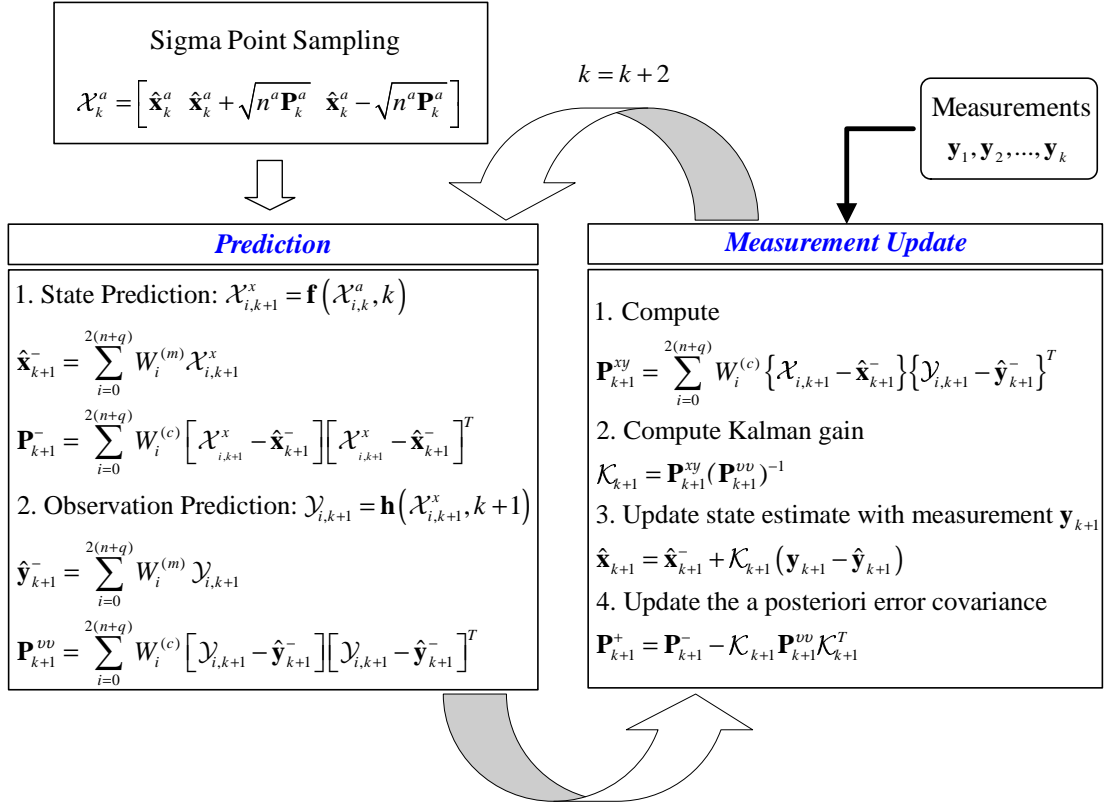


Fig. 3.4 Diagram of Predictor-Corrector Form of the Unscented Kalman Filter

Since the observation noise is additive and independent, the innovation covariance

\mathbf{P}_{k+1}^{vv} is computed by

$$\mathbf{P}_{k+1}^{vv} = \mathbf{P}_{k+1}^{yy} + \mathbf{R}_{k+1} \quad (3.54)$$

and the cross correlation matrix is determined by using

$$\mathbf{P}_{k+1}^{xy} = \sum_{i=0}^{2(n+q)} W_i^{(c)} \{\mathcal{X}_{i,k+1}^x - \hat{\mathbf{x}}_{k+1}^-\} \{\mathcal{Y}_{i,k+1} - \hat{\mathbf{y}}_{k+1}^-\}^T \quad (3.55)$$

The filter gain \mathcal{K}_{k+1} is computed by using Eq. (3.28) and the updated state estimate $\hat{\mathbf{x}}_{k+1}^+$ and covariance \mathbf{P}_{k+1}^+ can be obtained using Eqs. (3.24) and (3.25), respectively. The recursive Bayesian relations of the predictor-corrector structure for the unscented Kalman filter can be represented by the block diagram as shown in

Table 3.2 Unscented Kalman Filter (UKF) Algorithm

Initialization:

$$\hat{\mathbf{x}}_k^a = E[\mathbf{x}_k^a] = E[\mathbf{x}_k^T \mathbf{w}_k^T]^T \in \Re^{n+q}$$

$$\mathbf{P}_k^a = E[(\mathbf{x}_k^a - \hat{\mathbf{x}}_k^a)(\mathbf{x}_k^a - \hat{\mathbf{x}}_k^a)^T] = \begin{bmatrix} \mathbf{P}_k & \mathbf{P}_k^{xw} \\ \mathbf{P}_k^{xw} & \mathbf{Q}_k \end{bmatrix}$$

Sigma Points:

$$\boldsymbol{\sigma}_k^a = \sqrt{(n+q+\lambda)\mathbf{P}_k^a}$$

$$\mathcal{X}_{i,k}^a = ([\hat{\mathbf{x}}_k^a \quad \hat{\mathbf{x}}_k^a + \boldsymbol{\sigma}_k^a \quad \hat{\mathbf{x}}_k^a - \boldsymbol{\sigma}_k^a])_i$$

Propagation:

$$\mathcal{X}_{i,k+1} = \mathbf{f}(\mathcal{X}_{i,k}^a, k)$$

$$\hat{\mathbf{x}}_{k+1}^- = \sum_{i=0}^{2(n+q)} W_i^{(m)} \mathcal{X}_{i,k+1}$$

$$\mathbf{P}_{k+1}^- = \sum_{i=0}^{2(n+q)} W_i^{(c)} \{\mathcal{X}_{i,k+1} - \hat{\mathbf{x}}_{k+1}^-\} \{\mathcal{X}_{i,k+1} - \hat{\mathbf{x}}_{k+1}^-\}^T$$

$$\mathcal{Y}_{i,k+1} = \mathbf{h}(\mathcal{X}_{i,k+1}, k+1)$$

$$\hat{\mathbf{y}}_{k+1}^- = \sum_{i=0}^{2(n+q)} W_i^{(m)} \mathcal{Y}_{i,k+1}$$

$$\mathbf{P}_{k+1}^{vv} = \sum_{i=0}^{2(n+q)} W_i^{(c)} \{\mathcal{Y}_{i,k+1} - \hat{\mathbf{y}}_{k+1}^-\} \{\mathcal{Y}_{i,k+1} - \hat{\mathbf{y}}_{k+1}^-\}^T + \mathbf{R}_{k+1}$$

$$\mathbf{P}_{k+1}^{xy} = \sum_{i=0}^{2(n+q)} W_i^{(c)} \{\mathcal{X}_{i,k+1} - \hat{\mathbf{x}}_{k+1}^-\} \{\mathcal{Y}_{i,k+1} - \hat{\mathbf{y}}_{k+1}^-\}^T$$

Update:

$$\mathcal{K}_{k+1} = \mathbf{P}_{k+1}^{xy} (\mathbf{P}_{k+1}^{vv})^{-1}$$

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathcal{K}_{k+1} \mathbf{P}_{k+1}^{vv} \mathcal{K}_{k+1}^T$$

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathcal{K}_{k+1} \mathbf{v}_{k+1}$$

Fig. 3.4.

The UKF algorithm is summarized in Table 3.2. The formulation indicates that correlated noise sources can be implemented efficiently without any modification of the filter algorithms. For the special case in which both the process and measurement noises are purely additive, the computational complexity of the UKF can be reduced by adjusting the augmented state vector.⁵² In other words, the system state need not be augmented with the random noise variables. If the observation noise is introduced in a nonlinear fashion and is corrected with measurement noise, then the augmented state vector is expanded to include the observational terms too. This approach may more fully utilize the capacity of the unscented transformation, but it will be more computationally expensive due to the extra calculations. Note that no explicit calculation of the Jacobian and/or Hessian matrix is necessary to implement this algorithm, and the formulation is ideally suited for parallel computation since the propagations can be performed in parallel.

Several approaches have addressed the modifications of the EKF to enhance the computational stability. The matrix square root can be implemented by using a Cholesky factorization method and Joseph's algorithm⁷ that would guarantee non-negative covariance matrices. The square-root factorization method could be applied to the unscented Kalman filter to increase the numerical stability as well, and the algorithm is called the square-root UKF.⁵²

3.4 Divided Difference Filter

In this section, the proposed algorithm, referred to as the divided difference filter (DDF) proposed by Nørgaard,¹² is an efficient extension of the Kalman Filter for nonlinear systems. The DDF is described as a sigma point filter (SPF) in a unified

way where the filter linearizes the nonlinear dynamic and measurement functions by using an interpolation formula through systematically chosen sigma points. The linearization is based on polynomial approximations of the nonlinear transformations that are obtained by Stirling's interpolation formula, rather than the derivative-based Taylor series approximation. Conceptually, the implementation principle resembles that of the EKF, the implementation, however, is significantly simpler because it is not necessary to formulate the Jacobian and/or Hessian matrices of partial derivatives of the nonlinear dynamic and measurement equations. Thus, the new nonlinear state filter, divided difference filter (DDF), can also replace the extended Kalman filter (EKF) and its higher-order estimators in practical real-time applications that require accurate estimation, but less computational cost.

Nørgaard developed the divided difference filters that works on general discrete-time nonlinear models in which the noise sources are not assumed to be additive. DDF1 is a divided difference filter derived by using the first-order approximation, and DDF2 is a second-order divided difference filter. In this paper, however, we further formulated the divided difference filters in terms of the innovation vector approach and the additive process and measurement noise sources.

3.4.1 Linearization

First, an alternative linearization method called Sterling's interpolation is presented for the mean and covariance calculations. Consider a nonlinear function, $\mathbf{y} = \mathbf{h}(\mathbf{x}) \in \Re^m$ with a random variable $\mathbf{x} \in \Re^n$ with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{xx}}$. If the function \mathbf{h} is analytic, then the multi-dimensional Taylor series expansion about the mean $\bar{\mathbf{x}}$ is described (notation by Julier⁵⁰).

$$\mathbf{y} \simeq \mathbf{h}(\bar{\mathbf{x}} + \Delta\mathbf{x}) = \mathbf{h}(\bar{\mathbf{x}}) + \mathcal{D}_{\Delta\mathbf{x}}\mathbf{h} + \frac{1}{2!}\mathcal{D}_{\Delta\mathbf{x}}^2\mathbf{h} + \frac{1}{3!}\mathcal{D}_{\Delta\mathbf{x}}^3\mathbf{h} + \dots \quad (3.56)$$

where the operator $\mathcal{D}_{\Delta \mathbf{x}}^i \mathbf{h}$ evaluates the total derivative of $\mathbf{h}(\mathbf{x})$ and is described as

$$\mathcal{D}_{\Delta \mathbf{x}}^i \mathbf{h} = \left(\Delta x_1 \frac{\partial}{\partial x_1} + \Delta x_2 \frac{\partial}{\partial x_2} + \cdots \Delta x_n \frac{\partial}{\partial x_n} \right)^i \mathbf{h}(\mathbf{x}) \Big|_{\mathbf{x}=\bar{\mathbf{x}}} \quad (3.57)$$

The first and second order operators can be written as

$$\begin{aligned} \mathcal{D}_{\Delta \mathbf{x}} \mathbf{h} &= \left(\sum_{p=1}^n \Delta x_p \frac{\partial}{\partial x_p} \right) \mathbf{h}(\mathbf{x}) \Big|_{\mathbf{x}=\bar{\mathbf{x}}} \\ \mathcal{D}_{\Delta \mathbf{x}}^2 \mathbf{h} &= \left(\sum_{p=1}^n \sum_{q=1}^n \Delta x_p \Delta x_q \frac{\partial^2}{\partial x_p \partial x_q} \right) \mathbf{h}(\mathbf{x}) \Big|_{\mathbf{x}=\bar{\mathbf{x}}} \end{aligned} \quad (3.58)$$

The second order divided difference approximation of the function is formulated by using the vector form of Stirling's interpolation formula, which is similar to the extension of the Taylor series approximation¹²

$$\mathbf{y} \simeq \mathbf{h}(\mathbf{x}) + \tilde{\mathcal{D}}_{\Delta \mathbf{x}} \mathbf{h} + \frac{1}{2!} \tilde{\mathcal{D}}_{\Delta \mathbf{x}}^2 \mathbf{h} \quad (3.59)$$

The divided difference operators $\tilde{\mathcal{D}}_{\Delta \mathbf{x}}$, $\tilde{\mathcal{D}}_{\Delta \mathbf{x}}^2$ are given by

$$\begin{aligned} \tilde{\mathcal{D}}_{\Delta \mathbf{x}} \mathbf{h} &= \frac{1}{h} \left(\sum_{p=1}^n \Delta x_p \mu_p \delta_p \right) \mathbf{h}(\bar{\mathbf{x}}) \\ \tilde{\mathcal{D}}_{\Delta \mathbf{x}}^2 \mathbf{h} &= \frac{1}{h^2} \left(\sum_{p=1}^n \Delta x_p^2 \delta_p^2 + \sum_{p=1}^n \sum_{q=1, p \neq q}^n \Delta x_p \Delta x_q (\mu_p \delta_p)(\mu_q \delta_q) \right) \mathbf{h}(\bar{\mathbf{x}}) \end{aligned} \quad (3.60)$$

where h is an interval of length, $h = \sqrt{3}$ is usually set for a Gaussian distribution, and δ_p and μ_p denote the partial difference operator and the partial average operator respectively

$$\begin{aligned} \delta_p \mathbf{h}(\bar{\mathbf{x}}) &= \mathbf{h} \left(\bar{\mathbf{x}} + \frac{h}{2} \mathbf{e}_p \right) - \mathbf{h} \left(\bar{\mathbf{x}} - \frac{h}{2} \mathbf{e}_p \right) \\ \mu_p \mathbf{h}(\bar{\mathbf{x}}) &= \frac{1}{2} \left\{ \mathbf{h} \left(\bar{\mathbf{x}} + \frac{h}{2} \mathbf{e}_p \right) + \mathbf{h} \left(\bar{\mathbf{x}} - \frac{h}{2} \mathbf{e}_p \right) \right\} \end{aligned} \quad (3.61)$$

and \mathbf{e} is the p th unit vector along the coordinate axis in the space spanned by \mathbf{x} .

Eq. (3.59) is just a multi-dimensional interpolation formula. The following linear transformation of \mathbf{x} is introduced to illustrate how others can be derived

$$\mathbf{z} = \mathbf{S}_{\mathbf{x}}^{-1}\mathbf{x} \quad (3.62)$$

where $\mathbf{S}_{\mathbf{x}}$ is the Cholesky factor of the covariance matrix $\mathbf{P}_{\mathbf{xx}}$. A new function $\tilde{\mathbf{h}}$ is defined by

$$\tilde{\mathbf{h}}(\mathbf{z}) \equiv \mathbf{h}(\mathbf{S}_{\mathbf{x}}\mathbf{z}) = \mathbf{h}(\mathbf{x}) \quad (3.63)$$

The Taylor series approximation of $\tilde{\mathbf{h}}$ is identical to that of \mathbf{h} , while the interpolation formula does not yield the same results for $\tilde{\mathbf{h}}$ and \mathbf{h} due to the following

$$2\mu_p\delta_p\tilde{\mathbf{h}}(\bar{\mathbf{z}}) = \tilde{\mathbf{h}}(\bar{\mathbf{z}} + h\mathbf{e}_p) - \tilde{\mathbf{h}}(\bar{\mathbf{z}} - h\mathbf{e}_p) = \mathbf{h}(\bar{\mathbf{x}} + \mathbf{s}_p) - \mathbf{h}(\bar{\mathbf{x}} - h\mathbf{s}_p) \quad (3.64)$$

where \mathbf{s}_p denotes the p th column of $\mathbf{S}_{\mathbf{x}}$. Thus, $\tilde{\mathcal{D}}_{\Delta\mathbf{x}}\tilde{\mathbf{h}}$ and $\tilde{\mathcal{D}}_{\Delta\mathbf{x}}^2\tilde{\mathbf{h}}$ will be different from $\tilde{\mathcal{D}}_{\Delta\mathbf{z}}\tilde{\mathbf{h}}$ and $\tilde{\mathcal{D}}_{\Delta\mathbf{z}}^2\tilde{\mathbf{h}}$. In the following section the calculations of the mean and covariance, similar to the approach taken for the extended Kalman filter, are described by applying the interpolation formula with the function $\tilde{\mathbf{h}}(\mathbf{z})$.

3.4.2 First-Order Approximation

The first-order approximation of the nonlinear system is obtained by using the divided difference operator

$$\begin{aligned} \mathbf{y} &\simeq \tilde{\mathbf{h}}(\bar{\mathbf{z}}) + \tilde{\mathcal{D}}_{\Delta\mathbf{z}}\tilde{\mathbf{h}} \\ &= \tilde{\mathbf{h}}(\bar{\mathbf{z}}) + \frac{1}{h} \left(\sum_{p=1}^n \Delta z_p \mu_p \delta_p \right) \tilde{\mathbf{h}}(\bar{\mathbf{z}}) \end{aligned} \quad (3.65)$$

It is assumed that $\Delta\mathbf{z}$ is Gaussian with zero mean and the elements are uncorrelated. This assumption leads to an expression for the expectation of $\tilde{\mathbf{h}}$ given by the estimate

$$\bar{\mathbf{y}} = \mathbf{E} \left\{ \tilde{\mathbf{h}}(\bar{\mathbf{z}}) + \tilde{\mathcal{D}}_{\Delta\mathbf{z}}\tilde{\mathbf{h}} \right\} = \tilde{\mathbf{h}}(\bar{\mathbf{z}}) = \mathbf{h}(\bar{\mathbf{x}}) \quad (3.66)$$

The covariance estimate is defined by

$$\mathbf{P}_{yy} = \mathbf{E} \left\{ (\mathbf{y} - \bar{\mathbf{y}})(\mathbf{y} - \bar{\mathbf{y}})^T \right\} \quad (3.67)$$

and the covariance is written in terms of the divided difference operators as

$$\begin{aligned} \mathbf{P}_{yy} &= \mathbf{E} \left\{ \left(\tilde{\mathcal{D}}_{\Delta \mathbf{z}} \tilde{\mathbf{h}} \right) \left(\tilde{\mathcal{D}}_{\Delta \mathbf{z}} \tilde{\mathbf{h}} \right)^T \right\} \\ &= \frac{1}{4h^2} \sum_{p=1}^n \left[\tilde{\mathbf{h}}(\bar{\mathbf{z}} + h\mathbf{e}_p) - \tilde{\mathbf{h}}(\bar{\mathbf{z}} - h\mathbf{e}_p) \right] \left[\tilde{\mathbf{h}}(\bar{\mathbf{z}} + h\mathbf{e}_p) - \tilde{\mathbf{h}}(\bar{\mathbf{z}} - h\mathbf{e}_p) \right]^T \end{aligned} \quad (3.68)$$

It is assumed that the element of $\Delta \mathbf{z}$ is independent with symmetric distribution and the odd order moments are neglected. Applying Eq. (3.64) to Eq. (3.68) leads to the covariance estimate¹²

$$\mathbf{P}_{yy} = \frac{1}{4h^2} \sum_{p=1}^n [\mathbf{h}(\bar{\mathbf{x}} + h\mathbf{s}_{\mathbf{x},p}) - \mathbf{h}(\bar{\mathbf{x}} - h\mathbf{s}_{\mathbf{x},p})] [\mathbf{h}(\bar{\mathbf{x}} + h\mathbf{s}_{\mathbf{x},p}) - \mathbf{h}(\bar{\mathbf{x}} - h\mathbf{s}_{\mathbf{x},p})]^T \quad (3.69)$$

where $\mathbf{s}_{\mathbf{x},p}$ is the p th column of the square Cholesky factor of the covariance matrix $\mathbf{S}_{\mathbf{x}}$, and note that $\tilde{\mathbf{h}}(\bar{\mathbf{z}} \pm h\mathbf{e}_p) = \mathbf{h}(\bar{\mathbf{x}} \pm h\mathbf{s}_{\mathbf{x},p})$. Finally, the cross-covariance estimate, \mathbf{P}_{xy} can be derived in a similar way

$$\begin{aligned} \mathbf{P}_{xy} &= \mathbf{E} \left\{ (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{y} - \bar{\mathbf{y}})^T \right\} \\ &= \frac{1}{2h} \sum_{p=1}^n \mathbf{s}_{x,p} \left[\tilde{\mathbf{h}}(\bar{\mathbf{z}} + h\mathbf{e}_p) - \tilde{\mathbf{h}}(\bar{\mathbf{z}} - h\mathbf{e}_p) \right]^T \end{aligned} \quad (3.70)$$

which also can be written by

$$\mathbf{P}_{xy} = \frac{1}{2h} \sum_{p=1}^n \mathbf{s}_{x,p} [\mathbf{h}(\bar{\mathbf{x}} + h\mathbf{s}_{\mathbf{x},p}) - \mathbf{h}(\bar{\mathbf{x}} - h\mathbf{s}_{\mathbf{x},p})]^T \quad (3.71)$$

3.4.3 Second-Order Approximation

In a similar way the derivations of the mean and covariance can be obtained by applying the second-order polynomial approximation of the nonlinear $\tilde{\mathbf{h}}$ with the

interpolation formula¹²

$$\mathbf{y} \simeq \tilde{\mathbf{h}}(\bar{\mathbf{z}}) + \tilde{\mathcal{D}}_{\Delta\mathbf{z}}\tilde{\mathbf{h}} + \frac{1}{2!}\tilde{\mathcal{D}}_{\Delta\mathbf{z}}^2\tilde{\mathbf{h}} \quad (3.72)$$

and taking the expectation operator $\mathbf{E}\{\cdot\}$ to Eq. (3.71) provides the following

$$\bar{\mathbf{y}} = \frac{h^2 - n}{h^2}\tilde{\mathbf{h}}(\bar{\mathbf{z}}) + \frac{1}{2h^2}\sum_{p=1}^n \left[\tilde{\mathbf{h}}(\bar{\mathbf{z}} + h\mathbf{e}_p) + \tilde{\mathbf{h}}(\bar{\mathbf{z}} - h\mathbf{e}_p) \right] \quad (3.73)$$

and can be calculated as

$$\bar{\mathbf{y}} = \frac{h^2 - n}{h^2}\mathbf{h}(\bar{\mathbf{x}}) + \frac{1}{2h^2}\sum_{p=1}^n [\mathbf{h}(\bar{\mathbf{x}} + h\mathbf{s}_{\mathbf{x},p}) + \mathbf{h}(\bar{\mathbf{x}} - h\mathbf{s}_{\mathbf{x},p})] \quad (3.74)$$

The covariance estimate is defined in Eq. (3.67), and after some manipulations the derivation of the covariance estimate is obtained (see Ref. 12 for details)

$$\mathbf{P}_{yy} = \mathbf{E} \left\{ \left[\tilde{\mathcal{D}}_{\Delta\mathbf{z}}\tilde{\mathbf{h}} + \frac{1}{2}\tilde{\mathcal{D}}_{\Delta\mathbf{z}}^2\tilde{\mathbf{h}} \right] \left[\tilde{\mathcal{D}}_{\Delta\mathbf{z}}\tilde{\mathbf{h}} + \frac{1}{2}\tilde{\mathcal{D}}_{\Delta\mathbf{z}}^2\tilde{\mathbf{h}} \right]^T \right\} \quad (3.75)$$

which leads to the following

$$\begin{aligned} \mathbf{P}_{yy} &= \frac{1}{4h^2} \sum_{p=1}^n [\mathbf{h}(\bar{\mathbf{x}} + h\mathbf{s}_{\mathbf{x},p}) - \mathbf{h}(\bar{\mathbf{x}} - h\mathbf{s}_{\mathbf{x},p})] [\mathbf{h}(\bar{\mathbf{x}} + h\mathbf{s}_{\mathbf{x},p}) - \mathbf{h}(\bar{\mathbf{x}} - h\mathbf{s}_{\mathbf{x},p})]^T + \\ &\quad \frac{h^2 - 1}{4h^2} \sum_{p=1}^n [\mathbf{h}(\bar{\mathbf{x}} + h\mathbf{s}_{\mathbf{x},p}) + \mathbf{h}(\bar{\mathbf{x}} - h\mathbf{s}_{\mathbf{x},p}) - 2\mathbf{h}(\bar{\mathbf{x}})] [\mathbf{h}(\bar{\mathbf{x}} + h\mathbf{s}_{\mathbf{x},p}) + \mathbf{h}(\bar{\mathbf{x}} - h\mathbf{s}_{\mathbf{x},p}) - 2\mathbf{h}(\bar{\mathbf{x}})]^T \end{aligned} \quad (3.76)$$

The cross-covariance estimate \mathbf{P}_{xy} turns out to be the same as in the first-order approximation since the higher-order moments are canceled out

$$\begin{aligned} \mathbf{P}_{xy} &= \mathbf{E} \left\{ (\mathbf{S}_{\mathbf{x}}\Delta\mathbf{z}) \left(\tilde{\mathcal{D}}_{\Delta\mathbf{z}}\tilde{\mathbf{h}} + \frac{1}{2}\tilde{\mathcal{D}}_{\Delta\mathbf{z}}^2\tilde{\mathbf{h}} \right)^T \right\} \\ &= \mathbf{E} \left\{ (\mathbf{S}_{\mathbf{x}}\Delta\mathbf{z}) \left(\tilde{\mathcal{D}}_{\Delta\mathbf{z}}\tilde{\mathbf{h}} \right)^T \right\} \\ &= \frac{1}{2h} \sum_{p=1}^n \mathbf{s}_{x,p} \left[\tilde{\mathbf{h}}(\bar{\mathbf{z}} + h\mathbf{e}_p) - \tilde{\mathbf{h}}(\bar{\mathbf{z}} - h\mathbf{e}_p) \right]^T \end{aligned} \quad (3.77)$$

which leads to the following

$$\mathbf{P}_{xy} = \frac{1}{2h} \sum_{p=1}^n \mathbf{s}_{x,p} [\mathbf{h}(\bar{\mathbf{x}} + h\mathbf{s}_{\mathbf{x},p}) - \mathbf{h}(\bar{\mathbf{x}} - h\mathbf{s}_{\mathbf{x},p})]^T \quad (3.78)$$

After implementing the formulas for the mean and covariance estimates into the EKF operation structure, the first-order and second-order filtering algorithms for the DDF1 and the DDF2 are derived respectively in terms of the innovation vector used in the EKF and the UKF.

3.4.4 First-Order Divided Difference Filter

This subsection describes the divided difference filter algorithms based on the previously derived polynomial approximations for parameterizing the mean and covariance. Nørgaard developed the divided difference filters based on general discrete-time nonlinear models in which there is no assumption of the additivity of the noise sources.¹² In this paper, however, we further derive the divided difference filters with the additive process and measurement noises using the unified way of the innovation vector approach.

The first-order divided difference filter (DDF1) is derived for general discrete-time nonlinear equations

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{w}_k, k) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k, k) \end{aligned} \quad (3.79)$$

where $\mathbf{x}_k \in \mathfrak{R}^n$ is the $n \times 1$ state vector, $\mathbf{y}_k \in \mathfrak{R}^m$ is the $m \times 1$ observation vector. $\mathbf{w}_k \in \mathfrak{R}^q$ is the $q \times 1$ state noise process vector and $\mathbf{v}_k \in \mathfrak{R}^r$ is the $r \times 1$ measurement noise vector. It is assumed that the noise vectors are uncorrelated white Gaussian

processes with expected means and covariances

$$\begin{aligned} E \{ \mathbf{w}_k \} &= \bar{\mathbf{w}}, \quad E \left\{ [\mathbf{w}_k - \bar{\mathbf{w}}_k] [\mathbf{w}_j - \bar{\mathbf{w}}_k]^T \right\} = \mathbf{Q}_k \\ E \{ \mathbf{v}_k \} &= \bar{\mathbf{v}}, \quad E \left\{ [\mathbf{v}_k - \bar{\mathbf{v}}_k] [\mathbf{v}_j - \bar{\mathbf{v}}_k]^T \right\} = \mathbf{R}_k \end{aligned} \quad (3.80)$$

The DDF1 is formulated by using the first-order approximation represented. Conceptually the filter takes the same predictor-corrector structure in the EKF. First, the following square Cholesky factorizations are introduced

$$\mathbf{P}_0 = \mathbf{S}_x \mathbf{S}_x^T, \quad \mathbf{Q} = \mathbf{S}_w \mathbf{S}_w^T \quad (3.81)$$

As for the state propagation step, the predicted state vector $\hat{\mathbf{x}}_{k+1}^-$ is computed using Eq. (3.22), which is the same as for the EKF as follows

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}, k) \quad (3.82)$$

The predicted state covariance \mathbf{P}_{k+1}^- is determined by the symmetric matrix product

$$\mathbf{P}_{k+1}^- = \mathbf{S}_x^-(k+1) (\mathbf{S}_x^-(k+1))^T \quad (3.83)$$

where $\mathbf{S}_x^-(k+1) = \left[\mathbf{S}_{x\hat{x}}^{(1)}(k+1) \mathbf{S}_{xw}^{(1)}(k+1) \right]$ and each term is given by

$$\begin{aligned} \mathbf{S}_{x\hat{x}}^{(1)}(k+1) &= \frac{1}{2h} \{ \mathbf{f}_i(\hat{\mathbf{x}}_k + h\mathbf{s}_{x,j}, \bar{\mathbf{w}}_k) - \mathbf{f}_i(\hat{\mathbf{x}}_k - h\mathbf{s}_{x,j}, \bar{\mathbf{w}}_k) \} \\ \mathbf{S}_{xw}^{(1)}(k+1) &= \frac{1}{2h} \{ \mathbf{f}_i(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k + h\mathbf{s}_{w,j}) - \mathbf{f}_i(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k - h\mathbf{s}_{w,j}) \} \end{aligned} \quad (3.84)$$

where $\mathbf{s}_{x,j}$ is the column of \mathbf{S}_x and $\mathbf{s}_{w,j}$ is the column of \mathbf{S}_w obtained from Eq. (3.81), respectively. If the process noise vector is assumed to be simply additive, then the computation of the state covariance reduces since the derivation of the linearization matrix $\mathbf{S}_{x\hat{x}}^{(1)}(k+1)$ about the process noise \mathbf{w} is not required. Thus, the covariance

\mathbf{P}_{k+1}^- is computed as

$$\mathbf{P}_{k+1}^- = \mathbf{S}_{x\hat{x}}^{(1)}(k+1) \left(\mathbf{S}_{x\hat{x}}^{(1)}(k+1) \right)^T + \mathbf{Q}_{k+1} \quad (3.85)$$

Next, the square Cholesky factorizations are performed

$$\mathbf{P}_{k+1}^- = \mathbf{S}_x^- \mathbf{S}_x^{-T}, \quad \mathbf{R} = \mathbf{S}_v \mathbf{S}_v^T \quad (3.86)$$

The predicted observation vector $\hat{\mathbf{y}}_{k+1}^-$ and its predicted covariance are calculated in a similar fashion

$$\hat{\mathbf{y}}_{k+1}^- = \mathbf{h} \left(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1}, k+1 \right) \quad (3.87)$$

$$\mathbf{P}_{k+1}^{vv} = \mathbf{S}_v(k+1) \mathbf{S}_v^T(k+1) \quad (3.88)$$

where

$$\mathbf{S}_v(k+1) = \left[\mathbf{S}_{y\hat{x}}^{(1)}(k+1) \quad \mathbf{S}_{yv}^{(1)}(k+1) \right] \quad (3.89)$$

$$\begin{aligned} \mathbf{S}_{y\hat{x}}^{(1)}(k+1) &= \frac{1}{2h} \left\{ \mathbf{h}_i \left(\hat{\mathbf{x}}_{k+1}^- + h \mathbf{s}_{x,j}^-, \bar{\mathbf{v}}_{k+1} \right) - \mathbf{h}_i \left(\hat{\mathbf{x}}_{k+1}^- - h \mathbf{s}_{x,j}^-, \bar{\mathbf{v}}_{k+1} \right) \right\} \\ \mathbf{S}_{yv}^{(1)}(k+1) &= \frac{1}{2h} \left\{ \mathbf{h}_i \left(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1} + h \mathbf{s}_{v,j} \right) - \mathbf{h}_i \left(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1} - h \mathbf{s}_{v,j} \right) \right\} \end{aligned} \quad (3.90)$$

where $\mathbf{s}_{x,j}^-$ is the column of \mathbf{S}_x^- and $\mathbf{s}_{v,j}$ is the column of \mathbf{S}_v . Note that if it is assumed that the measurement noise vector is simply additive then the computation of the innovation covariance becomes easier since the derivation of the linearization matrix $\mathbf{S}_{yv}^{(1)}(k+1)$ about the measurement noise \mathbf{v} is not required. Thus, the innovation covariance \mathbf{P}_{k+1}^{vv} is computed as

$$\mathbf{P}_{k+1}^{vv} = \mathbf{P}_{k+1}^{yy} + \mathbf{R}_{k+1} \quad (3.91)$$

where

$$\mathbf{P}_{k+1}^{yy} = \mathbf{S}_{y\hat{x}}^{(1)}(k+1) \left(\mathbf{S}_{y\hat{x}}^{(1)}(k+1) \right)^T \quad (3.92)$$

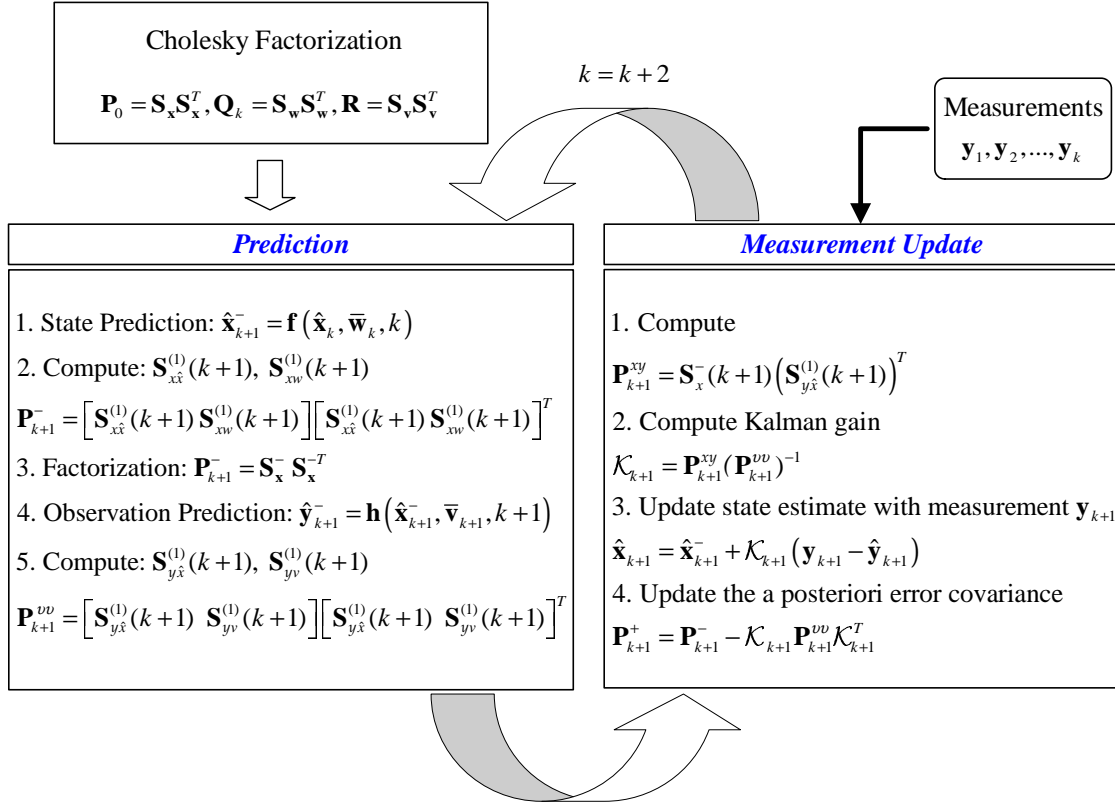


Fig. 3.5 Diagram of Predictor-Corrector Form of the Divided Difference Filter

Finally, the cross correlation matrix is determined by using

$$\mathbf{P}_{k+1}^{xy} = \mathbf{S}_x^-(k+1) \left(\mathbf{S}_{y\hat{x}}^{(1)}(k+1) \right)^T \quad (3.93)$$

The recursive Bayesian relations of the predictor-corrector structure for the unscented Kalman filter can be represented by the block diagram as shown in Fig. 3.5.

Note that the rectangular matrices $\mathbf{S}_x^-(k+1)$ and $\mathbf{S}_v(k+1)$ need to be transformed into the square Cholesky factors. This can be achieved through Householder triangularization.⁵⁴ Meanwhile, in the update process the filter gain \mathcal{K}_{k+1} , the estimated state vector $\hat{\mathbf{x}}_{k+1}^+$, and updated covariance \mathbf{P}_{k+1}^+ can be computed with the same formulas used in the EKF (Table 3.1). The DDF1 algorithm is described in Table 3.3.

3.4.5 Second-Order Divided Difference Filter

The second-order divided difference filter (DDF2) is obtained by using the calculation of the mean and covariance in the second-order polynomial approximation section. First, the following additional matrices containing divided difference are defined

$$\begin{aligned}\mathbf{S}_{x\hat{x}}^{(2)}(k+1) &= \frac{\sqrt{\gamma-1}}{2\gamma} \{ \mathbf{f}_i(\hat{\mathbf{x}}_k + h\mathbf{s}_{x,j}, \bar{\mathbf{w}}_k) + \mathbf{f}_i(\hat{\mathbf{x}}_k - h\mathbf{s}_{x,j}, \bar{\mathbf{w}}_k) - 2\mathbf{f}_i(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k) \} \\ \mathbf{S}_{xw}^{(2)}(k+1) &= \frac{\sqrt{\gamma-1}}{2\gamma} \{ \mathbf{f}_i(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k + h\mathbf{s}_{w,j}) + \mathbf{f}_i(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k - h\mathbf{s}_{w,j}) - 2\mathbf{f}_i(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k) \}\end{aligned}\quad (3.94)$$

where $\mathbf{s}_{x,j}$ is the j th column of \mathbf{S}_x , $\mathbf{s}_{w,j}$ is the j th column of \mathbf{S}_w , and $\gamma = h^2$ is a constant parameter.

The predicted state equation is given by

$$\begin{aligned}\hat{\mathbf{x}}_{k+1}^- &= \frac{\gamma - (n_x + n_w)}{\gamma} \mathbf{f}(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k) \\ &+ \frac{1}{2\gamma} \sum_{p=1}^{n_x} \{ \mathbf{f}(\hat{\mathbf{x}}_k + h\mathbf{s}_{x,p}, \bar{\mathbf{w}}_k) + \mathbf{f}(\hat{\mathbf{x}}_k - h\mathbf{s}_{x,p}, \bar{\mathbf{w}}_k) \} \\ &+ \frac{1}{2\gamma} \sum_{p=1}^{n_w} \{ \mathbf{f}(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k + h\mathbf{s}_{w,p}) + \mathbf{f}(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k - h\mathbf{s}_{w,p}) \}\end{aligned}\quad (3.95)$$

where n_x denotes the dimension of the state vector, and n_w is the dimension of process noise vector. It turned out that the prediction accuracy of the state estimate in the DDF2 is identical to that of the UKF.¹² A triangular Cholesky factor of the predicted covariance is obtained by the House-holder transformation of the compound matrix

$$\mathbf{S}_x^-(k+1) = \begin{bmatrix} \mathbf{S}_{x\hat{x}}^{(1)}(k+1) & \mathbf{S}_{xw}^{(1)}(k+1) & \mathbf{S}_{x\hat{x}}^{(2)}(k+1) & \mathbf{S}_{xw}^{(2)}(k+1) \end{bmatrix} \quad (3.96)$$

The predicted covariance is computed

$$\mathbf{P}_{k+1}^- = \mathbf{S}_x^-(k+1) (\mathbf{S}_x^-(k+1))^T \quad (3.97)$$

Theoretically, it is proved that the accuracy of the covariance prediction in the DDF2

Table 3.3 First-Order Divided Difference Filter (DDF) Algorithm

Initialization:

$$\hat{\mathbf{x}}_k = E[\mathbf{x}_k], \mathbf{P}_k = E \left[(\mathbf{x}_k - \hat{\mathbf{x}}_k) (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \right]$$

Square Cholesky Factorizations:

$$\mathbf{P}_k = \mathbf{S}_x \mathbf{S}_x^T, \mathbf{Q} = \mathbf{S}_w \mathbf{S}_w^T, \mathbf{R} = \mathbf{S}_v \mathbf{S}_v^T$$

$$\mathbf{S}_{x\hat{x}}^{(1)}(k+1) = \frac{1}{2h} \{ \mathbf{f}_i(\hat{\mathbf{x}}_k + h\mathbf{s}_{x,j}, \bar{\mathbf{w}}_k) - \mathbf{f}_i(\hat{\mathbf{x}}_k - h\mathbf{s}_{x,j}, \bar{\mathbf{w}}_k) \}$$

$$\mathbf{S}_{xw}^{(1)}(k+1) = \frac{1}{2h} \{ \mathbf{f}_i(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k + h\mathbf{s}_{w,j}) - \mathbf{f}_i(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k - h\mathbf{s}_{w,j}) \}$$

$$\mathbf{S}_x^-(k+1) = \left[\mathbf{S}_{x\hat{x}}^{(1)}(k+1) \mathbf{S}_{xw}^{(1)}(k+1) \right]$$

State and Covariance Propagation:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}, k)$$

$$\mathbf{P}_{k+1}^- = \mathbf{S}_x^-(k+1) (\mathbf{S}_x^-(k+1))^T$$

$$\mathbf{S}_{y\hat{x}}^{(1)}(k+1) = \frac{1}{2h} \{ \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^- + h\mathbf{s}_{x,j}^-, \bar{\mathbf{v}}_{k+1}) - \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^- - h\mathbf{s}_{x,j}^-, \bar{\mathbf{v}}_{k+1}) \}$$

$$\mathbf{S}_{yv}^{(1)}(k+1) = \frac{1}{2h} \{ \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1} + h\mathbf{s}_{v,j}) - \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1} - h\mathbf{s}_{v,j}) \}$$

$$\mathbf{S}_v(k+1) = \left[\mathbf{S}_{y\hat{x}}^{(1)}(k+1) \mathbf{S}_{yv}^{(1)}(k+1) \right]$$

Observation and Innovation Covariance Propagation:

$$\hat{\mathbf{y}}_{k+1}^- = \mathbf{h}(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1}, k+1)$$

$$\mathbf{P}_{k+1}^{vv} = \mathbf{S}_v(k+1) \mathbf{S}_v^T(k+1)$$

$$\mathbf{P}_{k+1}^{xy} = \mathbf{S}_x^-(k+1) \left(\mathbf{S}_{y\hat{x}}^{(1)}(k+1) \right)^T$$

Update:

$$\mathcal{K}_{k+1} = \mathbf{P}_{k+1}^{xy} (\mathbf{P}_{k+1}^{vv})^{-1}$$

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathcal{K}_{k+1} \mathbf{P}_{k+1}^{vv} \mathcal{K}_{k+1}^T$$

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathcal{K}_{k+1} \mathbf{v}_{k+1}$$

is close to that of the UKF (see Ref. 12 for details).

Next, the predicted observation vector $\hat{\mathbf{y}}_{k+1}^-$ and its predicted covariance are calculated in a similar fashion

$$\begin{aligned}\hat{\mathbf{y}}_{k+1}^- &= \frac{\gamma - (n_x + n_v)}{\gamma} \mathbf{h}(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1}) \\ &+ \frac{1}{2\gamma} \sum_{p=1}^{n_x} \{ \mathbf{h}(\hat{\mathbf{x}}_{k+1}^- + h\mathbf{s}_{x,p}^-, \bar{\mathbf{v}}_{k+1}) + \mathbf{h}(\hat{\mathbf{x}}_{k+1}^- - h\mathbf{s}_{x,p}^-, \bar{\mathbf{v}}_{k+1}) \} \\ &+ \frac{1}{2\gamma} \sum_{p=1}^{n_v} \{ \mathbf{h}(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1} + h\mathbf{s}_{v,p}) + \mathbf{h}(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1} - h\mathbf{s}_{v,p}) \}\end{aligned}\quad (3.98)$$

where n_v is the dimension of the measurement noise vector, $\mathbf{s}_{x,p}^-$ is the p th column of \mathbf{S}_x^- , and $\mathbf{s}_{v,p}$ is the p th column of \mathbf{S}_v . The innovation covariance matrix is given by

$$\mathbf{P}_{k+1}^{vv} = \mathbf{S}_v(k+1)\mathbf{S}_v^T(k+1) \quad (3.99)$$

where $\mathbf{S}_v(k+1)$ is the compound matrix

$$\mathbf{S}_v(k+1) = \begin{bmatrix} \mathbf{S}_{x\hat{x}}^{(1)}(k+1) & \mathbf{S}_{xw}^{(1)}(k+1) & \mathbf{S}_{x\hat{x}}^{(2)}(k+1) & \mathbf{S}_{xw}^{(2)}(k+1) \end{bmatrix} \quad (3.100)$$

$$\begin{aligned}\mathbf{S}_{y\hat{x}}^{(2)}(k+1) &= \frac{\sqrt{\gamma-1}}{2\gamma} \{ \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^- + h\mathbf{s}_{x,j}^-, \bar{\mathbf{v}}_{k+1}) + \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^- - h\mathbf{s}_{x,j}^-, \bar{\mathbf{v}}_{k+1}) \\ &\quad - 2\mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1}) \} \\ \mathbf{S}_{yv}^{(2)}(k+1) &= \frac{\sqrt{\gamma-1}}{2\gamma} \{ \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1} + h\mathbf{s}_{v,j}) + \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1} - h\mathbf{s}_{v,j}) \\ &\quad - 2\mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1}) \}\end{aligned}\quad (3.101)$$

The cross correlation matrix is the same as for the DDF1

$$\mathbf{P}_{k+1}^{xy} = \mathbf{S}_x^-(k+1) \left(\mathbf{S}_{y\hat{x}}^{(1)}(k+1) \right)^T \quad (3.102)$$

The DDF2 algorithm can also be described in the unified way used in Table 3.3 by replacing the first-order prediction formulas for the state and covariance with the second-order ones.

3.5 Finite Difference Filter

It was noted previously that in principle an optimal nonlinear filtering (NF) algorithm for the state estimation is achieved when the posterior conditional density function is obtained from the recursive Bayesian relations, the predictor and the corrector equations. For example, for discrete-discrete system dynamics and measurements, the *Chapman-Kolmogorov* equation¹ is used as a predictor for propagating the state probability density between measurements, and the powerful *Bayes' formula* as a corrector calculates the posterior density from the prior density and new measurements.⁴

However, an optimal filtering algorithm is intractable because in general the state probability density function (PDF) involves infinite dimensional integrations. Thus approximations to the optimal nonlinear filter must be adopted. In this section, a direct numerical approximation to the optimal nonlinear filter is investigated. A direct numerical approximation to the optimal nonlinear filter is to calculate the state conditional probability density by applying fast solvers to the *Kushner-Stratonovich* (or *Zakai*) equation in the case of continuous-continuous system models^{32-34,55} or to the *Fokker-Planck* equation in continuous-discrete system models.³⁵⁻³⁸ Then, quantities of interest such as the minimum mean square error (MMSE), the covariance, or the maximum likelihood state estimate can be constructed from the approximated posterior density.

When system dynamics are governed by a stochastic differential equation, the time evolution of the state probability density satisfies the Fokker-Planck equation, which is discretized on a grid to model nonlinearities such as non-Gaussian and multimodal behavior.³⁷ To model this time evolution numerically, the density must be truncated so that it can be modeled using a finite number of degrees of freedom. Conceptually, there are many ways^{56,57} to do this, such as projecting the density onto a

collection of functions with tractable time evolution, convolution methods, or wavelet methods.

This chapter describes computationally efficient approximate finite dimensional nonlinear filter for the state estimation of nonlinear systems with continuous-time state dynamics and discrete-time measurements. The Fokker-Planck equation can be solved accurately and efficiently using finite difference schemes.^{13,14} For discrete time measurement updates Bayes' formula can be applied recursively to combine measurement likelihoods derived from physical sensor models with the target state probability density. Measurements are used as soon as they become available to yield a real-time algorithm.

Finite difference methods can be broadly categorized as so called explicit or implicit schemes. One feature of explicit methods is that they can be easily solved with computational complexity that is proportional to the size of the grid. Explicit methods are simpler to implement than implicit schemes, but they suffer from the drawbacks that they are less accurate and can have poor numerical stability.⁵⁸ On the other hand, implicit schemes can lead to fast and accurate solutions with no greater effort than is required for an explicit solution scheme. In general, in implicit methods the inversion step can be complicated, but the inversion problem encountered in NF can be solved by using a type of alternative direction implicit scheme called the Dyakonov method.⁵⁸ To illustrate the broad applicability of this method, it is applied to a target tracking problem.

3.5.1 Exact Nonlinear Filter

3.5.1.1 Fokker-Planck Equation

The starting point for modeling time evolution in NLF is exactly the same as it is for Kalman filtering. The system dynamic model is represented by the Ito stochastic differential equation.¹ The Ito equation characterizes how target states and their probability densities evolve in time due to deterministic and random target motion effects. For the time-dependent target state \mathbf{x}_t , the continuous-time Ito stochastic equation is described by

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t) dt + \mathbf{G}(\mathbf{x}_t, t) d\beta_t, \quad t \geq t_0 \quad (3.103)$$

where $\mathbf{x}_t \in \Re^{n \times 1}$ represents the state vector of the system, $\mathbf{f} \in \Re^{n \times 1}$ is a nonlinear vector valued function, $\mathbf{G} \in \Re^{n \times r}$ is a real matrix, and $d\beta_t \in \Re^{r \times 1}$ is a vector Brownian motion process with $E\{d\beta_t d\beta_t^T\} = \mathbf{Q}(t) dt$. Note that in the Ito equation, \mathbf{f} characterizes the deterministic part of the dynamics while \mathbf{G} represents the part of the dynamics modeled as random. Suppose noisy measurements are taken from the discrete-time nonlinear equation given by

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_{t_k}, \mathbf{v}_k, k) \quad (3.104)$$

where $\mathbf{v}_k \in \Re^{m \times 1}$ is a zero-mean white Gaussian noise with covariance matrix \mathbf{R}_k . Define the collection of measurements taken up to current time t_k as $\mathbf{Y}_k = \{y_1, y_2, \dots, y_k\}$

Because the probability density function (PDF) summarizes all the statistical information about the state conditioned on the measurement \mathbf{Y}_k , NLF seeks the entire probability density function. It turns out that under fairly general assumptions¹ that the prior density for the system exists and is continuously differentiable with respect to time t_k and the state vector \mathbf{x}_{t_k} , the time evolution of the conditional den-

sity $p(\mathbf{x}_{t_k}|\mathbf{Y}_{k-1})$ satisfies the *Fokker-Planck* (FP) equation or *Kolmogorov's forward* equation

$$\frac{\partial}{\partial t}p(\mathbf{x}_{t_k}|\mathbf{Y}_{k-1}) = \mathcal{L}(p(\mathbf{x}_{t_k}|\mathbf{Y}_{k-1})) \quad (3.105)$$

where \mathcal{L} is the forward diffusion operator defined by

$$\mathcal{L}(p) \equiv -\sum_{i=1}^n \frac{\partial [\mathbf{f}_i p]}{\partial x_i} + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 [(\mathbf{G}\mathbf{Q}\mathbf{G}^T)_{ij} p]}{\partial x_i \partial x_j} \quad (3.106)$$

and the initial condition is given by $p(\mathbf{x}_{t_{k-1}}|\mathbf{Y}_{k-1})$. The first-order deterministic term is called the *drift* or *advective* term and the second-order derivative term is called *diffusion*. For this reason, FPEs are sometimes referred to as advective-diffusion equations.¹ Note that for discrete-time system dynamics and measurement equations the Fokker-Planck equation is replaced with the Chapman-Kolmogorov equation explained in Eq. (1.6) to propagate the state probability density function in time. On the other hand, for continuous-time system dynamics with continuous measurements the time evolution of the probability density function associated with the state is given by the solution to the Kushner-Stratonovich equation.^{33,34}

The Ito stochastic differential equation and the FPE applied directly to densities capture precisely the same information about the system motion, thus the solution to the FPE is the tool used for NLF calculations. The Markov nature of the Ito equation is reflected in the fact that the FPE is first order in time. Thus, all information about the state \mathbf{x}_{t_k} conditioned on the measurement \mathbf{Y}_k is completely described in terms of the instantaneous conditional PDF $p(\mathbf{x}_{t_k}|\mathbf{Y}_k)$.

3.5.1.2 Measurement Update

The target probability density depends on measurements through the likelihood $p(\mathbf{y}_k|\mathbf{x}_{t_k})$. This is a physical model for the probability to obtain measurement \mathbf{y}_k

given that the state is \mathbf{x}_{t_k} . When the measurements are conditionally independent and depend only on the instantaneous state of the target, then given a new observation \mathbf{y}_k , the measurement updated conditional density $p(\mathbf{x}_{t_k}|\mathbf{y}_k)$ is obtained from the predicted density $p(\mathbf{x}_{t_k}|\mathbf{Y}_{k-1})$ using Bayes' formula¹

$$p(\mathbf{x}_{t_k}|\mathbf{Y}_k) = \frac{p(\mathbf{y}_k|\mathbf{x}_{t_k}) p(\mathbf{x}_{t_k}|\mathbf{Y}_{k-1})}{\int p(\mathbf{y}_k|\mathbf{x}_{t_k}) p(\mathbf{x}_{t_k}|\mathbf{Y}_{k-1}) d\mathbf{x}_{t_k}} \quad (3.107)$$

Equations (3.105) and (3.107) represent the predictor and corrector equations for the exact optimal nonlinear filtering. The FPE in Eq. (3.105) is used to propagate the density function, while the measurement update formula in Eq. (3.107) is used to update the information about the state. Therefore, the exact optimal nonlinear filtering algorithm consists of a partial differential equation (the Fokker-Planck equation) that describes the time evolution of the conditional density between measurements, and a difference equation (Bayes' formula) that describes how the information of the density is updated by new measurements.

3.5.2 Approximate Nonlinear Filter

General closed form solutions of the predictor-corrector relations to optimal nonlinear filtering are intractable, thus approximate numerical methods must be adopted. Approximate nonlinear filtering methods have been proposed by many researchers^{14,37,38} to obtain the time evolution of the conditional density. In this section an efficient finite difference method^{13,14} that is used to solve the Fokker-Planck equation whose solution is the conditional density between measurements is investigated. For conceptual illustration a simple target tracking system model is considered to build intuition about the finite difference filtering algorithm.

Consider vehicle motion in a plane with nearly coordinated turns (CT) for a

higher dimensional model of FPE. The target state vector $\mathbf{x} \in \Re^{5 \times 1}$ is given by

$$\mathbf{x} = [x, \dot{x}, y, \dot{y}, \omega]^T \quad (3.108)$$

The Ito stochastic differential equation is described by

$$d \begin{bmatrix} x_t \\ \dot{x}_t \\ y_t \\ \dot{y}_t \\ \omega_t \end{bmatrix} = \begin{bmatrix} \dot{x} \\ -\omega \dot{y} \\ \dot{y} \\ \omega \dot{x} \\ 0 \end{bmatrix} dt + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d\beta_{\dot{x}} \\ d\beta_{\dot{y}} \\ d\beta_{\omega} \end{bmatrix} \quad (3.109)$$

where ω is the rotation rate about the vertical axis, and the covariance matrix $\mathbf{Q}(t)$ from $E \{d\beta_t d\beta_t^T\} = \mathbf{Q}(t)dt$ is

$$\mathbf{Q}(t) = \begin{bmatrix} q_{\dot{x}} & 0 & 0 \\ 0 & q_{\dot{y}} & 0 \\ 0 & 0 & q_{\omega} \end{bmatrix} \quad (3.110)$$

The nonlinear system function $\mathbf{f}(\mathbf{x}, t)$ from the Ito equation is written by

$$\mathbf{f} = \begin{bmatrix} \dot{x} \\ -\omega \dot{y} \\ \dot{y} \\ \omega \dot{x} \\ 0 \end{bmatrix} \quad (3.111)$$

and the random component \mathbf{G} of the model is driven by a 3D Brownian process noise

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.112)$$

Then, the resulting FPE is expressed by

$$\frac{\partial p}{\partial t} = -\dot{x} \frac{\partial p}{\partial x} - \dot{y} \frac{\partial p}{\partial y} + \omega \dot{y} \frac{\partial p}{\partial \dot{x}} - \omega \dot{x} \frac{\partial p}{\partial \dot{y}} + \frac{q_{\dot{x}}}{2} \frac{\partial^2 p}{\partial \dot{x}^2} + \frac{q_{\dot{y}}}{2} \frac{\partial^2 p}{\partial \dot{y}^2} + \frac{q_{\omega}}{2} \frac{\partial^2 p}{\partial \omega^2} \quad (3.113)$$

The first four terms are a collection of one-way wave equations propagating the density down the x , y , \dot{x} , and \dot{y} axes with velocities \dot{x} , \dot{y} , $-\omega \dot{y}$, and $\omega \dot{x}$, respectively. The remaining three terms generate diffusions along the \dot{x} , \dot{y} , and ω axes. There is no diffusion along the x and y axes.

3.5.2.1 Prediction Equation

The following algorithm is based on the finite difference numerical method proposed by Kastella.¹⁴ For a finite difference filtering (FDF) algorithm, the alternating direction implicit (ADI) scheme⁵⁸ is used to solve the Fokker-Planck equation (FPE). Space and time are discretized on a uniform grid with time resolution Δt and spatial resolution $\Delta \mathbf{x} = [\Delta x, \Delta \dot{x}, \Delta y, \Delta \dot{y}, \Delta \omega]^T$. Then, the FPE can be expressed by

$$\frac{\partial p}{\partial t} = \sum_i A_i p \quad (3.114)$$

where $p(\mathbf{x}, t)$ is a solution to the FPE subject to an appropriate boundary condition, and the operators A_i are defined by

$$A_1 = -\dot{x} \frac{\partial}{\partial x} \quad (3.115)$$

$$A_2 = \frac{q_{\dot{x}}}{2} \frac{\partial^2}{\partial \dot{x}^2} + \omega \dot{y} \frac{\partial}{\partial \dot{x}} \quad (3.116)$$

$$A_3 = -\dot{y} \frac{\partial}{\partial y} \quad (3.117)$$

$$A_4 = \frac{q_{\dot{y}}}{2} \frac{\partial^2}{\partial \dot{y}^2} - \omega \dot{x} \frac{\partial}{\partial \dot{y}} \quad (3.118)$$

$$A_5 = \frac{q_{\omega}}{2} \frac{\partial^2}{\partial \omega^2} \quad (3.119)$$

Dyakonov Scheme: The Crank-Nicholson scheme⁵⁸ is used to approximate the time derivative for the FPE, which is obtained by using a Taylor series in time for $p(\mathbf{x}, t_k + \Delta t/2)$ leading to

$$\frac{p^{k+1} - p^k}{\Delta t} = \frac{1}{2} \sum_i A_i p^{k+1} + \frac{1}{2} \sum_i A_i p^k + O(\Delta t^2) \quad (3.120)$$

where $O(\Delta t^2)$ represents remainder terms from the truncation of the Taylor series that are proportional to (Δt^2) . This can be rearranged to yield

$$\left(1 - \frac{\Delta t}{2} \sum_i A_i\right) p^{k+1} = \left(1 + \frac{\Delta t}{2} \sum_i A_i\right) p^k + O(\Delta t^3) \quad (3.121)$$

Direct inversion of this expression is computationally expensive, thus an equivalent expression that is easy to invert is obtained by using the operator product identity

$$\prod_i \left(1 \pm \frac{\Delta t}{2} A_i\right) = 1 \pm \frac{\Delta t}{2} \sum_i A_i + \left(\frac{\Delta t}{2}\right)^2 \sum_{i < j} A_i A_j + O(\Delta t^3) \quad (3.122)$$

and the fact $p^{k+1} - p^k = O(\Delta t)$ yields

$$\prod_i \left(1 - \frac{\Delta t}{2} A_i\right) p^{k+1} = \prod_i \left(1 + \frac{\Delta t}{2} A_i\right) p^k + O(\Delta t^3) \quad (3.123)$$

which plays a key role in the finite difference method.¹⁴ It is much easier to solve numerically, and no additional approximation error on the grid is incurred in this factorization.

Now, consider space discretization $A_{i\Delta\mathbf{x}}$ of the operator A_i

$$\prod_i \left(1 - \frac{\Delta t}{2} A_{i\Delta\mathbf{x}}\right) p^{k+1} = \prod_i \left(1 + \frac{\Delta t}{2} A_{i\Delta\mathbf{x}}\right) p^k + O(\Delta t^3) + O(\Delta\mathbf{x}^2 \Delta t) \quad (3.124)$$

If \mathcal{P} denotes an approximation to the density p defined on the grid, it leads to

$$\prod_i \left(1 - \frac{\Delta t}{2} A_{i\Delta\mathbf{x}}\right) \mathcal{P}^{k+1} = \prod_i \left(1 + \frac{\Delta t}{2} A_{i\Delta\mathbf{x}}\right) \mathcal{P}^k \quad (3.125)$$

To propagate the density the above equation is solved for \mathcal{P}^{k+1} . Let N_A denote the number of operators A_i in the FPE (for example, $N_A = 5$, for CT model). Then the *Dyakonov scheme* is expressed by¹⁴

$$\tilde{\mathcal{P}}^k = \prod_i \left(1 + \frac{\Delta t}{2} A_{i\Delta\mathbf{x}}\right) \mathcal{P}^k \quad (3.126)$$

$$\left(1 - \frac{\Delta t}{2} A_{i\Delta\mathbf{x}}\right) \tilde{\mathcal{P}}^{k+i/N_A} = \tilde{\mathcal{P}}^{k+(i-1)/N_A} \quad (3.127)$$

$$\mathcal{P}^{k+1} = \tilde{\mathcal{P}}^{k+1} \quad (3.128)$$

where $\tilde{\mathcal{P}}^{k+i/N_A}$ is an intermediate result, and $i = 1, \dots, N_A$. The key idea in the Dyakonov scheme comes from the fact that each factor $(1 - \Delta t/2 A_{i\Delta\mathbf{x}})$ is easily inverted separately, simplifying the calculation.

The operators A_i is discretized, and the following abbreviate is defined

$$\mathcal{P}(k\Delta t, x \pm \Delta x, \dot{x}, y, \dot{y}, \omega) \equiv \mathcal{P}_{x \pm \Delta x} \quad (3.129)$$

Similar definitions for other terms $\mathcal{P}_{\dot{x} \pm \Delta \dot{x}}, \dots, \mathcal{P}_{\omega \pm \Delta \omega}$ are defined in the same way.

Using central difference for the spatial derivatives in the CT model leads to

$$A_{1\Delta x}\mathcal{P} = -\frac{\dot{x}}{2\Delta x}(\mathcal{P}_{x+\Delta x} - \mathcal{P}_{x-\Delta x}) \quad (3.130)$$

$$A_{2\Delta x}\mathcal{P} = -\frac{q_{\dot{x}}}{2\Delta \dot{x}^2}(\mathcal{P}_{\dot{x}+\Delta \dot{x}} - 2\mathcal{P} + \mathcal{P}_{\dot{x}-\Delta \dot{x}}) + \frac{\omega \dot{y}}{2\Delta \dot{x}}(\mathcal{P}_{\dot{x}+\Delta \dot{x}} - \mathcal{P}_{\dot{x}-\Delta \dot{x}}) \quad (3.131)$$

$$A_{3\Delta x}\mathcal{P} = -\frac{\dot{y}}{2\Delta y}(\mathcal{P}_{y+\Delta y} - \mathcal{P}_{y-\Delta y}) \quad (3.132)$$

$$A_{4\Delta x}\mathcal{P} = \frac{q_{\dot{y}}}{2\Delta \dot{y}^2}(\mathcal{P}_{\dot{y}+\Delta \dot{y}} - 2\mathcal{P} + \mathcal{P}_{\dot{y}-\Delta \dot{y}}) - \frac{\omega \dot{x}}{2\Delta \dot{y}}(\mathcal{P}_{\dot{y}+\Delta \dot{y}} - \mathcal{P}_{\dot{y}-\Delta \dot{y}}) \quad (3.133)$$

$$A_{5\Delta x}\mathcal{P} = -\frac{q_{\omega}}{2\Delta \omega^2}(\mathcal{P}_{\omega+\Delta \omega} - 2\mathcal{P} + \mathcal{P}_{\omega-\Delta \omega}) \quad (3.134)$$

Thomas Algorithm: All that remains to implement the Dyakonov method is to solve the intermediate steps in Eq. (3.125) for $\tilde{\mathcal{P}}^{k+i/N_A}$ in terms of $\tilde{\mathcal{P}}^{k+(i-1)/N_A}$. For the operators in Eqs. (3.130)~(3.134), each of the intermediate step in Eq. (3.125) is a tridiagonal system which can be solved using Thomas's algorithm.⁵⁸ Each of the intermediate step in Eq. (3.125) is a collection of 1-dimensional equations of the form

$$a_j\mathcal{P}_{j-1} + b_j\mathcal{P}_j + c_j\mathcal{P}_{j+1} = d_j, j = 1, \dots, m-1 \quad (3.135)$$

where the \mathcal{P}_j are unknowns, and a_j , b_j , c_j , and d_j are known, and boundary conditions are $\mathcal{P}_0 = \beta_0$, $\mathcal{P}_m = \beta_m$. The index j runs over one of x, \dot{x}, \dots, w for each value of i in Eq. (3.125). For example, in the CV model with $i = 1$, the solution of Eq. (3.125) involves $A_{i\Delta x}$ on the left-hand side so i corresponds to the x -index. For $i = 2$ we have $A_{i\Delta x}$ and i corresponds to the \dot{x} -index. The correspondence is similar for the CT model and Eq. (3.135) can be solve using

$$\mathcal{P}_j = r_{j+1}\mathcal{P}_{j+1} + s_j, \quad j = 0, \dots, m-1 \quad (3.136)$$

where

$$r_{j+1} = -(a_j \mathcal{P}_j + b_j)^{-1} c_j \quad (3.137)$$

$$s_{j+1} = -(a_j \mathcal{P}_j + b_j)^{-1} (d_j - a_j r_j) \quad (3.138)$$

which reduces to $\mathcal{P}_0 = r_1 + s_1$ and $\mathcal{P}_0 = \beta_0$. These can be satisfied with $r_1 = 0$ and $s_1 = \beta_0$. With these initial values, Eq. (3.138) can be used to obtain the remaining r_j and s_j , for $j = 2, \dots, m$. Then Eq. (3.136) is used to find \mathcal{P}_j , starting with $\mathcal{P}_m = \beta_m$. The Dyakonov scheme with Thomas's algorithm is utilized for solving the tridiagonal system.

3.5.2.2 Measurement Update

The measurement update equation in Bayes' relation in Eq. (3.107) is solved by using both the predicted density $p(\mathbf{x}_{t_k} | \mathbf{Y}_{k-1})$ and the measurement likelihood density $p(\mathbf{y}_k | \mathbf{x}_{t_k})$. The predicted conditional density $p(\mathbf{x}_{t_k} | \mathbf{Y}_{k-1})$ is the solution to the Fokker-Planck equation, and in this chapter the ADI finite difference numerical scheme is used to solve the partial differential equation. Suppose measurements are taken from the discrete-time nonlinear equation given in Eq. (3.104), then the vector-valued Gaussian measurement likelihood $p(\mathbf{y}_k | \mathbf{x}_{t_k}) \in \Re^{m \times 1}$ is expressed by

$$p(\mathbf{y}_k | \mathbf{x}_{t_k}) = \frac{1}{(2\pi)^{m/2} |\mathbf{R}|^{1/2}} \exp \left\{ -\frac{1}{2} [\mathbf{y}_k - \mathbf{h}(\mathbf{x}_{t_k}, k)]^T \mathbf{R}^{-1} [\mathbf{y}_k - \mathbf{h}(\mathbf{x}_{t_k}, k)] \right\} \quad (3.139)$$

Note that Eqs. (3.105) and (3.107) form the recursive predictor and corrector relations of the probability density evolution, respectively.

3.5.2.3 Finite Difference Nonlinear Filter

The finite-difference nonlinear filtering algorithm explained above can be represented in the block diagram form as shown in Fig. 3.6. In the diagram, the predicted density

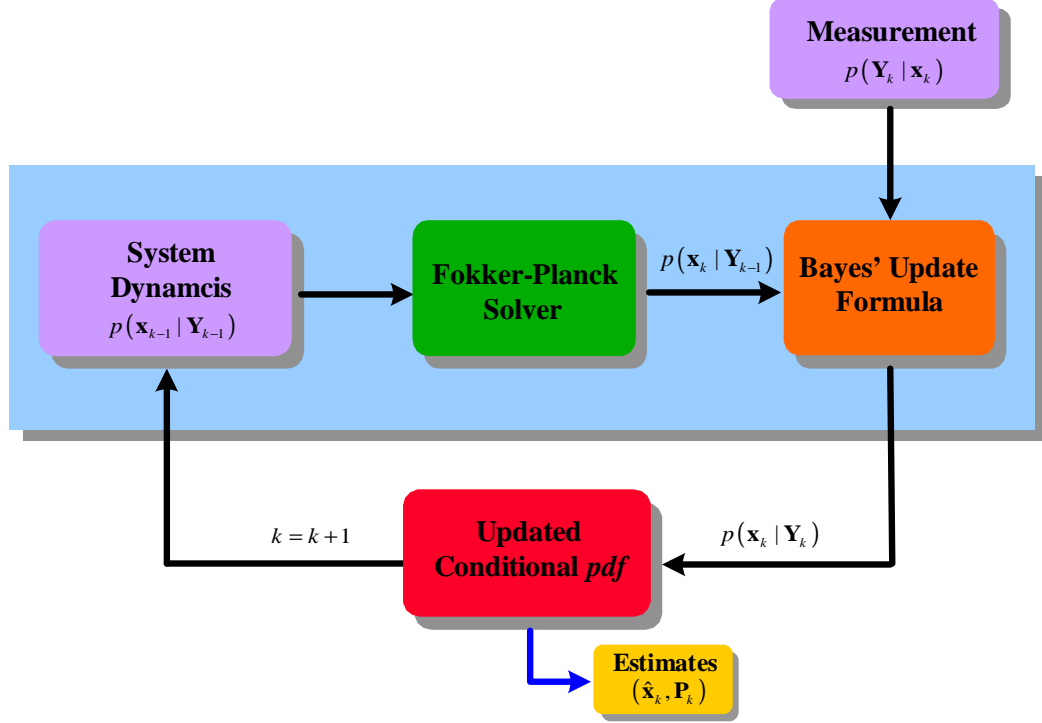


Fig. 3.6 Diagram of Finite-Difference Nonlinear Filtering

in the propagation step is obtained by using the ADI finite difference scheme, and the posterior density is computed by applying the Bayes' formula. After the conditional probability $p(\mathbf{x}_{t_k} | \mathbf{Y}_k)$ is computed, the minimum mean square error state estimate $\hat{\mathbf{x}}_{t_k}$ is computed by taking the expected value of \mathbf{x}_{t_k}

$$\hat{\mathbf{x}}_{t_k} = \int \mathbf{x}_{t_k} p(\mathbf{x}_{t_k} | \mathbf{Y}_k) d\mathbf{x}_{t_k} \quad (3.140)$$

while the covariance is computed by

$$\mathbf{P}_{t_k} = \int (\mathbf{x}_{t_k} - \hat{\mathbf{x}}_{t_k})(\mathbf{x}_{t_k} - \hat{\mathbf{x}}_{t_k})^T p(\mathbf{x}_{t_k} | \mathbf{Y}_k) d\mathbf{x}_{t_k} \quad (3.141)$$

CHAPTER IV

PARTICLE FILTERING

4.1 Introduction

For nonlinear dynamic systems, the approximated Gaussian filters introduced in the previous chapters provide an efficient algorithm for on-line filtering. For example, the Gaussian nonlinear filters include the extended Kalman filter,^{1,7} the iterated extended Kalman filter (IEKF),⁷ the Gaussian mixture filter (GMF),³ the sigma point filters (SPFs),^{8,11,12} and the finite difference filter.¹⁴ These methods are based on local linearization of the nonlinear system equations.

Since most of dynamical systems in all fields of applications such as autonomous navigation systems are nonlinear and non-Gaussian, a significant challenge to engineers and scientists is to find efficient methods for on-line, real-time estimation and prediction of the dynamical systems from the sequential observations. To date, however, there has not been a universally effective algorithm for dealing with nonlinear and non-Gaussian system. Recently, researchers have begun to pay attention to a new class of filtering methods based on the sequential Monte Carlo (SMC) approach, which is a simulation-based filter.¹⁷ Sequential Monte Carlo can be loosely defined as a set of methods that use a Monte Carlo simulation scheme in order to solve on-line estimation and prediction problems. More precisely, the sequential Monte Carlo technique achieves the filtering by recursively generating a set of weighted samples of the state variables or parameters.

Basic sequential Monte Carlo methods had been introduced in the physics, statistics, and automatic control literature.⁵⁹ However, all the earlier SMC methods^{60–62} implemented in the literature is based on only plain sequential importance sampling

(SIS) step, which forms the basis for most sequential Monte Carlo filters. The major contribution to this class of algorithm was made to be practical use by implementing a resampling stage.¹⁵ In treating dynamical systems, the sequential Monte Carlo method utilizes discrete samples to represent a complicated probability distribution and use *importance sampling*, and *weighted resampling* to complete the on-line filtering.¹⁵ After the appearance of the sequential Monte Carlo method with the resampling for nonlinear and non-Gaussian state-space models have caught attentions in vary different research fields. The sequential Monte Carlo approach is known as the bootstrap filtering,¹⁵ the condensation algorithm,¹⁹ and the particle filtering.²⁰ The flexible nature of the Monte Carlo simulations results in these methods often being more adaptive to some features of the target system.²¹ There have also been many recent modifications and improvements on the method.^{63–65} As efficient variants on the particle filter, hybrid particle filters that combine the particle filter with the standard nonlinear filters such as the EKF and the SPFs are presented.

In this dissertation, various particle filters are investigated within a unified framework of the sequential importance sampling algorithm.

4.2 Optimal Recursive Bayesian Estimation

In many filtering applications the objective is to estimate the posterior probability density for the states by making some observations. Consider the nonlinear state space model

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{\Gamma}_k \mathbf{w}_k \quad (4.1)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (4.2)$$

where $\mathbf{x}_k \in \mathfrak{R}^{n \times 1}$ is the state vector, $\mathbf{y}_k \in \mathfrak{R}^{m \times 1}$ is the observation vector, $\mathbf{w}_k \in \mathfrak{R}^{q \times 1}$ and $\mathbf{v}_k \in \mathfrak{R}^{r \times 1}$ are the process and measurement noises respectively. Let $\mathbf{X}_k = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k)$ and $\mathbf{Y}_k = (\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_k)$ be the stacked vectors of states and observations up to time step k . Assume that \mathbf{w}_k and \mathbf{v}_k are both independent and have known density distributions. In this case the state of the system is a Markov process¹⁶

$$p(\mathbf{X}_k) = p(\mathbf{x}_0) \prod_{i=1}^k p(\mathbf{x}_i | \mathbf{x}_{i-1}) \quad (4.3)$$

and the measurements \mathbf{Y}_k are conditionally independent given the states \mathbf{X}_k

$$p(\mathbf{Y}_k | \mathbf{X}_k) = \prod_{i=0}^k p(\mathbf{y}_i | \mathbf{x}_i) \quad (4.4)$$

Obviously the size of these expressions grows as time evolves if we were to calculate everything from scratch. To be able to estimate the *a posteriori* in real time, we need a way to use the estimation that we have at time k to calculate the estimation at time $k + 1$. The following recursive equations are used²²

$$p(\mathbf{x}_{k+1} | \mathbf{Y}_k) = \int p(\mathbf{x}_{k+1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Y}_k) d\mathbf{x}_k \quad (4.5)$$

$$p(\mathbf{x}_{k+1} | \mathbf{Y}_{k+1}) = \frac{p(\mathbf{y}_{k+1} | \mathbf{x}_{k+1}) p(\mathbf{x}_{k+1} | \mathbf{Y}_k)}{p(\mathbf{y}_{k+1} | \mathbf{Y}_k)} \quad (4.6)$$

The initial *a posteriori* density $p(\mathbf{x}_0 | \mathbf{y}_0)$ is obtained by

$$p(\mathbf{x}_0 | \mathbf{y}_0) = \frac{p(\mathbf{y}_0 | \mathbf{x}_0) p(\mathbf{x}_0)}{p(\mathbf{y}_0)} \quad (4.7)$$

The first equation is called the *time update* equation (or *prediction*) and the second is called the *measurement update* equation. The *likelihood* probability density $p(\mathbf{y}_k | \mathbf{x}_k)$ in Eq. (4.6) is computed by the *a priori* measurement noise density $p(\mathbf{v}_k)$ and the measurement equation. Similarly, the *state transition* density $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$ in Eq. (4.5) is calculated by using the *a priori* process noise density $p(\mathbf{w}_k)$ as well as the dynamic

equation.

For most applications, closed-form solutions for $p(\mathbf{x}_k|\mathbf{Y}_k)$ are intractable due to the integration in Eq. (4.5). Depending on the characteristics of the system, there exist different methods of estimating $p(\mathbf{x}_k|\mathbf{Y}_k)$. In a broad sense there are three different cases of filtering, namely linear Gaussian, nonlinear Gaussian, and nonlinear/Non-Gaussian. First, for a linear-Gaussian case where the system dynamic and measurement equations are linear, and the *a priori* initial state and the noise sequences are Gaussian, the recursive Bayesian estimation in Eqs. (4.5) and (4.6) leads to the Kalman filter. Second, if the system equations are nonlinear with the assumption of Gaussian distributions, the Gaussian filters provide the sub-optimal recursive algorithm. Finally, for nonlinear and non-Gaussian problems, the optimal Bayesian equations are solved by sequential Monte-Carlo methods. For intuitive understanding, loosely defined concepts of the particle filter and other filters (EKF, UKF, DDF) are illustrated in Fig. 4.1, where the sampling difference between the unscented filtering and the particle filtering is depicted. In a broad sense, the unscented Kalman filter belongs to the particle filters in that it performs sequential estimation based on a set of sampled particles drawn deterministically from probability densities. The difference is that the sigma point filters (UKF, DDF) make a Gaussian assumption to simplify the recursive Bayesian estimation whereas the particle filter performs estimation based on the form of general nonlinear models and non-Gaussian distributions.

4.3 Particle Filtering

Sequential Monte Carlo techniques achieve the filtering by recursively generating a set of weighted samples of the state variables. Each sampled particle has some kind

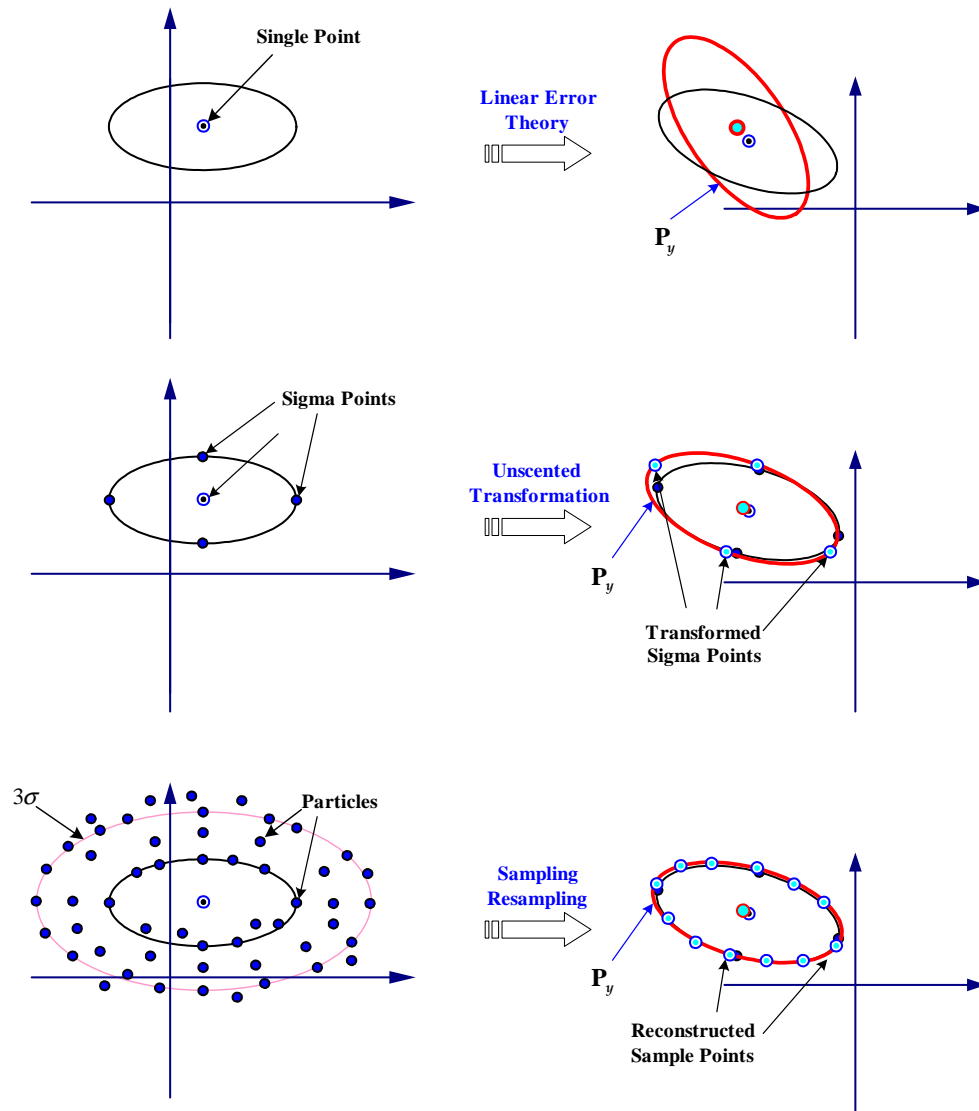


Fig. 4.1 Conceptual Description of Sampling Difference between Recursive Filtering Algorithms; EKF (Top), UKF (Middle), and PF (Bottom)

of weight indicating how well its states agree to the measurements. The samples and their corresponding weights are combined to form an estimate of the desired *posterior* distributions. For every time step the samples are more and more likely to drift away from the real state, meaning that most of the weights will tend to zero. To prevent this from happening, the samples with smaller weights will die and those with larger weights will multiply in some kind of evolution process. Therefore, the implementation of the particle filter consists of three important operations; 1) generation of particles (sampling step), 2) computation of the particle weights (importance step), and 3) resampling. The first two steps form the sequential importance sampling (SIS) algorithm. The SIS filter with the resampling is termed as the generic particle filter or SIS-R algorithm.

The theory of particle filtering is represented in a very brief manner without proofs. More details and an investigation of particle filters are available in Ref. 47.

4.3.1 Sequential Importance Sampling

Consider the discrete-time nonlinear model

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, k) + \mathbf{w}_k \quad (4.8)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k \quad (4.9)$$

where the process and measurement noises are assumed independent with known densities

$$\mathbf{w}_k \sim p_{w_k}(\cdot), \quad \mathbf{v}_k \sim p_{v_k}(\cdot) \quad (4.10)$$

For the case where the state of the system is a Markov process, any state distribution functions can be expressed by

$$p(\mathbf{X}_k) = p(\mathbf{x}_0) \prod_{i=1}^k p(\mathbf{x}_i | \mathbf{x}_{i-1}) \quad (4.11)$$

where the prior distribution of the state at time $k = 0$ is given by $p(\mathbf{x}_0)$. The observations are conditionally independent given the states

$$p(\mathbf{Y}_k | \mathbf{X}_k) = \prod_{i=0}^k p(\mathbf{y}_i | \mathbf{x}_i) \quad (4.12)$$

Since the model is neither linear nor Gaussian, the posterior cannot be represented in any other way than the total probability density function. Recursive Monte Carlo simulations give an estimate of $p(\mathbf{X}_k | \mathbf{Y}_k)$. Let us consider the recursive equation

$$p(\mathbf{X}_k | \mathbf{Y}_k) = \frac{p(\mathbf{y}_k | \mathbf{X}_k, \mathbf{Y}_{k-1}) p(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{Y}_{k-1})}{p(\mathbf{y}_k | \mathbf{Y}_{k-1})} p(\mathbf{X}_{k-1} | \mathbf{Y}_{k-1}) \quad (4.13)$$

$$= \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p(\mathbf{y}_k | \mathbf{Y}_{k-1})} p(\mathbf{X}_{k-1} | \mathbf{Y}_{k-1}) \quad (4.14)$$

where the last equality follows from the fact that the system is a Markov process as stated above.

The particle filter estimates the density function as a discretized version by utilizing a large number of samples. Let $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$ be samples drawn from the posterior. Then the expression for the estimate of the posterior is

$$\hat{p}(\mathbf{X}_k | \mathbf{Y}_k) = \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{X}_k - \mathbf{X}_k^{(i)}) \quad (4.15)$$

where $\delta(\mathbf{X}_k)$ is the *Dirac delta* function. In this case all samples are equally correct as samples from the posterior, since they are drawn from the posterior itself. Therefore, their weights in the sum are equal and can be set to one. In order to satisfy the law of total probability, the estimate has to be multiplied with $1/N$.

This estimate can be used to calculate different moments of the posterior, for example the expectation and covariance, according to

$$\begin{aligned}
\hat{\mathbf{x}} &= E \{ \mathbf{x} \} = \int \mathbf{x} p(\mathbf{x}) d\mathbf{x} \\
&= \int \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x} - \mathbf{x}^{(i)}) \mathbf{x} d\mathbf{x} \\
&= \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(i)}
\end{aligned} \tag{4.16}$$

$$\begin{aligned}
\mathbf{P} &\approx \int \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x} - \mathbf{x}^{(i)}) (\mathbf{x} - \hat{\mathbf{x}}) (\mathbf{x} - \hat{\mathbf{x}})^T d\mathbf{x} \\
&= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}^{(i)} - \hat{\mathbf{x}}) (\mathbf{x}^{(i)} - \hat{\mathbf{x}})^T
\end{aligned} \tag{4.17}$$

Now, the samples cannot be drawn from the posterior since it is unknown. Instead they are drawn from a known probability density $q(\mathbf{X}_k | \mathbf{Y}_k)$. Bayes's rule provides

$$q(\mathbf{X}_k | \mathbf{Y}_k) = q(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{Y}_k) q(\mathbf{X}_{k-1} | \mathbf{Y}_k) \tag{4.18}$$

$$= q(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{Y}_k) q(\mathbf{X}_{k-1} | \mathbf{Y}_{k-1}) \tag{4.19}$$

where the last equality is the result from the restriction that the states at time k and older are independent of the measurement at time $k-1$. This means that we can draw $\{\mathbf{x}_k^{(i)}\}_{i=1}^N$ from $q(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{Y}_k)$ and form the set $\{\mathbf{X}_k^{(i)} = \{\mathbf{X}_{k-1}^{(i)}, \mathbf{x}_k^{(i)}\}\}_{i=1}^N$ without adjusting $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$.

To use these samples to estimate the posterior each sample is associated with the so called importance weight

$$w_k^{(i)} = \frac{p(\mathbf{X}_k^{(i)} | \mathbf{Y}_k)}{q(\mathbf{X}_k^{(i)} | \mathbf{Y}_k)} = c_k \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} | \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k)} w_{k-1}^{(i)} \tag{4.20}$$

where $c_k = p(\mathbf{Y}_{k-1})/p(\mathbf{Y}_k)$. Only the relative relationship between the weights is important and c_k can therefore be neglected. This gives the weight equation

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} | \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k)} \quad (4.21)$$

A simple but efficient choice is to draw from the state propagation density

$$q(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{Y}_k) = q(\mathbf{x}_k | \mathbf{x}_{k-1}) \quad (4.22)$$

Then, the corresponding weight update equation becomes

$$w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) \quad (4.23)$$

With the samples drawn from $q(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{Y}_k)$ along with the importance weights the new estimate of the posterior $p(\mathbf{X}_k | \mathbf{Y}_k)$ is given by

$$\hat{p}(\mathbf{X}_k | \mathbf{Y}_k) = \sum_{i=1}^N \bar{w}_k^{(i)} \delta(\mathbf{X}_k - \mathbf{X}_k^{(i)}) \quad (4.24)$$

where

$$\bar{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^N w_k^{(j)}} \quad (4.25)$$

Then, the estimated mean value $\hat{\mathbf{x}}_k^+$ and covariance \mathbf{P}_k^+ are computed in terms of the current state \mathbf{x}_k and the importance weights $\bar{w}_k^{(i)}$

$$\hat{\mathbf{x}}_k^+ = E\{\mathbf{x}_k\} = \sum_{i=1}^N \bar{w}_k^{(i)} \mathbf{x}_k^{(i)} \quad (4.26)$$

$$\mathbf{P}_k^+ = E\left\{[\mathbf{x}_k - E\{\mathbf{x}_k\}][\mathbf{x}_k - E\{\mathbf{x}_k\}]^T\right\} \quad (4.27)$$

$$\approx \sum_{i=1}^N \bar{w}_k^{(i)} \left(\mathbf{x}_k^{(i)} - \hat{\mathbf{x}}_k^+\right) \left(\mathbf{x}_k^{(i)} - \hat{\mathbf{x}}_k^+\right)^T \quad (4.28)$$

4.3.2 Resampling

As time evolves the samples tend to spread and the weights will be almost zero for most of the samples,^{20,22} which means that they do not contribute much to the estimation of the posterior. It also means that the estimate in the interesting region is crude, since there are not many samples contributing. This phenomenon is known as the degeneracy problem in the SIS particle filter.²¹ In the next sections we investigate methods to check when this is happening, and how to solve the problem.

4.3.2.1 Effective Sample Size

Effective sample size is a way to measure how well the samples are concentrated in the interesting region.⁶⁶ By comparing the covariance of a set of samples drawn from the posterior and the covariance obtained through the use of importance sampling we will get a measurement of the sampling efficiency. This in turn will give an expression for the effective sample size. It is shown that the effective sample size can be estimated by^{67,68}

$$\hat{N}_{\text{eff}} \approx \frac{1}{\sum_i \left(\bar{w}_k^{(i)} \right)^2} \quad (4.29)$$

If all the weights are equal, the effective sample size will be N . One way to decide when the samples have spread far enough is to use a lower threshold for the effective sample size. Later $N_{th} = 2N/3$ will be used as the threshold.

When the samples move away from the real state their weights decrease, called sample degeneracy.²¹ This in turn will decrease the effective sample size, which eventually will pass the threshold. When this happens we draw N new samples from $\{\mathbf{X}_k^{(i)}\}$ with replacement, where the probability of choosing $\mathbf{X}_k^{(i)}$ is $\bar{w}_k^{(i)}$. The new set of samples are drawn from the estimate of the posterior and all the weights should therefore be set to $1/N$. By doing this, the samples with high weights will be multi-

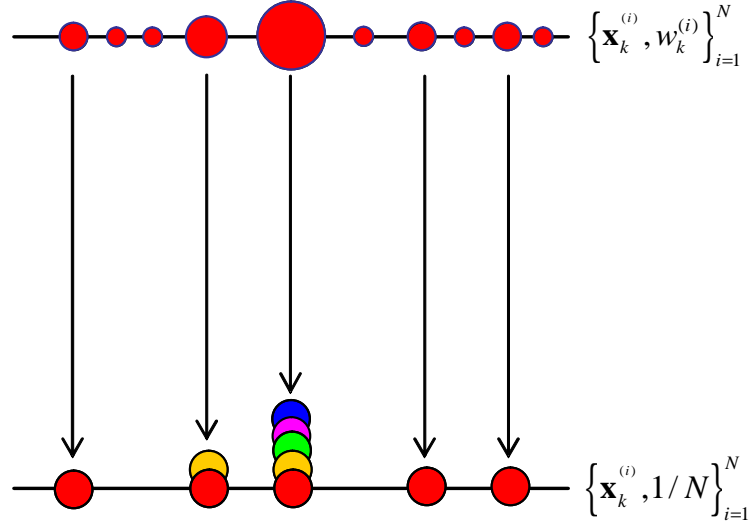


Fig. 4.2 Diagram for Particle Resampling

plied, which is described in Fig. 4.2. In this way the cloud of samples is concentrated in the interesting region.

There is a price to pay for this solution. In other words, the samples are no longer independent because some of them will be copies of the same sample. As long as we do not resample too often, this problem turns out to be of less importance. In this thesis a minimum number of time steps between *resampling* is used.

4.3.2.2 Resampling Algorithms

One of the methods that can mitigate the effects of the sample degeneracy is to use resampling whenever a significant degeneracy is observed.²¹ The basic idea of resampling is to eliminate particles with small weights and to concentrate on particle with large weights. In this section, three resampling algorithms are described briefly.

Simple Random Resampling A direct approach to implementing the resampling would consist of generating N independent and identically distributed (i.i.d.) random variables from the uniform distribution, sorting them in ascending order and

comparing them with the cumulative sum of normalized weights.^{17,67} First, calculate the thresholds by doing a cumulative sum of the normalized weights, in any order.

Then, for each index i

1. Draw a uniform random number u_i between 0 and 1, $u_i = \mathcal{U}[0, 1]$
2. Use a search algorithm (binary search) to locate the position of u_i within the thresholds
3. Set the resampled index according to the index of the location of u_i

Note that the idea of the random resampling algorithm is simple, but its implementation is computationally inefficient. Below alternative algorithms that are computationally inexpensive are introduced.

Residual Resampling This is a method described by Liu and Chen.⁶⁷ The procedure is

1. Retain $s_i = [N\bar{w}_k^{(i)}]$ copies of $\mathbf{x}_k^{(i)}$, $i = 1, \dots, n$, where $\bar{w}_k^{(i)}$ is the renormalized weight of $w_k^{(i)}$. Let $N_r = N - \sum_{i=1}^n s_i$
2. Obtain N_r i.i.d draws from $\mathbf{x}_{1:k}^{(i)}$ with probabilities proportional to $N\bar{w}_k^{(i)} - s_i$ $i = 1, \dots, N$
3. Let $\bar{w}^{(i)} = 1/N$ for i, \dots, N , i.e., reset the weights to 1

The residual resampling is preferred over the random resampling in that it saves computational load and provides reduced variation in the estimates.^{67,68}

Systematic Resampling Another efficient resampling scheme that use a minimum variance method is described,⁶⁹ in which the uniform distribution $\mathcal{U}[a, b]$ is utilized on the interval $[a, b]$. This method proceeds in the same way as the direct resampling algorithm where each value u_i is independently drawn from $\mathcal{U}[0, 1]$. However, u_i in the systematic resampling algorithm is drawn by using the following

Table 4.1 Systematic Resampling (SR) Algorithm

1. Initialization at time $i = 1$
 - Set $c_1 = 0$
2. For $i = 2, \dots, N$
 - Construct $c_i = c_{i-1} + w_k^i$
3. Set $i = 1$
4. Draw a starting point
 - $u_1 \sim \mathcal{U}[0, 1/N]$
5. For $j = 1, \dots, N$
 - Construct $u_j = u_1 + (j - 1)/N$
 - While $u_j > c_i$
 - * $i = i + 1$
 - Otherwise
 - * assign sample: $\mathbf{x}_k^j = \mathbf{x}_k^i$
 - * assign weight: $w_k^j = 1/N$

scheme

$$u_1 \sim \mathcal{U}[0, 1/N] \quad (4.30)$$

$$u_i = u_1 + \frac{i}{N} \quad (4.31)$$

The systematic resampling algorithm is described in Table 4.1.

4.3.3 Generic Particle Filter Algorithm

The theory described in this chapter is only to be considered as a motivation of how particle filters work. Algorithm illustrated in Table 4.2 is a summary of the particle filter or the SIS with the resampling stage. In implementing this algorithm, the choice of the proposal or importance distribution is the most critical design issue. It starts by sampling samples from the prior distribution and calculates the first set of weights from the first measurement. In each iteration the samples are drawn according to a selected importance distribution. Then, the weights are updated by using the selected proposal distribution and the drawn samples. The overall procedures for the generic particle filtering algorithm is depicted in Fig. 4.3.

4.4 SIR Particle Filtering

As a special case of the SIS algorithm, the sampling important sampling (SIR) approach proposed by Gordon¹⁵ is illustrated in this section. The SIR filter is an MC method that can be applied to recursive Bayesian filtering problems. The SIR algorithm is rather straightforward and can be derived easily from the SIS algorithm by an appropriate choice of the importance density and the resampling step. The optimal proposal distribution which minimizes the variance on the importance weights is give by^{17,67}

$$q(\mathbf{x}_k | \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k}) = p(\mathbf{x}_k | \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k}) \quad (4.32)$$

However, sampling from this proposal distribution is impractical for arbitrary densities. Thus, a simple and efficient choice of the importance density is the transition prior density¹⁷

$$q(\mathbf{x}_k | \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}) \quad (4.33)$$

Table 4.2 Generic Particle Filter (PF) Algorithm

- Initialization: at time $k = 0$
 1. For $i = 1, \dots, N$,
 - Sampling from the prior $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$
 2. For $i = 1, \dots, N$,
 - Calculate $w_0^{(i)} = p(\mathbf{y}_0 | \mathbf{x}_0^{(i)})$
 - Calculate the total weight $w_T = \sum_i^N w_0^{(i)}$
 - Normalize $w_0^{(i)} = w_T^{-1} w_0^{(i)}$
- Prediction and Update: For each time $k \geq 1$
 1. For $i = 1, \dots, N$,
 - Sample $\mathbf{x}_k^{(i)} \sim q(\mathbf{x}_k | \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k)$
 - Calculate the importance weights $w_k^{(i)} = w_{k-1}^{(i)} \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} | \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k)}$
 2. Calculate the total weight $w_T = \sum_i^N w_k^{(i)}$
 3. For $i = 1, \dots, N$,
 - Normalize $w_k^{(i)} = w_T^{-1} w_k^{(i)}$
 4. If $(N_{\text{eff}} < N_{th})$, then Choose either (a) or (b)
 - (a) Apply resampling algorithm
 - * $[\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N] = \text{Resample}(\text{RR}, \text{SR})[\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N]$
 - (b) Apply resampling algorithm
 - * Set the weights, $w_k^{(i)} = 1/N$

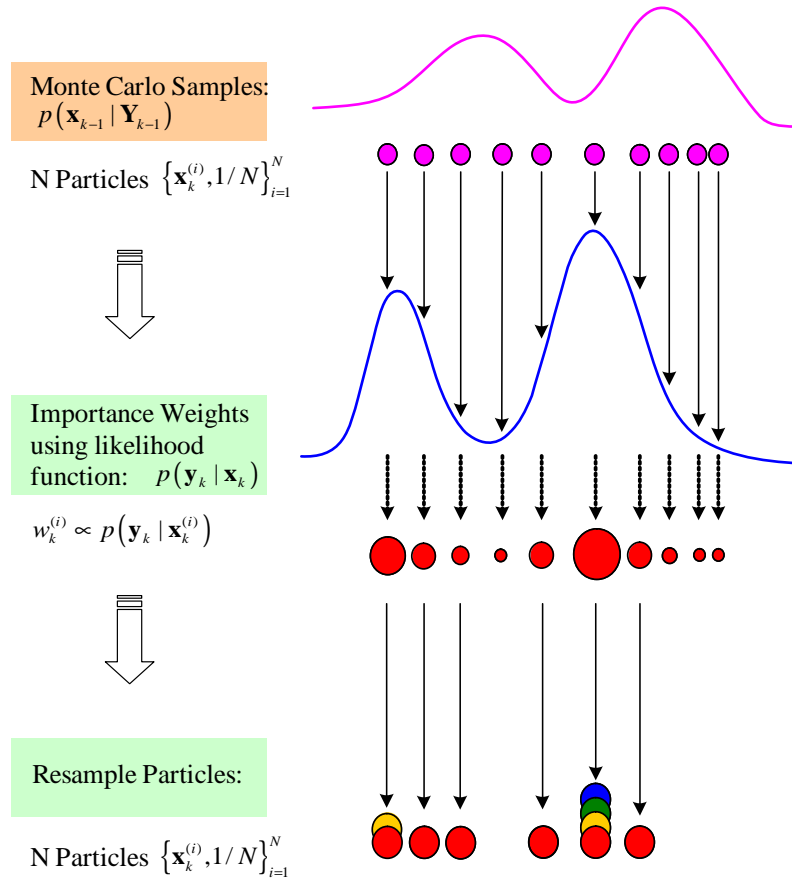


Fig. 4.3 Systematic Diagram for Generic Particle Filtering

This proposal density has samples drawn in the form

$$\mathbf{x}_k^{(i)} \sim p\left(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}\right) \quad (4.34)$$

The practical procedures for generating a sample $\mathbf{x}_k^{(i)}$ is achieved by

1. First generating a process noise sample $\mathbf{v}_{k-1}^{(i)} \sim p_v(\mathbf{v}_{k-1})$ where p_v is the pdf of the noise \mathbf{v}_{k-1}

2. Then, substituting the samples $\mathbf{x}_{k-1}^{(i)}$ and $\mathbf{v}_{k-1}^{(i)}$ into the dynamic system function, i.e., $\mathbf{x}_k^{(i)} = \mathbf{f}\left(\mathbf{x}_{k-1}^{(i)}, \mathbf{v}_{k-1}^{(i)}\right)$

The generic update equation for the weights is given by

$$w_k \propto w_{k-1} p(\mathbf{y}_k | \mathbf{x}_k) \frac{p(\mathbf{x}_k | \mathbf{x}_{k-1})}{q(\mathbf{x}_k | \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k})} \quad (4.35)$$

For this particular choice of the importance density in Eq. (4.33), the corresponding weight update equation becomes

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) \quad (4.36)$$

It is noted that since resampling in the SIR algorithm is applied at every time step, the prior weights are all equal to $w_{k-1}^{(i)} = 1/N$. Thus, in this case the update weights becomes

$$w_k^{(i)} \propto p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) \quad (4.37)$$

The SIR particle filtering algorithm is illustrated in Table 4.3.

4.5 Improving Particle Filters

Many variant algorithms have been proposed by scientists and engineers to compensate for the drawbacks of the particle degeneracy and sample impoverishment, and improve the generic particle filtering algorithm. In a broad sense, the methods can

Table 4.3 SIR Particle Filter Algorithm

- Initialization: at time $k = 0$
 1. For $i = 1, \dots, N$,
 - Sampling $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$
 2. For $i = 1, \dots, N$,
 - Calculate $w_0^{(i)} = p(\mathbf{y}_0 | \mathbf{x}_0^{(i)})$
 - Calculate the total weight $w_T = \sum_i^N w_0^{(i)}$
 - Normalize $w_0^{(i)} = w_T^{-1} w_0^{(i)}$
- Prediction and Update: For each time $k \geq 1$
 1. For $i = 1, \dots, N$,
 - Sample $\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)})$
 - Calculate $w_k^{(i)} = p(\mathbf{y}_k | \mathbf{x}_k^{(i)})$
 2. Calculate the total weight $w_T = \sum_i^N w_k^{(i)}$
 3. For $i = 1, \dots, N$,
 - Normalize $w_k^{(i)} = w_T^{-1} w_k^{(i)}$
 4. Apply resampling algorithm
 - $[\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N] = \text{Resample}(\text{RR}, \text{SR})[\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N]$

be divided into the following four categories:

(1) Choice of Proposal Distribution: A first method for choosing an optimal importance density involves in maximizing the effective sample size \hat{N}_{eff} . In doing this, the optimal density function minimizes the variance of the weights $\bar{w}_k^{(i)}$.²² However, the calculation of the optimal important density requires to evaluate an multi-dimensional integral, which is not be tractable in practical applications.

(2) Local Linearization: An optimal importance density can be approximated by incorporating the most current measurement through a bank of the standard nonlinear filters.^{22,65} The approximated density propagates the particles towards the likelihood function and consequently the hybrid particle filter performs better than the SIR filter.

(3) Regularization: The resampling reduces the effects of the degeneracy phenomena, but it causes other practical problem called *sample impoverishment*.^{70,71} It comes from the fact the sampled particles with high weights are selected many times and thus contain many repeated points leading to a loss of diversity among the the particles. A modified particle filtering algorithm in which the resampling process is performed upon a kernel-based density estimation can be a potential solution to handle the sample impoverish effect.⁷²

(4) MCMC Move: Markov chain Monte Carlo (MCMC) methods provide a relatively easy way of generating samples from any probability distribution. It can also be a potential solution to the sample impoverishment in resampling step as well as the regularization scheme.^{72,73}

(5) Rao-Blackwellization: Some components of the model may have linear dynamics and can be well estimated using a conventional Kalman filter. The Kalman filter is combined with a particle filter to reduce the number of particles needed to obtain a given level of performance.^{74,75} This method can reduce the variance of the

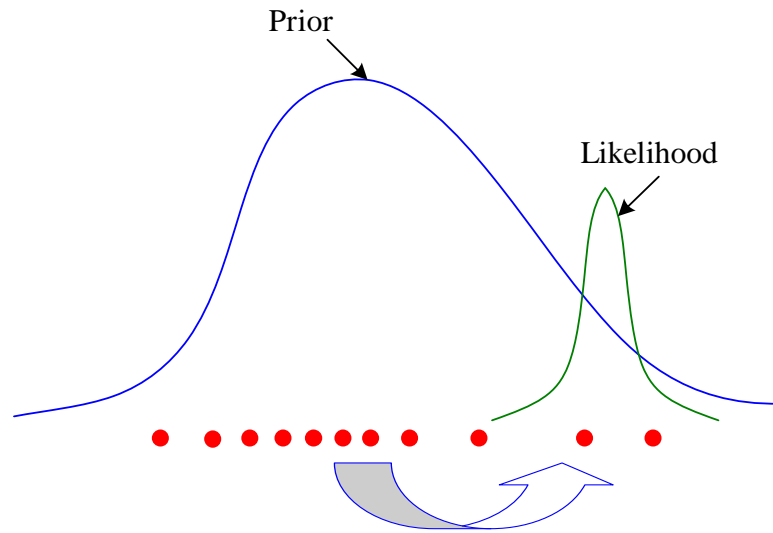


Fig. 4.4 Concept for Moving Samples to Regions of High Likelihood

MC estimates as well as the number of samples.

Specific description about each method is made in the following sections.

4.6 Local Linearization Particle Filter

The sample degeneracy of the SIS algorithm results from the fact that the variance of the importance weights increases over time. An optimal importance density can be approximated by incorporating the most current measurement through a bank of the standard nonlinear filters such as the extended Kalman filter, unscented Kalman filters, or the divided difference filter^{22,65} This process is important if the likelihood lies in one of the tails of the prior distribution or it is too much narrow due to small measurement error. Fig. 4.4 describes the overall concept of how to include the most current observation into the proposal distribution and move the samples to the regions of high likelihood. The basic idea is to use a separate nonlinear filter, $EKF(i)$, $UKF(i)$, or $DDF(i)$, for each particle index i in order to generate and propagate a

Gaussian importance distribution

$$q\left(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)},\mathbf{y}_k\right)=\mathcal{N}\left(\hat{\mathbf{x}}_k^{(i)},\hat{\mathbf{P}}_k^{(i)}\right) \quad (4.38)$$

where $\hat{\mathbf{x}}_k^{(i)}$ and $\hat{\mathbf{P}}_k^{(i)}$ are estimates of the mean and covariance computed from EKF(i), UKF(i) or DDF(i) at time k using measurement \mathbf{y}_k . The proposed particle filter is referred to as the *local linearization particle filter* (LLPF) in a unified way, and also called the *extended Kalman particle filter* (EKPF),²² *unscented particle filter* (UPF),⁶⁵ and *divided difference particle filter* (DDPF), individually. A single cycle of this filter is given by the algorithm in Table 4.4. This routine performs resampling at every time step, and therefore the importance weights are not passed on from one cycle to the next.

The local linearization method for approximation of the importance density propagates the particles towards the likelihood function, and consequently the LLPF performs better than the SIR filter. The additional computational cost of using such an importance density is often more than offset by reduction in the number of samples required to achieve a certain level of performance. Using UKF instead of the EKF in the local linearization particle filter is reported to improve the performance.⁶⁵ In this dissertation, in addition, the divided difference filter based particle filter is derived, which will be named as divided difference particle filter (DDPF). The UKPF and DDPF can be called sigma point particle filters (SPPFs) in a unified way.

4.6.1 Extended Kalman Particle Filter

The optimal importance density can be approximated by incorporating the most current measurement \mathbf{y}_k through a bank of the standard nonlinear filters such as the extended Kalman filter.²² This is a popular method for devising proposal distributions that approximate the optimal importance distribution. It relies on the

Table 4.4 Local Linearization Particle Filter (LLPF)

Initialization: At time $k = 0$

1. For $i = 1, \dots, N$, sample $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$
2. For each $i = 1, \dots, N$, calculate the weights $w_0^{(i)} = p(\mathbf{y}_0 | \mathbf{x}_0^{(i)})$
and normalize $\bar{w}_0^{(i)} = \frac{w_0^{(i)}}{\sum_j w_0^{(j)}}$

Prediction and Update: For each time $k \geq 1$

1. For each $i = 1, \dots, N$, run $\{EKF/UKF/DDF\}$
 $[\hat{\mathbf{x}}_{k-1}^{(i)}, \hat{\mathbf{P}}_{k-1}^{(i)}] = EKF/UKF/DDF(\mathbf{x}_{k-1}^{(i)}, \mathbf{P}_{k-1}^{(i)})$
2. For $i = 1, \dots, N$, draw a sample from importance density
 $\mathbf{x}_k^{(i)} \sim \mathcal{N}(\mathbf{x}_k^{(i)}; \hat{\mathbf{x}}_k^{(i)}, \hat{\mathbf{P}}_k^{(i)})$
3. For each $i = 1, \dots, N$, calculate $w_k^{(i)} = p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) \bar{w}_{k-1}^{(i)}$
and normalize the importance weights $\bar{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_j w_k^{(j)}}$
4. If resampling ($N_{\text{eff}} < N_{th}$)
then, set the weights $\bar{w}_{k-1}^{(i)} = \frac{1}{N}$ and resample with $\{\mathbf{x}_k^{(i)}, \bar{w}_k^{(i)}\}$

Output: A set of samples is used to approximate the posterior distribution

1. $\hat{p}(\mathbf{x}_k | \mathbf{Y}_k) = \sum_{i=1}^N \bar{w}_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)})$
2. $\hat{\mathbf{x}}_k = \sum_{i=1}^N \bar{w}_k^{(i)} \mathbf{x}_k^{(i)}$
3. $\hat{\mathbf{P}}_k = \sum_{i=1}^N \bar{w}_k^{(i)} (\hat{\mathbf{x}}_k - \mathbf{x}_k^{(i)}) (\hat{\mathbf{x}}_k - \mathbf{x}_k^{(i)})^T$

first-order Taylor series expansions of the likelihood and transition distribution as well as a Gaussian assumption on all the random variables. In the framework, the EKF approximates the optimal minimum-mean square error estimator of the system state by calculating the conditional mean of the state given all of the observations. This is done in a recursive framework by propagating the Gaussian approximation of the posterior distribution through time and combining it at each time step with the new observation.

The proposed particle filter is referred to as the local linearization particle filter (LLPF) in a unified way, and is also called the *extended Kalman particle filter* (EKPF), respectively.¹⁷ The algorithm of the extended Kalman particle filter is summarized in Table. 4.5.

4.6.2 Unscented Particle Filter

The unscented Kalman filter (UKF) is able to more accurately propagate the mean and covariance of the Gaussian approximation of the state distribution than the EKF. In comparison to the EKF, the UKF tends to generate more accurate estimates of the true covariance of the state. Distributions generated by the UKF generally have a bigger support overlap with the true posterior distribution than the overlap achieved by the EKF estimates. This is related to the fact that the UKF calculates the posterior covariance accurately to the 3rd order, whereas the EKF relies on a first-order approximation. This makes the UKF a better candidate for a more accurate proposal distribution generation within the particle filter framework. The UKF also has the ability to scale the approximation errors in the higher order moments of the posterior distribution such as *kurtosis*, allowing for heavier tailed distributions. Because the sigma point set used in the UKF is deterministically designed to capture certain characteristics of the prior distribution, one can explicitly optimize the algorithm to

Table 4.5 Extended Kalman Particle Filter (EKPF) Algorithm

Initialization: At time $k = 0$

1. For $i = 1, \dots, N$, sample $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$
2. For each $i = 1, \dots, N$, calculate the weights $w_0^{(i)} = p(\mathbf{y}_0 | \mathbf{x}_0^{(i)})$
and normalize $\bar{w}_0^{(i)} = \frac{w_0^{(i)}}{\sum_j w_0^{(j)}}$

Prediction and Update: For each time $k \geq 1$

1. For each $i = 1, \dots, N$, update the particles with the EKF

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1}^{(i)} &= \mathbf{f}(\mathbf{x}_{k-1}^{(i)}) \\ \mathbf{P}_{k|k-1}^{(i)} &= \mathbf{F}_k^{(i)} \mathbf{P}_{k-1}^{(i)} \left(\mathbf{F}_k^{(i)}\right)^T + \mathbf{Q}_k \\ \mathcal{K}_k^{(i)} &= \mathbf{P}_{k|k-1}^{(i)} \left(\mathbf{H}_k^{(i)}\right)^T \left[\mathbf{H}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)} \left(\mathbf{H}_k^{(i)}\right)^T + \mathbf{R}_k\right]^{-1} \\ \hat{\mathbf{x}}_k^{(i)} &= \hat{\mathbf{x}}_{k|k-1}^{(i)} + \mathcal{K}_k^{(i)} \left(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}^{(i)})\right) \\ \hat{\mathbf{P}}_k^{(i)} &= \mathbf{P}_{k|k-1}^{(i)} - \mathcal{K}_k^{(i)} \mathbf{H}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)}\end{aligned}$$

2. For $i = 1, \dots, N$, draw a sample from importance density

$$\mathbf{x}_k^{(i)} \sim \mathcal{N}(\mathbf{x}_k^{(i)}; \hat{\mathbf{x}}_k^{(i)}, \hat{\mathbf{P}}_k^{(i)})$$

3. For each $i = 1, \dots, N$, calculate $w_k^{(i)} = p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) \bar{w}_{k-1}^{(i)}$
and normalize the importance weights $\bar{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_j w_k^{(j)}}$

4. If resampling ($N_{\text{eff}} < N_{th}$)

then, set the weights $\bar{w}_{k-1}^{(i)} = \frac{1}{N}$ and resample with $\{\mathbf{x}_k^{(i)}, \bar{w}_k^{(i)}\}$

Output: A set of samples that is used to approximate the posterior distribution

1. $\hat{p}(\mathbf{x}_k | \mathbf{Y}_k) = \sum_{i=1}^N \bar{w}_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)})$
2. $\hat{\mathbf{x}}_k = E(\mathbf{x}_k | \mathbf{Y}_k) \approx \sum_{i=1}^N \bar{w}_k^{(i)} \mathbf{x}_k^{(i)}$

work with distributions that have heavier tails than the Gaussian distribution, such as the Cauchy or Student-t distributions. This characteristic makes the UKF very attractive for the generation of proposal distributions.

The new filter that results from using the UKF for the generation of proposal distributions within a particle filter framework is called the *unscented particle filter* (UPF). The UPF algorithm is illustrated in Table 4.6.

4.6.3 Divided Difference Particle Filter

The DDF can be described as one of the sigma point Kalman filters (SPFs) in a unified way where the filter linearizes the nonlinear dynamic and measurement functions by using an interpolation formula through systematically chosen sigma points. The linearization is based on polynomial approximations of the nonlinear transformations that are obtained by Stirling's interpolation formula, rather than the derivative-based Taylor series approximation. Conceptually, the implementation principle resembles that of the EKF, the implementation, however, is significantly simpler because it is not necessary to formulate the Jacobian and/or Hessian matrices of partial derivatives of the nonlinear dynamic and measurement equations. Thus, the new nonlinear state filter, divided difference filter (DDF), can also replace the extended Kalman filter (EKF) and its higher-order estimators in practical real-time applications that require accurate estimation but less computational cost.

In this section, the DDF used for the generation of proposal distributions is integrated with a particle filtering framework, leading to a *divided difference particle filter* (DDPF). The DDPF algorithm is illustrated in Table 4.7.

Table 4.6 Unscented Particle Filter (UPF) Algorithm

Initialization: At time $k = 0$

1. For $i = 1, \dots, N$, sample $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$
2. For each $i = 1, \dots, N$, calculate the weights $w_0^{(i)} = p(\mathbf{y}_0 | \mathbf{x}_0^{(i)})$

$$\begin{aligned} \hat{\mathbf{x}}_0^{(i)} &= E\left\{\mathbf{x}_0^{(i)}\right\}, & \mathbf{P}_0^{(i)} &= E\left\{(\mathbf{x}_0^{(i)} - \hat{\mathbf{x}}_0^{(i)})(\mathbf{x}_0^{(i)} - \hat{\mathbf{x}}_0^{(i)})^T\right\} \\ \mathbf{x}_0^{a,(i)} &= \begin{bmatrix} \mathbf{x}_0^{(i)} \\ \mathbf{w}_0 \end{bmatrix}, & \mathbf{P}_0^{a,(i)} &= \begin{bmatrix} \mathbf{P}_0^{(i)} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_0 \end{bmatrix} \end{aligned}$$

Prediction and Update: For each time $k \geq 1$

1. For each $i = 1, \dots, N$, update the particles with the UKF

- Draw sigma points:

$$\mathcal{X}_k^{a,(i)} = \left[\hat{\mathbf{x}}_k^{a,(i)} \hat{\mathbf{x}}_k^{a,(i)} + \sqrt{(n+q+\lambda) \mathbf{P}_k^{a,(i)}} \hat{\mathbf{x}}_k^{a,(i)} - \sqrt{(n+q+\lambda) \mathbf{P}_k^{a,(i)}} \right]$$

- Prediction and update:

$$\mathcal{X}_{k+1}^{x,(i)} = \mathbf{f}\left(\mathcal{X}_k^{a,(i)}, k\right), \quad \hat{\mathbf{x}}_{k+1|k}^{(i)} = \sum_{j=0}^{2(n+q)} W_j^{(m)} \mathcal{X}_{j,k+1}^{x,(i)}$$

$$\mathbf{P}_{k+1|k}^{(i)} = \sum_{j=0}^{2(n+q)} W_j^{(c)} \left[\mathcal{X}_{j,k+1}^{x,(i)} - \hat{\mathbf{x}}_{k+1|k}^{(i)} \right] \left[\mathcal{X}_{j,k+1}^{x,(i)} - \hat{\mathbf{x}}_{k+1|k}^{(i)} \right]^T$$

$$\mathcal{Y}_{k+1}^{(i)} = \mathbf{h}\left(\mathcal{X}_{k+1}^{x,(i)}, k+1\right), \quad \hat{\mathbf{y}}_{k+1|k}^{(i)} = \sum_{j=0}^{2(n+q)} W_j^{(m)} \mathcal{Y}_{j,k+1}^{(i)}$$

$$\mathbf{P}_{k+1}^{vv} = \sum_{j=0}^{2(n+q)} W_j^{(c)} \left[\mathcal{Y}_{j,k+1}^{(i)} - \hat{\mathbf{y}}_{k+1|k}^{(i)} \right] \left[\mathcal{Y}_{j,k+1}^{(i)} - \hat{\mathbf{y}}_{k+1|k}^{(i)} \right]^T$$

$$\mathbf{P}_{k+1}^{xy} = \sum_{j=0}^{2(n+q)} W_j^{(c)} \left[\mathcal{X}_{j,k+1}^{x,(i)} - \hat{\mathbf{x}}_{k+1|k}^{(i)} \right] \left[\mathcal{Y}_{j,k+1}^{(i)} - \hat{\mathbf{y}}_{k+1|k}^{(i)} \right]^T$$

$$\mathcal{K}_{k+1} = \mathbf{P}_{k+1}^{xy} (\mathbf{P}_{k+1}^{vv})^{-1}$$

$$\hat{\mathbf{x}}_{k+1}^{(i)} = \hat{\mathbf{x}}_{k+1|k}^{(i)} + \mathcal{K}_{k+1} \left(\mathbf{y}_k - \mathbf{h}\left(\hat{\mathbf{x}}_{k+1|k}^{(i)}\right) \right)$$

$$\hat{\mathbf{P}}_{k+1}^{(i)} = \mathbf{P}_{k+1|k}^{(i)} - \mathcal{K}_{k+1} \mathbf{P}_{k+1}^{vv} (\mathcal{K}_{k+1})^T$$

2. For $i = 1, \dots, N$, draw samples $\mathbf{x}_k^{(i)} \sim \mathcal{N}\left(\mathbf{x}_k^{(i)}; \hat{\mathbf{x}}_k^{(i)}, \hat{\mathbf{P}}_k^{(i)}\right)$

3. For each $i = 1, \dots, N$, calculate $w_k^{(i)} = p\left(\mathbf{y}_k | \mathbf{x}_k^{(i)}\right) \bar{w}_{k-1}^{(i)}$ and $\bar{w}_k^{(i)} = w_k^{(i)} / \sum_j w_k^{(j)}$

4. If resampling ($N_{\text{eff}} < N_{th}$), set $\bar{w}_{k-1}^{(i)} = \frac{1}{N}$ and resample with $\left\{\mathbf{x}_k^{(i)}, \bar{w}_k^{(i)}\right\}$

Output: $\hat{p}(\mathbf{x}_k | \mathbf{Y}_k) = \sum_{i=1}^N \bar{w}_k^{(i)} \delta\left(\mathbf{x}_k - \mathbf{x}_k^{(i)}\right)$, $\hat{\mathbf{x}}_k = E(\mathbf{x}_k | \mathbf{Y}_k) \approx \sum_{i=1}^N \bar{w}_k^{(i)} \mathbf{x}_k^{(i)}$

Table 4.7 Divided Difference Particle Filter (DDPF) Algorithm

Initialization: At time $k = 0$

1. For $i = 1, \dots, N$, sample $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$
2. A priori information and factorization

$$\begin{aligned}\hat{\mathbf{x}}_0^{(i)} &= E\left\{\mathbf{x}_0^{(i)}\right\}, \quad \mathbf{P}_0^{(i)} = E\left\{(\mathbf{x}_0^{(i)} - \hat{\mathbf{x}}_0^{(i)})(\mathbf{x}_0^{(i)} - \hat{\mathbf{x}}_0^{(i)})^T\right\} \\ \mathbf{P}_0^{(i)} &= \mathbf{S}_x^{(i)} \left(\mathbf{S}_x^{(i)}\right)^T, \quad \mathbf{Q}_k = \mathbf{S}_w \mathbf{S}_w^T, \quad \mathbf{R}_k = \mathbf{S}_v \mathbf{S}_v^T\end{aligned}$$

Prediction and Update: For each time $k \geq 1$

1. For each $i = 1, \dots, N$, update the particles with the DDF

• Prediction:

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k}^{(i)} &= \mathbf{f}\left(\hat{\mathbf{x}}_k^{(i)}, \bar{\mathbf{w}}_k, k\right), \quad \hat{\mathbf{y}}_{k+1|k}^{(i)} = \mathbf{h}\left(\hat{\mathbf{x}}_{k+1|k}^{(i)}, \bar{\mathbf{v}}_{k+1}, k+1\right) \\ \mathbf{S}_{x\hat{x}}^{(i)}(k+1) &= \frac{1}{2h} \left\{ \mathbf{f}\left(\hat{\mathbf{x}}_k^{(i)} + h\mathbf{s}_x^{(i)}, \bar{\mathbf{w}}_k\right) - \mathbf{f}\left(\hat{\mathbf{x}}_k^{(i)} - h\mathbf{s}_x^{(i)}, \bar{\mathbf{w}}_k\right) \right\} \\ \mathbf{S}_{xw}^{(i)}(k+1) &= \frac{1}{2h} \left\{ \mathbf{f}\left(\hat{\mathbf{x}}_k^{(i)}, \bar{\mathbf{w}}_k + h\mathbf{s}_{w,j}\right) - \mathbf{f}\left(\hat{\mathbf{x}}_k^{(i)}, \bar{\mathbf{w}}_k - h\mathbf{s}_{w,j}\right) \right\} \\ \mathbf{P}_{k+1|k}^{(i)} &= \begin{bmatrix} \mathbf{S}_{x\hat{x}}^{(i)}(k+1) & \mathbf{S}_{xw}^{(i)}(k+1) \end{bmatrix} \begin{bmatrix} \mathbf{S}_{x\hat{x}}^{(i)}(k+1) & \mathbf{S}_{xw}^{(i)}(k+1) \end{bmatrix}^T\end{aligned}$$

• Factorization: $\mathbf{P}_{k+1|k}^{(i)} = \mathbf{S}_x^{-,(i)} \left(\mathbf{S}_x^{-,(i)}\right)^T$

• Update:

$$\begin{aligned}\mathbf{S}_{y\hat{x}}^{(i)}(k+1) &= \frac{1}{2h} \left\{ \mathbf{h}\left(\hat{\mathbf{x}}_{k+1|k}^{(i)} + h\mathbf{s}_x^{-,(i)}, \bar{\mathbf{v}}_{k+1}\right) - \mathbf{h}\left(\hat{\mathbf{x}}_{k+1|k}^{(i)} - h\mathbf{s}_x^{-,(i)}, \bar{\mathbf{v}}_{k+1}\right) \right\} \\ \mathbf{S}_{yv}^{(i)}(k+1) &= \frac{1}{2h} \left\{ \mathbf{h}\left(\hat{\mathbf{x}}_{k+1|k}^{(i)}, \bar{\mathbf{v}}_{k+1} + h\mathbf{s}_v\right) - \mathbf{h}\left(\hat{\mathbf{x}}_{k+1|k}^{(i)}, \bar{\mathbf{v}}_{k+1} - h\mathbf{s}_v\right) \right\} \\ \mathbf{P}_{k+1}^{vv} &= \begin{bmatrix} \mathbf{S}_{y\hat{x}}^{(i)}(k+1) & \mathbf{S}_{yv}^{(i)}(k+1) \end{bmatrix} \begin{bmatrix} \mathbf{S}_{y\hat{x}}^{(i)}(k+1) & \mathbf{S}_{yv}^{(i)}(k+1) \end{bmatrix}^T \\ \mathbf{P}_{k+1}^{xy} &= \begin{bmatrix} \mathbf{S}_{x\hat{x}}^{(i)}(k+1) & \mathbf{S}_{xw}^{(i)}(k+1) \end{bmatrix} \left(\mathbf{S}_{y\hat{x}}^{(i)}(k+1)\right)^T\end{aligned}$$

$$\mathcal{K}_{k+1} = \mathbf{P}_{k+1}^{xy} (\mathbf{P}_{k+1}^{vv})^{-1}$$

$$\hat{\mathbf{x}}_{k+1}^{(i)} = \hat{\mathbf{x}}_{k+1|k}^{(i)} + \mathcal{K}_{k+1} \left(\mathbf{y}_k - \mathbf{h}\left(\hat{\mathbf{x}}_{k+1|k}^{(i)}\right)\right)$$

$$\hat{\mathbf{P}}_{k+1}^{(i)} = \mathbf{P}_{k+1|k}^{(i)} - \mathcal{K}_{k+1} \mathbf{P}_{k+1}^{vv} (\mathcal{K}_{k+1})^T$$

2. For $i = 1, \dots, N$, draw samples $\mathbf{x}_k^{(i)} \sim \mathcal{N}\left(\mathbf{x}_k^{(i)}; \hat{\mathbf{x}}_k^{(i)}, \hat{\mathbf{P}}_k^{(i)}\right)$
3. For each $i = 1, \dots, N$, calculate $w_k^{(i)} = p\left(\mathbf{y}_k | \mathbf{x}_k^{(i)}\right) \bar{w}_{k-1}^{(i)}$ and $\bar{w}_k^{(i)} = w_k^{(i)} / \sum_j w_k^{(j)}$
4. If resampling ($N_{\text{eff}} < N_{th}$), set $\bar{w}_{k-1}^{(i)} = \frac{1}{N}$ and resample with $\left\{\mathbf{x}_k^{(i)}, \bar{w}_k^{(i)}\right\}$

Output: $\hat{p}(\mathbf{x}_k | \mathbf{Y}_k) = \sum_{i=1}^N \bar{w}_k^{(i)} \delta\left(\mathbf{x}_k - \mathbf{x}_k^{(i)}\right)$, $\hat{\mathbf{x}}_k = E(\mathbf{x}_k | \mathbf{Y}_k) \approx \sum_{i=1}^N \bar{w}_k^{(i)} \mathbf{x}_k^{(i)}$

4.7 Regularized Particle Filter

Note that the resampling methods explained in the previous section play a role of reducing the degeneracy problem of the samples, which is a common problem of the SIS particle filter. As a suitable means for measuring the sample degeneracy of the algorithm the effective sample size was introduced.⁶⁷ However, the resampling reduces the effects of the degeneracy phenomena, it causes other practical problem called *sample impoverishment*. It comes from the fact the sampled particles with high weights are selected many times, and thus contain many repeated points leading to a loss of diversity among the particles.⁷⁰ Modified particle filtering algorithms^{70,72} have been suggested to handle the sample impoverish effect. In this section, the regularized particle filter (RPF)⁷⁰ as a potential solution to the sample impoverishment is illustrated. The RPF is nothing but a modified SIR particle filter in which the resampling process is performed upon a density estimation. Before going into details of the RPF algorithms, the problem of density estimation is briefly reviewed in the following.

4.7.1 Density Estimation

It has been shown that the resampling step is performed by generating a new set of samples from an approximate discrete representation of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (4.39)$$

However, it is useful to perform the resampling from a functional distribution that is obtained by estimating the underlying density function. In a broad sense density estimation methods fall into the following two categories: parametric estimation and non-parametric estimation methods. The parametric estimation approach assumes

that the underlying density comes from a specific family of densities, the normal distributions, to estimate the parameters of the mean and variance of the normal distribution. In contrast, the non-parametric density estimation methods are preferred for the problems where the distribution family may not be known *a priori*, or may not be reconstructed by parametric forms.⁷⁶

The simplest method for obtaining a density estimate is to utilize a histogram where samples are drawn over equally spaced bins and bars have a height equal to the number of samples within the underlying bin. The histogram is limited in the sense that it provides a piecewise constant estimate to a smooth density and depends heavily on the number and center of the bins chosen.⁷⁶

The general non-parametric density estimator for a scalar case is represented in a form

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^N w(x, x^i) \quad (4.40)$$

where $w(x, x^i)$ is a weight function or *kernel* and x^i is the sample support value. The weight function produces a probability associated with all x in the neighborhood of the support value x^i . The simplest weight function is a rectangular box

$$\hat{p}(x) = \frac{1}{Nh} \sum_{i=1}^N w\left(\frac{x - x^i}{h}\right) \quad (4.41)$$

and

$$w(\mathcal{X}) = \begin{cases} 1/2 & \text{if } \|\mathcal{X}\| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.42)$$

where $2h$ is the rectangle width, which determines the amount of smoothing of the data and become a loose means of measure of bandwidth. Note that in all forms of kernel density estimation, there should be a trade-off between resolution and smoothness, which is adjusted by changing the size or variance of the kernel.

4.7.2 Kernel Methods

In this section a brief review of a scalar kernel method is illustrated. For a kernel estimation, it is desirable to choose a smooth weight function. If the weight function $w(x, x^i)$ is described in a closed functional form it is referred to as a *kernel*. A Gaussian density is usually chosen as candidate kernel $K(x)$ and has many advantageous computational properties. Then, the kernel density is a symmetric probability density function such that it satisfies⁷⁰

$$\int K(x) dx = 1, \quad \int xK(x) dx = 0, \quad \int \|x\|^2 K(x) dx < \infty \quad (4.43)$$

The Gaussian kernel density estimator is represented by

$$\hat{p}(x) = \frac{1}{N\sigma} \sum_{i=1}^N K_G\left(\frac{x - x^i}{\sigma}\right) \quad (4.44)$$

and

$$K_G = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\} \quad (4.45)$$

where σ is the variance of a underlying Gaussian density function.

A question that now arises is that what is the best kernel for estimating a density function. The best kernel is chosen to minimize the integrated mean-squared errors. In this choice, the optimal kernel is to use the *paraboloid* or *Epanechnikov* kernel⁷⁶

$$K_E = \begin{cases} \frac{3}{4\sqrt{5}} \left(1 - \frac{x^2}{5}\right) & \|x\| \leq \sqrt{5} \\ 0 & \text{otherwise} \end{cases} \quad (4.46)$$

The next question is how to decide a bandwidth σ that measures a level of smoothness in the kernel estimation. Different values of bandwidth can produce different density estimates. The common way of measuring performance is to choose a bandwidth that reflects the mean-squared roughness in the underlying density. For the Gaussian

kernel a good choice is

$$\sigma_{opt} \approx \frac{1}{N^{1/5}} \sqrt{\Sigma} \quad (4.47)$$

where Σ is the variance in the sample data.

The density estimation by using the kernel method requires convolution calculations, which makes the method difficult and demands a reasonable approximation technique. For the Gaussian kernel, computation can be minimized by using a fast Fourier transform (FFT) implementation for the convolution. Another potential solution to this problem is achieved by representing the data with a set of the data support value $\{x^i\}_{i=1}^N$ and by defining a new set of support values $\{\mathcal{X}^j\}_{j=1}^M$ for the underlying density estimate. Then, the kernel is evaluated for all pair-wise combinations of the data and estimate support values

$$\hat{p}(\mathcal{X}^j) = \frac{1}{N} \sum_{i=1}^N K(\mathcal{X}^j, x^i), \quad j = 1, \dots, M \quad (4.48)$$

4.7.3 Regularized Particle Filter

The RPF resamples from a continuous approximation of the probability density $p(\mathbf{x}_k | \mathbf{y}_{1:k})$, which is obtained by using the *kernel* density estimation method⁷⁰

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \approx \hat{p}(\mathbf{x}_k | \mathbf{y}_{1:k}) = \sum_{i=1}^{N_s} w_k^i K_h(\mathbf{x}_k - \mathbf{x}_k^i) \quad (4.49)$$

and

$$K_h(\mathbf{x}) = \frac{1}{h^{n_x}} K\left(\frac{\mathbf{x}}{h}\right) \quad (4.50)$$

where K_h is the rescaled kernel density, $h > 0$ is the kernel bandwidth, n_x is the dimension of the state vector \mathbf{x} , and w_k^i is a normalized weight. The multivariate Gaussian kernel is given by

$$K(\mathbf{x}) = \frac{1}{(2\pi)^{n_x/2}} \exp\left\{-\frac{1}{2}\mathbf{x}^T \mathbf{x}\right\} \quad (4.51)$$

The kernel density is a symmetric probability density function satisfying the following

$$\int \mathbf{x} K(\mathbf{x}) d\mathbf{x} = 0, \quad \int \|\mathbf{x}\|^2 K(\mathbf{x}) d\mathbf{x} < \infty \quad (4.52)$$

The multivariate kernel $K(\cdot)$ and bandwidth h are chosen to minimize the mean integrated square error (MISE) between the true posterior density and the corresponding estimated density, which is defined by²¹

$$\text{MISE} = E \left\{ \int [\hat{p}(\mathbf{x}_k | \mathbf{y}_{1:k}) - p(\mathbf{x}_k | \mathbf{y}_{1:k})]^2 d\mathbf{x}_k \right\} \quad (4.53)$$

For the special case where all the samples have the same weight, the optimal choice of the Kernel is the *Epanechnikov* kernel⁷⁰

$$K_E(\mathbf{x}) = \begin{cases} \frac{n_x+2}{C_{n_x}} (1 - \mathbf{x}^T \mathbf{x}), & \text{if } \mathbf{x}^T \mathbf{x} < 1 \\ 0, & \text{otherwise} \end{cases} \quad (4.54)$$

where C_{n_x} is the volume of the unit n_x -dimensional sphere. It is necessary to prewhiten the data by transforming it to have unit variance before smoothing with a symmetric Kernel, which is equal to utilizing a density estimate in the form

$$\hat{p}(\mathbf{x}) = \frac{1}{N h^{n_x} |\mathbf{S}|^{-1/2}} \sum_{i=1}^N \left(\frac{[\mathbf{x} - \mathbf{x}^i]^T \mathbf{S}^{-1} [\mathbf{x} - \mathbf{x}^i]}{h^2} \right) \quad (4.55)$$

When the underlying density is the multivariate Gaussian Kernel, the optimal choice for the bandwidth is

$$h_{opt} = \left[\frac{4}{N(2n_x + 1)} \right]^{1/(n_x+4)} \quad (4.56)$$

The optimal window size for the multivariate Epanechnikov Kernel is given by⁷⁶

$$h_{opt} = \left[\frac{8n_x(n_x + 2)(n_x + 4)(2\sqrt{\pi})^{n_x}}{N(2n_x + 1)c_{n_x}} \right]^{1/(n_x+4)} \quad (4.57)$$

Note that the RPF is different from the generic particle filter in that when performing the resampling step the regularization step is placed additionally. From

the practical point of view, the RPF performance is better than that of the SIR in cases where sample impoverishment is severe due to the process small noise, but it requires additional computational load. Details of the algorithm of the regularized particle filter (RPF) is summarized as follow in Table 4.8.

4.8 Markov Chain Monte Carlo Method

Markov chain Monte Carlo (MCMC) methods provide a relatively easy way of generating samples from any probability distribution. It can also be a potential solution to the sample impoverishment in resampling step as well as the regularization scheme explained previously.

For illustration, consider a collection of random variables $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k\}$, where \mathbf{x}_k may be regarded as the state of a system at time k . A Markov chain is a process with the property

$$p(\mathbf{x}_k | \mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}) \quad (4.58)$$

where $p(\cdot | \cdot)$ is the conditional transition Kernel. The property of the Markov chain says that the probability distribution of any state is dependent only on the previous state.

The stationary distribution of a Markov chain, $\pi(\mathbf{x})$, is defined by

$$\pi(\mathbf{x}) = \int_y \pi(\mathbf{x}) p(\mathbf{x} | \mathbf{y}) d\mathbf{y} \quad (4.59)$$

where $p(\mathbf{x} | \mathbf{y})$ is the transition probability. It may be hard to choose a Markov chain that satisfies the invariance condition in Eq. (4.59), thus a time reversibility condition is placed.²² The transition probability $p(\mathbf{y} | \mathbf{x})$ is said to be reversible with respect to

Table 4.8 Regularized Particle Filter (RPF) Algorithm

- Initialization:** At time $k = 0$
1. For $i = 1, \dots, N$
 - Draw samples $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$
 2. For $i = 1, \dots, N$
 - Calculate $w_0^{(i)} = p(\mathbf{y}_0 | \mathbf{x}_0^{(i)})$
 - Calculate total weight $w_{total} = \sum_{i=1}^N w_0^{(i)}$
 - Normalize the weight $\bar{w}_0^{(i)} = w_{total}^{-1} w_0^{(i)}$
- Prediction and Update:** For each time $k \geq 1$
1. Calculate the effective sample size N_{eff}
 2. If $(N_{\text{eff}} < N_{th})$
 - Calculate the empirical covariance \mathbf{S}_k of $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N$
 - Compute \mathbf{D}_k such that $\mathbf{D}_k \mathbf{D}_k^T = \mathbf{S}_k$
 - Apply resampling algorithm
 - * $\left[\{\mathbf{x}_k^i, w_k^i, -\}_{i=1}^N \right] = \text{RESAMPLE} \left[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N \right]$
 - For $i = 1, \dots, N$
 - * Draw $\epsilon^i \sim K$ from the Epanechnikov Kernel
 - * $\mathbf{x}_k^i = \mathbf{x}_k^i + h_{opt} \mathbf{D}_k \epsilon^i$
 3. Go back to $k = k + 1$

$\pi(\mathbf{x})$ if it satisfies the following relation

$$\pi(\mathbf{x})p(\mathbf{y}|\mathbf{x}) = \pi(\mathbf{x})p(\mathbf{x}|\mathbf{y}) \quad (4.60)$$

which is called detailed balance.

More detailed theoretical foundations about the MCMC algorithms are found in the references.^{77,78} In this section, implementation of the MCMC based on the Metropolis-Hastings algorithm⁶⁸ is illustrated by combining it with the importance sampling.

4.8.1 Metropolis-Hastings Algorithm

Most algorithms for Markov Chain Monte Carlo estimation are based on the algorithm of Hastings,⁷⁹ which is a generalization of the algorithm of Metropolis.⁸⁰ The Metropolis-Hastings algorithm resembles the previously described SIR algorithms in that a proposal distribution is used to generate the samples. However, the output of the algorithm is a Markov chain so the proposal density may depend on the current state of the chain.⁶⁸

An MH step of invariant target distribution $\pi(x)$ and proposal distribution $q(x^*|x)$ involves sampling a candidate value x^* given the current value x according to $q(x^*|x)$. The Markov chain then moves towards x^* with acceptance probability $\alpha(x, x^*)$, otherwise it remains at x . The MH algorithm is simple, but it is subject to the design of the proposal distribution $q(x^*|x)$.

Suppose the i th iteration of the Markov chain is denoted by a bracketed superscript $x^{(i)}$. The acceptance probability $\alpha(x^*, x^{(i-1)})$ is written by

$$\alpha(x^*, x^{(i-1)}) = \begin{cases} r(x^*|x^{(i-1)}) & \text{if } r(x^*|x^{(i-1)}) \leq 1 \\ 1 & \text{if } r(x^*|x^{(i-1)}) > 1 \end{cases} \quad (4.61)$$

Table 4.9 Metropolis-Hastings Algorithm

1. Initialize by setting $i = 0$ and draw a starting point $x^{(0)}$
2. For $i = 1$ to N
 - Draw samples from proposal density $x^* \sim q(x^*|x^{(i)})$
 - Draw a uniformly distributed random number u between 0 and 1, $u \sim \mathcal{U}(0, 1)$
 - Compute the acceptance probability $\alpha(x^*, x^{(i)})$
 - If $u < \alpha(x^*, x^{(i)})$
 - * accept the move by setting $x^{(i+1)} = x^*$
 - Else
 - * set $x^{(i+1)} = x^{(i)}$
3. $i = i + 1$ and return to item 2

where the ratio $r(x^*|x^{(i-1)})$ is

$$r(x^*|x^{(i-1)}) = \frac{\pi(x^*)}{\pi(x^{(i-1)})} \frac{q(x^{(i-1)}|x^*)}{q(x^*|x^{(i-1)})} \quad (4.62)$$

If the candidate is accepted the chain moves to the new position, while a rejection of the candidate leaves the chain at the current position in the state space. The MH pseudo algorithm is summarized in Table 4.9.

4.8.1.1 Proposal Choice

Even though the Metropolis-Hastings algorithm will be invariant for many choices of $q(x^*|x^{(i)})$ the choice of proposal distribution will determine the efficiency of the algorithm.⁶⁸ A simplistic way of choosing the proposal is to have it fixed, and independent of the current state of the chain. The *independence sampler*⁸¹ with a proposal distribution $q(x^*|x^{(i)}) = q(x^*)$ is a simple example of the MH algorithm, which yields

an acceptance probability given by

$$\alpha(x^*, x^{(i)}) = \min\left(1, \frac{z(x^*)}{z(x^{(i)})}\right) \text{ where } z(x) = \frac{\pi(x)}{q(x)} \quad (4.63)$$

From the *Metropolis algorithm* symmetric proposal distributions $q(x^{(i)}|x^*) = q(x^*|x^{(i)})$ are considered, then the acceptance probability reduces to⁸¹

$$\alpha(x^*, x^{(i)}) = \min\left(1, \frac{\pi(x^*)}{\pi(x^{(i)})}\right) \quad (4.64)$$

Note that the independent sampler and the Metropolis algorithm are two simple examples of the MH algorithm.

Some properties of the MH algorithm are summarized. First, the normalizing constant of the target distribution is not required. Second, although the pseudo MH algorithm makes use of a single chain, it is easy to simulate several independent chains in parallel. Finally, the success or failure of the algorithm often is dependent of the choice of the proposal distribution. Different choices of the proposal standard deviation σ lead to very different results.

4.8.2 Gibbs Sampler

The *Gibbs sampling algorithm*⁸² is the most commonly applied MCMC algorithm. The Gibbs sampling algorithm can be seen as an extended Metropolis-Hastings procedure where proposal samples are drawn directly from the full conditional distributions. Suppose a state vector \mathbf{x} is n -dimensional and the full conditional distribution is expressed by

$$p(x_j | x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n) \quad (4.65)$$

Each component is updated separately

$$q(x_j^{(i)}) = p(x_j | x_1^{(i-1)}, \dots, x_{j-1}^{(i-1)}, x_{j+1}^{(i-1)}, \dots, x_n^{(i-1)}) \quad (4.66)$$

Table 4.10 Gibbs Sampling Algorithm

1. Initialize by setting $i = 0$ and choose a starting point $\mathbf{x}^{(0)}$
2. For $i = 1$ to N
 - Draw samples from the full conditional distribution
 - * Sample $x_1^{(i)} \sim p\left(x_1|x_2^{(i-1)}, x_3^{(i-1)}, \dots, x_n^{(i-1)}\right)$
 - * Sample $x_2^{(i)} \sim p\left(x_2|x_1^{(i-1)}, x_3^{(i-1)}, \dots, x_n^{(i-1)}\right)$
 - * \vdots
 - * Sample $x_n^{(i)} \sim p\left(x_n|x_1^{(i-1)}, x_2^{(i-1)}, \dots, x_{n-1}^{(i-1)}\right)$
 - Output $x_j^{(i)}$
3. Increase $i = i + 1$ and return to item 2

where $x_j^{(i)}$ is the i th iterate in the Markov chain of the j th component of the state vector. Since the Gibbs sampler can be viewed as a special case of the MH algorithm, it is possible to introduce MH steps into the Gibbs sampler. For example, when the full conditionals are available and belong to the family of standard distributions the new samples directly drawn. Otherwise, samples are drawn with MH steps embedded within the Gibbs algorithm.⁸¹ It is obvious that the acceptance probability is always unity (substituting into Eq.(4.61)). Thus, all candidates are accepted and calculation of the acceptance probability is not required.

The algorithm is summarized in Table 4.10, where the i th iterate deterministic version of the Gibbs sampler is used.

4.9 Rao-Blackwellized Particle Filter

From the central limit theorem, it can be shown that the estimation error is independent of the dimensionality of the problem. However, it turns out that the dimension

has influence on the number of samples needed. Tests show that the same number of samples will yield a better accuracy for a problem of low dimension than for one of high dimension. The higher the dimensionality, the more samples needed to cover the state space efficiently is an intuitive explanation. In some problems, there is a structure in the state space formulation. This may be used to split the problem into one part that can be estimated in closed form, and leave the other part to simulation based methods. By using this structure, the accuracy increases while using the same number of samples. The next section provides a description of the Rao-Blackwellization algorithm in a way that is suitable for this dissertation. More theory can be found in Refs. 74 and 75.

Consider the discrete-time state-space models

$$\mathbf{x}_{k+1}^{pf} = \mathbf{f}^{pf}(\mathbf{x}_k^{pf}) + \mathbf{F}_k^{pf}(\mathbf{x}_k^{pf}) \mathbf{x}_k^{kf} + \mathbf{G}_k^{pf}(\mathbf{x}_k^{pf}) \mathbf{w}_k^{pf} \quad (4.67)$$

$$\mathbf{x}_{k+1}^{kf} = \mathbf{f}^{kf}(\mathbf{x}_k^{pf}) + \mathbf{F}_k^{kf}(\mathbf{x}_k^{pf}) \mathbf{x}_k^{kf} + \mathbf{G}_k^{kf}(\mathbf{x}_k^{pf}) \mathbf{w}_k^{kf} \quad (4.68)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k^{pf}) + \mathbf{H}_k(\mathbf{x}_k^{pf}) \mathbf{x}_k^{kf} + \mathbf{v}_k \quad (4.69)$$

where $\mathbf{x}_k = \begin{bmatrix} (\mathbf{x}_k^{pf})^T & (\mathbf{x}_k^{kf})^T \end{bmatrix}^T$. The superscript “ pf ” denotes that the state vector will be estimated by using the particle filter, and the state vector with the “ kf ” superscript can be estimated by using any filtering method such as the Kalman filter.

It is assumed that the process noise has Gaussian distribution

$$\mathbf{w}_k = \begin{bmatrix} \mathbf{w}_k^{pf} \\ \mathbf{w}_k^{kf} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k), \quad \mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_k^{pf} & \mathbf{M}_k \\ \mathbf{M}_k^T & \mathbf{Q}_k^{kf} \end{bmatrix}, \quad \mathbf{Q}_k^{pf} > 0 \quad (4.70)$$

and the measurement noise has zero mean and Gaussian distribution

$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \quad \mathbf{R}_k > 0 \quad (4.71)$$

The distribution of the initial state \mathbf{x}_0^{kf} is Gaussian

$$\mathbf{x}_0^{kf} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_0^{kf}), \quad \mathbf{P}_0^{kf} > 0 \quad (4.72)$$

The purpose of the *Rao-Blackwellized filter* is also to get the recursive estimation

$$p(\mathbf{x}_k | \mathbf{Y}_k) = p(\mathbf{x}_k^{pf}, \mathbf{x}_k^{kf} | \mathbf{Y}_k) \quad (4.73)$$

The direct approach is to apply the particle filter to the entire state vector as described in the previous section. However, for the class of system in Eq. (4.67) there exists a more efficient way. Consider the posterior density $p(\mathbf{X}_k^{pf}, \mathbf{x}_k^{kf} | \mathbf{Y}_k)$. Using Bayes's rule this density can be factorized into two parts

$$p(\mathbf{X}_k^{pf}, \mathbf{x}_k^{kf} | \mathbf{Y}_k) = p(\mathbf{x}_k^{kf} | \mathbf{X}_k^{pf}, \mathbf{Y}_k) p(\mathbf{X}_k^{pf} | \mathbf{Y}_k) \quad (4.74)$$

where $p(\mathbf{x}_k^{kf} | \mathbf{X}_k^{pf}, \mathbf{Y}_k) = \mathcal{N}(\hat{\mathbf{x}}_{k|k}^{kf}, \mathbf{P}_{k|k}^{kf})$, the recursive mean and covariance are estimated by the Kalman filter, and the particle filter is used to estimate $p(\mathbf{X}_k^{pf} | \mathbf{Y}_k)$.

For the state-space model in Eq. (4.67), there are two state transition equations where new samples \mathbf{x}_{k+1}^{pf} for the particle filter provide information for the state vector \mathbf{x}_{k+1}^{kf} . The Kalman filter equations are adjusted in order to take this information into consideration when estimating \mathbf{x}_{k+1}^{kf} . During the time update there is a second measurement update that comes from the new particle filter samples. The new measurement update is not the same as the first since the process noise is correlated between \mathbf{x}_{k+1}^{pf} and \mathbf{x}_{k+1}^{kf}

$$p(\mathbf{x}_k^{kf} | \mathbf{X}_k^{pf}, \mathbf{Y}_k) = \mathcal{N}(\hat{\mathbf{x}}_{k|k}^{kf}, \mathbf{P}_{k|k}^{kf}) \quad (4.75)$$

$$p(\mathbf{x}_k^{kf} | \mathbf{X}_k^{pf}, \mathbf{Y}_k) = \mathcal{N}(\hat{\mathbf{x}}_{k|k}^{kf}, \mathbf{P}_{k|k}^{kf}) \quad (4.76)$$

Update:

$$\hat{\mathbf{x}}_{k|k}^{kf} = \hat{\mathbf{x}}_{k|k-1}^{kf} + \mathcal{K}_k \left(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k^p) - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}^{kf} \right) \quad (4.77)$$

$$\mathbf{P}_k^{kf} = \mathbf{P}_{k|k-1}^{kf} - \mathcal{K}_k^{kf} \mathbf{S}_k^{kf} \left(\mathcal{K}_k^{kf} \right)^T \quad (4.78)$$

$$\mathcal{K}_k^{kf} = \mathbf{P}_{k|k-1}^{kf} \mathbf{H}_k^T \left(\mathbf{S}_k^{kf} \right)^{-1} \quad (4.79)$$

$$\mathbf{S}_k^{kf} = \mathbf{H}_k \mathbf{P}_{k|k-1}^{kf} \mathbf{H}_k^T + \mathbf{R}_k \quad (4.80)$$

Prediction:

$$\hat{\mathbf{x}}_{k+1|k}^{kf} = \left(\bar{\mathbf{F}}_k^{kf} - \mathcal{K}_k^{pf} \mathbf{F}_k^{pf} \right) \hat{\mathbf{x}}_{k|k}^{kf} + \left(\mathbf{D}_k + \mathcal{K}_k^{pf} \right) \left(\mathbf{x}_{k+1}^{pf} - \mathbf{f}^{pf}(\mathbf{x}_k^{pf}) \right) + \mathbf{f}^{pf}(\mathbf{x}_k^{pf}) \quad (4.81)$$

$$\mathbf{P}_{k+1|k}^{kf} = \bar{\mathbf{F}}_k^{kf} \mathbf{P}_{k|k}^{kf} \left(\bar{\mathbf{F}}_k^{kf} \right)^T + \mathbf{G}_k^{kf} \bar{\mathbf{Q}}_k^{kf} \left(\mathbf{G}_k^{kf} \right)^T - \mathcal{K}_k^{pf} \mathbf{S}_k^{pf} \left(\mathcal{K}_k^{pf} \right)^T \quad (4.82)$$

$$\mathcal{K}_k^{pf} = \bar{\mathbf{F}}_k^{kf} \mathbf{P}_{k|k}^{kf} \left(\mathbf{F}_k^{pf} \right)^T \left(\mathbf{S}_k^{pf} \right)^{-1} \quad (4.83)$$

$$\mathbf{S}_k^{pf} = \mathbf{G}_k^{pf} \mathbf{Q}_k^{pf} \left(\mathbf{G}_k^{pf} \right)^T + \mathbf{F}_k^{pf} \mathbf{P}_{k|k}^{kf} \left(\mathbf{F}_k^{pf} \right)^T \quad (4.84)$$

where

$$\mathbf{D}_k = \mathbf{G}_k^{kf} \mathbf{M}_k^T \left(\mathbf{G}_k^{pf} \mathbf{Q}_k^{pf} \right)^{-1} \quad (4.85)$$

$$\bar{\mathbf{F}}_k^{kf} = \mathbf{F}_k^{kf} - \mathbf{D}_k \mathbf{F}_k^{pf} \quad (4.86)$$

$$\bar{\mathbf{Q}}_k^{kf} = \mathbf{Q}_k^{kf} - \mathbf{M}_k^T \left(\mathbf{Q}_k^{pf} \right)^{-1} \mathbf{M}_k \quad (4.87)$$

The second density on the right hand side in Eq. (4.74) can be expressed recursively by using Bayes's rule repeatedly

$$p \left(\mathbf{X}_k^{pf} | \mathbf{Y}_k \right) = \frac{p(\mathbf{y}_k | \mathbf{X}_k^{pf}, \mathbf{Y}_{k-1}) p(\mathbf{x}_k^{pf} | \mathbf{X}_{k-1}^{pf}, \mathbf{Y}_{k-1})}{p(\mathbf{y}_k | \mathbf{Y}_{k-1})} p(\mathbf{X}_{k-1}^{pf} | \mathbf{Y}_{k-1}) \quad (4.88)$$

Since the system and measurement equations for \mathbf{x}_k^{pf} are nonlinear, the estimation in Eq. (4.88) is achieved by using the particle filter.⁷⁵ The distributions for

$p(\mathbf{y}_k | \mathbf{X}_k^{pf}, \mathbf{Y}_{k-1})$ and $p(\mathbf{x}_k^{pf} | \mathbf{X}_{k-1}^{pf}, \mathbf{Y}_{k-1})$ are given by

$$p(\mathbf{y}_{k+1} | \mathbf{X}_k^{pf}, \mathbf{Y}_k) = \mathcal{N}(\mathbf{h}(\hat{\mathbf{x}}_{k+1}^{pf}) + \mathbf{H}_{k+1} \hat{\mathbf{x}}_{k+1|k}^{kf}, \mathbf{H}_{k+1} \mathbf{P}_{k+1|k}^{kf} \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1}) \quad (4.89)$$

$$p(\mathbf{x}_k^{pf} | \mathbf{X}_k^{pf}, \mathbf{Y}_k) = \mathcal{N}(\mathbf{f}^{pf}(\mathbf{x}_k^{pf}) + \mathbf{F}_k^{pf} \hat{\mathbf{x}}_{k+1|k}^{kf}, \mathbf{F}_k^{pf} \mathbf{P}_{k|k}^{pf} (\mathbf{F}_k^{pf})^T + \mathbf{G}_k^{pf} \mathbf{Q}_k^{pf} (\mathbf{G}_k^{pf})^T) \quad (4.90)$$

For the particle filter algorithm, $p(\mathbf{x}_k^{pf} | \mathbf{X}_{k-1}^{pf}, \mathbf{Y}_{k-1})$ is utilized as an importance function

$$q(\mathbf{x}_k^{pf} | \mathbf{X}_{k-1}^{pf}, \mathbf{Y}_k) = p(\mathbf{x}_k^{pf} | \mathbf{X}_{k-1}^{pf}, \mathbf{Y}_{k-1}) \quad (4.91)$$

Then, the importance weight is calculated

$$w(\mathbf{X}_k^{pf}) = \frac{p(\mathbf{y}_k | \mathbf{X}_k^{pf}, \mathbf{Y}_{k-1})}{\rho(\mathbf{X}_{k-1}^{pf})} w(\mathbf{X}_{k-1}^{pf}) \quad (4.92)$$

For the update procedure, the resampling weight $\rho(\mathbf{X}_{k-1}^{pf})$ is chosen⁷⁵

$$\rho(\mathbf{X}_{k-1}^{pf}) = p(\mathbf{y}_k | \{\mathbf{X}_{k-1}^{pf}, \hat{\mathbf{x}}_{k|k-1}^{pf}\}, \mathbf{Y}_{k-1}) w(\mathbf{X}_{k-1}^{pf}) \quad (4.93)$$

where $\hat{\mathbf{x}}_{k|k-1}^{pf}$ is a prediction based on \mathbf{X}_{k-1}^{pf} and is typically given by

$$\hat{\mathbf{x}}_{k|k-1}^{pf} = \mathbf{f}^{pf}(\hat{\mathbf{x}}_{k-1}^{pf}) + \mathbf{F}_{k-1}^{pf} \hat{\mathbf{x}}_{k-1|k-1}^{pf} \quad (4.94)$$

For each $\mathbf{X}_{k-1}^{pf, (i)}$, the Kalman filter algorithms can be applied to estimates $\hat{\mathbf{x}}_{k|k}^{kf, (i)}$ and $\mathbf{P}_{k|k}^{kf, (i)}$. Details of the algorithm of the Rao-Blackwellised particle filter are summarized in Table 4.11.

As stated in the beginning of this section, the purpose of *Rao-Blackwellization* is to reduce the number of particles for a given estimation precision. This approach reduces the computational load, while the accuracy can be sustained.

An important special case is when the matrices \mathbf{F}_k^{pf} , \mathbf{G}_k^{pf} , \mathbf{F}_k^{kf} , \mathbf{G}_k^{kf} , and \mathbf{H}_k are independent of \mathbf{x}_k^{pf} . In this case the estimates of $\hat{\mathbf{x}}_k^{pf}$, \mathbf{x}_k^{kf} and $\hat{\mathbf{P}}_k^{pf}$ are straightfor-

Table 4.11 Rao-Blackwellized Particle Filter (RBPF) Algorithm

Initialization: At time $k = 0$

1. For $i = 1, \dots, N$, sample $\mathbf{x}_0^{pf, (i)} \sim p(\mathbf{x}_0^{pf})$ and set $\{\hat{\mathbf{x}}_0^{kf, (i)}, \mathbf{P}_0^{kf, (i)}\} = \{\mathbf{0}, \mathbf{P}_0^{kf}\}$
2. For each $i = 1, \dots, N$, calculate the weights $w_0^{(i)} = p(\mathbf{y}_0 | \mathbf{x}_0^{pf, (i)})$
and normalize $\bar{w}_0^{(i)} = \frac{w_0^{(i)}}{\sum_j w_0^{(j)}}$
3. For each $i = 1, \dots, N$, compute $\{\hat{\mathbf{x}}_{0|0}^{kf, (i)}, \mathbf{P}_{0|0}^{kf, (i)}\}$

Prediction and Update: For each time $k \geq 1$

1. For each $i = 1, \dots, N$, calculate $\rho_{k-1}^{(i)} = p(\mathbf{y}_k | \{\mathbf{X}_{k-1}^{pf, (i)}, \hat{\mathbf{x}}_{k|k-1}^{pf, (i)}\}, \mathbf{Y}_{k-1}) \bar{w}_{k-1}^{(i)}$ with
 $\hat{\mathbf{x}}_{k|k-1}^{pf, (i)} = \mathbf{f}^{pf}(\hat{\mathbf{x}}_{k-1}^{pf, (i)}) + \mathbf{F}_{k-1}^{pf} \hat{\mathbf{x}}_{k-1|k-1}^{pf, (i)}$
and normalize $\bar{\rho}_{k-1}^{(i)} = \frac{\rho_{k-1}^{(i)}}{\sum_j \rho_{k-1}^{(j)}}$
2. If resampling ($N_{\text{eff}} < N_{th}$), apply one of the resampling procedures described in Section 3.2.2 on $\{\bar{\rho}_{k-1}^{(i)}\}_{i=1}^N$
Otherwise $\rho_{k-1}^{(i)} = \frac{1}{N}$
3. For $i = 1, \dots, N$, sample $\mathbf{x}_k^{pf, (i)} \sim p(\mathbf{x}_k^{pf} | \mathbf{X}_{k|k-1}^{pf, (i)}, \mathbf{Y}_{k-1})$
4. For each $i = 1, \dots, N$, compute $\{\hat{\mathbf{x}}_{k|k-1}^{kf, (i)}, \mathbf{P}_{k|k-1}^{kf, (i)}\}$
5. For each $i = 1, \dots, N$, update $w_k^{(i)} = p(\mathbf{y}_k | \mathbf{X}_k^{pf, (i)}, \mathbf{Y}_{k-1}) \frac{\bar{w}_{k-1}^{(i)}}{\bar{\rho}_{k-1}^{(i)}}$
and normalize $\bar{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_j w_k^{(j)}}$
6. For each $i = 1, \dots, N$, compute $\{\hat{\mathbf{x}}_{k|k}^{kf, (i)}, \mathbf{P}_{k|k}^{kf, (i)}\}$

ward⁷⁵

$$\hat{\mathbf{x}}_k^{pf} = \sum_{i=1}^N \bar{w}_k^{(i)} \mathbf{x}_k^{pf, (i)} \quad (4.95)$$

$$\hat{\mathbf{x}}_k^{kf} \approx \sum_{i=1}^N \bar{w}_k^{(i)} \hat{\mathbf{x}}_{k|k}^{kf, (i)} \quad (4.96)$$

$$\mathbf{P}_k^{pf} = \sum_{i=1}^N \bar{w}_k^{(i)} \left(\hat{\mathbf{x}}_k^{pf, (i)} - \hat{\mathbf{x}}_k^{pf} \right) \left(\hat{\mathbf{x}}_k^{pf, (i)} - \hat{\mathbf{x}}_k^{pf} \right)^T \quad (4.97)$$

Even though we are interested in the state estimates, we would like to have an estimate of the covariance for the linear part. It can be shown that the covariance can be estimated by

$$\mathbf{P}_{k|k}^{kf} = \sum_{i=1}^N \bar{w}_k^{(i)} \left[\mathbf{P}_{k|k}^{kf, (i)} + \left(\hat{\mathbf{x}}_{k|k}^{kf, (i)} - \hat{\mathbf{x}}_{k|k}^{kf} \right) \left(\hat{\mathbf{x}}_{k|k}^{kf, (i)} - \hat{\mathbf{x}}_{k|k}^{kf} \right)^T \right] \quad (4.98)$$

4.10 Cramér-Rao Bounds for Nonlinear Filtering

For a general optimal nonlinear filtering problem, it was indicated in Chapter 1 that the optimal recursive Bayesian estimator requires the calculation of the posterior density of the state vector as a function of time. A closed form analytic solution to this optimal filtering problem is not tractable in general, and in practical applications nonlinear filtering is represented by an approximated and suboptimal filtering algorithm. Despite the absence of a closed form solution, the best achievable error performance for nonlinear filtering can be assessed by considering lower bounds on the mean squared error (MSE). Lower bounds give an indication of performance limitations, and it can be used to determine whether imposed performance is realistic or not.

A commonly used lower bound is the Cramér-Rao lower bound (CRLB), which is defined to be the inverse of the Fisher information matrix and provides a lower

bound on the performance of any unbiased estimator of an unknown parameter vector.⁴⁷ This provides a powerful tool that has been used to assess the performance of unbiased estimators of parameters for deterministic dynamical motion.⁶⁸ In the case of uncertain dynamical motion the posterior Cramér-Rao lower bound (PCRLB) has been used to determine performance bounds for recursive Bayesian estimators of the uncertain target state.⁸³ Determining PCRLBs represents an even more challenging problem. The reason is that for calculating the Fisher information matrix, it is necessary to consider both the effect of measurement uncertainty as well as uncertainty in the random state. The key research work for the PCRLBs done in Ref. 83 provides a Riccati-like recursion formula that sequentially determines the PCRLBs for state estimation for the nonlinear filtering problem. The approach is based on the discrete-time nonlinear system with additive Gaussian process and measurement noises. A more general derivation for determining PCRLBs has been proposed for the discrete-time nonlinear filtering problem,⁸⁴ where any additive Gaussian assumption can be avoided.

This chapter starts by defining the CRLB and providing some related background information. The general recursive PCRLB formulation for nonlinear filtering problem is described along with special cases (see Ref. 84 for details).

4.10.1 Cramér-Rao Lower Bounds

Let $\hat{\mathbf{x}}$ be an unbiased estimate of a parameter vector \mathbf{x} , based on the measurement vector \mathbf{y} . Then the CRLB for the error covariance matrix is defined to be the inverse of the Fisher information matrix \mathbf{J} ⁴⁷

$$\mathbf{P} \triangleq E \left\{ [\hat{\mathbf{x}} - \mathbf{x}] [\hat{\mathbf{x}} - \mathbf{x}]^T \right\} \geq \mathbf{J}^{-1} \quad (4.99)$$

where the inequality means that the difference $\mathbf{P} - \mathbf{J}^{-1}$ is a positive semi-definite matrix. If \mathbf{x} is an unknown and random parameter vector, the posterior Cramer-Rao lower bound (PCRLB) is given by⁴⁷

$$\mathbf{J}_{ij} = E \left\{ -\frac{\partial^2 \log p(\mathbf{y}, \mathbf{x})}{\partial x^i \partial x^j} \right\} \quad (4.100)$$

where $p(\mathbf{y}, \mathbf{x})$ is the joint probability density function, and the expectation $E\{\cdot\}$ is with respect to both \mathbf{x} and \mathbf{y} . Let ∇ and Δ be operators of the first and second-order partial derivatives, respectively

$$\nabla_{\mathcal{X}} \equiv \left[\frac{\partial}{\partial \mathcal{X}_1}, \dots, \frac{\partial}{\partial \mathcal{X}_n} \right]^T \quad (4.101)$$

$$\Delta_{\mathcal{X}}^{\mathcal{Y}} \equiv \nabla_{\mathcal{X}} \nabla_{\mathcal{Y}}^T \quad (4.102)$$

Then, Eq. (4.100) can be rewritten by

$$\mathbf{J} = E \{ -\Delta_{\mathbf{x}}^{\mathbf{y}} \log p(\mathbf{y}, \mathbf{x}) \} \quad (4.103)$$

Since $p(\mathbf{y}, \mathbf{x}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, it can be easily shown that the information matrix \mathbf{J} can be decomposed into two additive parts⁸⁴

$$\mathbf{J} = \mathbf{J}_{\mathcal{D}} + \mathbf{J}_{\mathcal{P}} \quad (4.104)$$

where $\mathbf{J}_{\mathcal{D}}$ represents the information obtained from the measurement data, and $\mathbf{J}_{\mathcal{P}}$ represents the a priori information, respectively

$$\mathbf{J}_{\mathcal{D}} = E \{ -\Delta_{\mathbf{x}}^{\mathbf{y}} \log p(\mathbf{y}|\mathbf{x}) \} \quad (4.105)$$

$$\mathbf{J}_{\mathcal{P}} = E \{ -\Delta_{\mathbf{x}}^{\mathbf{x}} \log p(\mathbf{x}) \} \quad (4.106)$$

4.10.2 Posterior Cramér-Rao Lower Bounds

Consider the general discrete-time nonlinear system

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{w}_k) \quad (4.107)$$

$$\mathbf{y}_{k+1} = \mathbf{h}_{k+1}(\mathbf{x}_{k+1}, \mathbf{v}_{k+1}) \quad (4.108)$$

where \mathbf{x} is the state vector, \mathbf{y} is the measurement vector, \mathbf{f}_k and \mathbf{h}_{k+1} are nonlinear functions of \mathbf{x}_k , and \mathbf{w}_k and \mathbf{v}_{k+1} are independent white noise processes. Suppose measurements are available at discrete time epochs. The purpose is to calculate the PCRLB for unbiased estimators $\hat{\mathbf{x}}_k$ of the state vector \mathbf{x}_k , given the available sensor measurement set $\mathbf{Y}_k = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k]$

The sequence of the posterior Fisher information matrix for estimating state vector \mathbf{x}_k obeys the Riccati-like recursion equation given by

$$\mathbf{J}_{k+1} = \mathbf{D}_k^{22} - \mathbf{D}_k^{21} (\mathbf{J}_k + \mathbf{D}_k^{11})^{-1} \mathbf{D}_k^{12} \quad (4.109)$$

where

$$\mathbf{D}_k^{11} = E \left\{ -\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right\} \quad (4.110)$$

$$\mathbf{D}_k^{12} = E \left\{ -\Delta_{\mathbf{x}_k}^{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right\} = (\mathbf{D}_k^{21})^T \quad (4.111)$$

$$\mathbf{D}_k^{22} = E \left\{ -\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right\} + E \left\{ -\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \log p(\mathbf{y}_{k+1} | \mathbf{x}_{k+1}) \right\} \quad (4.112)$$

Note that the initial information matrix \mathbf{J}_0 can be calculated from the a priori probability density function $p(\mathbf{x}_0)$

$$\mathbf{J}_0 = E \left\{ -\Delta_{\mathbf{x}}^{\mathbf{x}} \log p(\mathbf{x}_0) \right\} \quad (4.113)$$

4.10.3 Posterior Cramér-Rao Lower Bound for Gaussian Noise

In this section the special case where the process and measurement noises are additive Gaussian is considered.

4.10.3.1 Nonlinear Gaussian Filtering Problem

Suppose that the nonlinear system has the form

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k \quad (4.114)$$

$$\mathbf{y}_{k+1} = \mathbf{h}_{k+1}(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} \quad (4.115)$$

where \mathbf{w}_k and \mathbf{v}_{k+1} are independent white Gaussian noise processes with zero mean and covariance matrices \mathbf{Q}_k and \mathbf{R}_{k+1} , respectively. Based on these assumptions, it follows that⁸⁴

$$-\log p(\mathbf{x}_{k+1}|\mathbf{x}_k) = c_1 + \frac{1}{2} [\mathbf{x}_{k+1} - \mathbf{f}_k(\mathbf{x}_k)]^T \mathbf{Q}_k^{-1} [\mathbf{x}_{k+1} - \mathbf{f}_k(\mathbf{x}_k)] \quad (4.116)$$

$$-\log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}) = c_2 + \frac{1}{2} [\mathbf{y}_{k+1} - \mathbf{h}_{k+1}(\mathbf{x}_{k+1})]^T \mathbf{R}_{k+1}^{-1} [\mathbf{y}_{k+1} - \mathbf{h}_{k+1}(\mathbf{x}_{k+1})] \quad (4.117)$$

where c_1 and c_2 are constants. Then, each term in the recursive Fisher information matrix equation is computed by

$$\mathbf{D}_k^{11} = E \left\{ [\nabla_{\mathbf{x}_k} \mathbf{f}_k^T(\mathbf{x}_k)] \mathbf{Q}_k^{-1} [\nabla_{\mathbf{x}_k} \mathbf{f}_k^T(\mathbf{x}_k)]^T \right\} \quad (4.118)$$

$$\mathbf{D}_k^{12} = -E \left\{ [\nabla_{\mathbf{x}_k} \mathbf{f}_k^T(\mathbf{x}_k)] \mathbf{Q}_k^{-1} \right\} = (\mathbf{D}_k^{21})^T \quad (4.119)$$

$$\mathbf{D}_k^{22} = \mathbf{Q}_k^{-1} + E \left\{ [\nabla_{\mathbf{x}_{k+1}} \mathbf{h}_{k+1}^T(\mathbf{x}_{k+1})] \mathbf{R}_{k+1}^{-1} [\nabla_{\mathbf{x}_{k+1}} \mathbf{h}_{k+1}^T(\mathbf{x}_{k+1})]^T \right\} \quad (4.120)$$

The most difficult task in practical application of the posterior CRLB is the calculation of the expectation operator $E\{\cdot\}$. A Monte Carlo integration method can be applied to implement the theoretical posterior CRLB formulation.

4.10.3.2 Linear Gaussian Filtering Problem

Suppose that the linear system is given by

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \quad (4.121)$$

$$\mathbf{y}_{k+1} = \mathbf{H}_{k+1} \mathbf{x}_{k+1} + \mathbf{v}_{k+1} \quad (4.122)$$

where \mathbf{w}_k and \mathbf{v}_{k+1} are independent white Gaussian noise processes with zero mean and covariance matrices \mathbf{Q}_k and \mathbf{R}_{k+1} , respectively. Then the terms in the Fisher information matrix equation are expressed by⁸⁴

$$\mathbf{D}_k^{11} = \mathbf{F}_k^T \mathbf{Q}_k^{-1} \mathbf{F}_k \quad (4.123)$$

$$\mathbf{D}_k^{12} = -\mathbf{F}_k^T \mathbf{Q}_k^{-1} = (\mathbf{D}_k^{21})^T \quad (4.124)$$

$$\mathbf{D}_k^{22} = \mathbf{Q}_k^{-1} + \mathbf{H}_{k+1}^T \mathbf{R}_{k+1}^{-1} \mathbf{H}_{k+1} \quad (4.125)$$

Substituting these into the recursive Fisher information \mathbf{J}_{k+1} in Eq. (8.1) yields

$$\mathbf{J}_{k+1} = \mathbf{Q}_k^{-1} + \mathbf{H}_{k+1}^T \mathbf{R}_{k+1}^{-1} \mathbf{H}_{k+1} - \mathbf{Q}_k^{-1} \mathbf{F}_k (\mathbf{J}_k + \mathbf{F}_k^T \mathbf{Q}_k^{-1} \mathbf{F}_k)^{-1} \mathbf{F}_k^T \mathbf{Q}_k^{-1} \quad (4.126)$$

Finally, the recursive Fisher information matrix equation \mathbf{J}_{k+1} is rewritten by applying the Matrix inversion lemma⁸⁴

$$\mathbf{J}_{k+1} = (\mathbf{Q}_k + \mathbf{F}_k \mathbf{J}_k^{-1} \mathbf{F}_k^T)^{-1} + \mathbf{H}_{k+1}^T \mathbf{R}_{k+1}^{-1} \mathbf{H}_{k+1} \quad (4.127)$$

Note that the PCRLB for the linear Gaussian filtering problem is equivalent to the error covariance matrix of the Kalman filter.

CHAPTER V

ADAPTIVE FILTERING

This chapter presents new practical adaptive nonlinear filters for recursive estimation of the state and parameters of nonlinear systems with unknown noise statistics. The adaptive nonlinear filters combine adaptive estimation techniques for system noise statistics with the nonlinear filters that include the unscented Kalman filter and divided difference filter. The purpose of the integrated filters is to not only compensate for the nonlinearity effects neglected from linearization by utilizing nonlinear filters, but also to take into account the system modeling errors by adaptively estimating the noise statistics and unknown parameters.

5.1 Introduction

The optimality of the linear Kalman filtering depends on the assumptions that the first and second moment statistics of the observational and system noise are correctly specified, as well as the *a priori* information of the state values is properly selected.¹ For nonlinear systems, however, it is difficult to obtain optimal filtering algorithms for a finite dimensional system. The exact nonlinear equations are approximated for suboptimal nonlinear estimation. Thus, nonlinearities neglected by approximations of the system equations and methods used for the approximation can affect the performance of the nonlinear filters. There have been two approaches for approximated nonlinear filtering algorithms.²⁸ Most nonlinear filters employ a Taylor-series expansion or interpolation method in order to approximate the nonlinear system and measurement equations to compute the conditional mean and covariance. Another approach is based on the determination of the exact equations satisfied by the conditional density functions and conditional expectations.²⁸ A recursive filtering

algorithm can be derived by approximating the conditional density functions. However, the computational burden for approximating the conditional density functions increases as the dimension of the state vector increases. The simplest and most widely used approximate nonlinear filter is the extended Kalman filter (EKF), which works on the principle that the state distribution is approximated by a Gaussian random variable, and the state is then propagated through the first-order linearization of the nonlinear equations. The series approximations can, however, introduce large errors due to the neglected nonlinearities, which lead to degraded performance of the filters. For example, the difficulty appears when the noise in the measurements is small or of the same order compared to the nonlinearity in the measurement functions.⁸⁵

Recently there have been researches about new efficient nonlinear filtering techniques^{8,10,12} in which the nonlinear filters generalize elegantly to nonlinear systems without the burdensome linearization steps. Thus, truncation errors due to linearization can be compensated. These filters include the unscented Kalman filter (UKF), the central difference filter (CDF) and the divided difference filter (DDF), and they are also called Sigma Point Filters (SPFs) in a unified way.³¹ Lee and Alfriend^{31,86} have utilized the nonlinear filters for state and parameter estimation in orbit determination and prediction, and have shown that the state-of-art new nonlinear filters lead to faster and accurate convergent solutions. Even though the SPFs can mitigate the nonlinearity effects and produce accurate estimation of the state variables, the nonlinear filters can contain errors in the estimation if the statistics of the system and measurement noise are incorrectly specified due to inaccurate observations or unmodeled accelerations in the system dynamic model.⁷

In order to avoid these problems when the filtering algorithms are implemented process and measurement noise statistics must be selected properly. This procedure is a process for tuning the filter, which is usually performed manually by a trial

and error method, and is repeated until the best choice of the noise parameters is determined. The process of selecting the appropriate values of the noise covariance matrices is discussed in detail in Maybeck.⁸⁷ For automated filter tuning, Powell⁸⁸ has utilized a numerical optimization algorithm, the Downhill Simplex Method, where the process covariance matrix is determined by minimizing the performance function, that is simply the sum of the RMS state estimation errors. Even though the tuning method can compensate for the model errors or uncertainties, it can result in limited filtering performance because the tuning method provides constant noise variances. However, in reality the model error characteristics have time-varying values.

To efficiently improve the description of model errors and robustly handle uncertainties of the sensor and process noises, an adaptive filter can be applied such that the values of the covariance matrices can be estimated so as to produce consistency between the corresponding residuals and their statistics in the prediction evolution equations.²³ There have been many investigations of adaptive filtering of nonlinear systems in various engineering researches. The method of maximum likelihood estimation (MMLE) is a technique applied to Kalman Filters. This was originally proposed by Mehra,²³ and variants of the technique have been used in many filtering applications. Sage and Husa²⁴ presented the development of optimal adaptive Bayes estimation algorithms for discrete linear filtering with unknown prior statistics. Most of the adaptive filtering methods are applied to linear systems, but not to nonlinear systems.

In this work the maximum likelihood estimator of Maybeck⁸⁷ and the more intuitive approach of Myers and Tapley²⁵ are considered as noise adaptation algorithms. The basic premise of this method is to use the measurement and state residuals to modify the parameter values for sensor and process noises. Since the methods are derived based on linear systems a variant algorithm to the nonlinear systems has

been employed to satisfactorily estimate the system error covariance.⁸⁹ The modification was made by introducing a window scale factor that is decided by a trial-error method. In this paper, a new procedure is designed where the numerical optimization technique introduced by Powell is utilized to automatically estimate the scale factor for the system noise covariance adaptation. The new automated adaptive algorithms are integrated into the UKF and DDF such that new efficient adaptive sigma point filtering (ASPF) algorithms are developed. The proposed adaptive nonlinear filters focus on taking into account the unknown time-varying noise statistics of dynamic systems, as well as compensating the modeling errors due to the neglected nonlinearity effects and unknown system parameters. For the state and parameter estimation the joint state estimation method is used by combining the parameters and states into a vector form.

5.2 Optimality Conditions of Kalman Filter

When the KF is optimal, the innovation sequence \mathbf{v}_k should be white with zero mean.⁹⁰ In other words, this means that if $\hat{\mathbf{x}}_k^+$ is an optimal estimate, there is no information left in the innovation sequence to improve the prediction. The *a priori* estimate error is defined by the difference between the true and predicted state vectors

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k^- \quad (5.1)$$

Then, the innovation vector is written by

$$\mathbf{v}_k = \mathbf{H}\mathbf{e}_k + \mathbf{v}_k \quad (5.2)$$

Substituting the Kalman filtering equations and the system model into Eq. (5.1) provides

$$\mathbf{e}_k = \mathbf{F} [\mathbf{I} - \mathcal{K}_{k-1} \mathbf{H}] \mathbf{e}_{k-1} + \mathbf{w}_{k-1} - \mathbf{F} \mathcal{K}_{k-1} \mathbf{v}_{k-1} \quad (5.3)$$

Taking the expectation $E \{ \cdot \}$ of the error vector \mathbf{e}_k , we have

$$E \{ \mathbf{e}_k \} = \mathbf{F} [\mathbf{I} - \mathcal{K}_{k-1} \mathbf{H}] E \{ \mathbf{e}_{k-1} \} \quad (5.4)$$

Thus, $E \{ \mathbf{e}_k \}$ eventually depends only on $E \{ \mathbf{e}_0 \}$, which is assumed to be zero. The expectation of the innovation vector is given by

$$E \{ \mathbf{v}_k \} = \mathbf{H} E \{ \mathbf{e}_k \} \quad (5.5)$$

Thus, the innovation vector has zero mean. If restoring Eq. (5.3) recursively with time lag $j > 0$, then the error at any epoch k can be expressed⁹¹

$$\begin{aligned} \mathbf{e}_k = & \left\{ \prod_{i=k-j}^{k-1} \mathbf{F} [\mathbf{I} - \mathcal{K}_i \mathbf{H}] \right\} \mathbf{e}_{k-j} + \sum_{i=k-j}^{k-1} \left\{ \prod_{m=i+1}^{k-1} \mathbf{F} [\mathbf{I} - \mathcal{K}_m \mathbf{H}] \right\} \mathbf{w}_i \\ & - \sum_{i=k-j}^{k-1} \left\{ \prod_{m=i+1}^{k-1} \mathbf{F} [\mathbf{I} - \mathcal{K}_m \mathbf{H}] \right\} \mathbf{F} \mathcal{K}_i \mathbf{v}_i \end{aligned} \quad (5.6)$$

The first component of Eq. (5.6) will attenuate with time if the Kalman gain is optimal. The second term shows the influence of input noise, and the last term illustrates the process of measurement noise smoothing. The lag j covariance of the innovation sequence is given by⁹¹

$$\mathbf{S}_j = \mathbf{H} \left\{ \prod_{i=k-j+1}^{k-1} \mathbf{F} [\mathbf{I} - \mathcal{K}_i \mathbf{H}] \right\} \{ \mathbf{F} [\mathbf{I} - \mathcal{K}_{k-j} \mathbf{H}] \mathbf{P}_{k-j}^- \mathbf{H}^T - \mathbf{F} \mathcal{K}_{k-j} \mathbf{R} \} \quad (5.7)$$

If the Kalman gain \mathcal{K}_{k-j} is optimal, then the term \mathbf{S}_j becomes zero, $\mathbf{S}_j = 0$ for $j \neq 0$. This means that the innovation sequence is white when the filter is optimal. Therefore, a well-designed optimal filter will attenuate the initial state errors, smooth the effects of system and measurement errors, and whiten the innovation sequence.

5.3 Maybeck's Estimator

Myers and Tapley²⁵ derived an adaptive filter based on empirical estimators that can estimate both the covariance matrix and the bias of system model errors. To obtain an explicit maximum-likelihood estimator it is assumed that the system noise is slowly varying, but remains stationary over N time steps. The process noise adaptation algorithm is expressed as follows

$$\hat{\mathbf{Q}}_k = \frac{1}{N} \sum_{i=k-N+1}^k \{ \mathcal{K}_j^T \mathbf{v}_j \mathbf{v}_j^T \mathcal{K}_j^T - [\mathbf{F} \mathbf{P}_{i-1} \mathbf{F}^T - \mathbf{P}_i^+] \} \quad (5.8)$$

where \mathbf{v}_k is the innovation vector, and $\hat{\mathbf{Q}}_0$ must be specified.

Maybeck used a maximum-likelihood estimator for designing an adaptive filter that can estimate the system errors covariance matrix. He considered that the noise was essentially stationary over N sample periods, but he considered only the case of an unbiased system noise, which is the difference between the Maybeck and Myers methods. It has been shown that they are identical under Maybeck's assumptions that the bias is known and equal to zero.⁹¹ In this case, the unbiased estimator of MT becomes

$$\hat{\mathbf{Q}}_k = \frac{1}{N} \sum_{i=k-N+1}^k \{ \mathbf{q}_{s,i} \mathbf{q}_{s,i}^T - [\mathbf{F} \mathbf{P}_{i-1} \mathbf{F}^T - \mathbf{P}_i^+] \} \quad (5.9)$$

where the term $\mathbf{q}_{s,i}$ is given by

$$\mathbf{q}_{s,i} \equiv \hat{\mathbf{x}}_i^+ - \hat{\mathbf{x}}_i^- = \mathcal{K}_i [\mathbf{y}_i - \mathbf{H} \hat{\mathbf{x}}_i^-] \quad (5.10)$$

This can be written in terms of the innovation vector

$$\mathbf{q}_{s,i} = \mathcal{K}_i \mathbf{v}_i \quad (5.11)$$

which leads to the same equation of the process noise adaptive algorithms in Eq. (5.8). Note that the adaptive Kalman filter algorithms are based on linear systems, but

not nonlinear systems, thus it can't be applied to the nonlinear systems directly. Therefore, for nonlinear system application modification is necessary.

5.4 Automated Adaptive Nonlinear Estimator

Most of the adaptive Kalman filters^{23,25,87} have been derived based on linear dynamic systems, but there are only a few publications about adaptive filtering for nonlinear system.^{27,89} The routines presented in this thesis are nonlinear adaptive algorithms, which are modified from the linear adaptive algorithm in order to integrate the sigma point filters to nonlinear systems. A new procedure is introduced to estimate the system error covariance with a numerical optimization method. In principle, an adaptive filter can estimate both the system and the observational errors. However, adaptive filtering algorithms that try to update both the observational noise and the system noise are not robust, since it is not easy to distinguish between errors in \mathbf{Q}_k and \mathbf{R}_k .⁹¹ Usually, the measurement noise statistics are relatively well known compared to the system model error. In this paper, the adaptive estimation of the process noise covariance \mathbf{Q}_k is considered.

5.4.1 Process Noise Covariance Estimation

The modified adaptive filtering algorithm by Busse⁸⁹ is based on Maybeck's adaptation algorithm, and requires a proper selection of a window size that controls the level of the variance update. In this section, the algorithms are briefly reviewed. From Maybeck's unbiased adaptation algorithm in Eq. (5.9), the observation of \mathbf{Q}_k was rewritten as the difference between the state estimate before and after the measurement update

$$\mathbf{Q}^* = \Delta \mathbf{x}_k \Delta \mathbf{x}_k^T + \mathbf{P}_k^- - \mathbf{P}_k^+ - \hat{\mathbf{Q}}_k^- \quad (5.12)$$

where the term $\Delta \mathbf{x}_k \equiv \hat{\mathbf{x}}_k^+ - \hat{\mathbf{x}}_k^- = \mathbf{q}_{s,k}$ is the state residual and represents the difference between the state estimate before and after the measurement update. $\hat{\mathbf{Q}}_k^-$ is the current expected process noise covariance. If the residual has a large value, then it indicates that the future state prediction is not accurate enough. The first term in the above equation is a measure of the state residual, and the next term is a measure of the correction in the expected change of covariance. It is rewritten and becomes obvious conceptually

$$\mathbf{Q}^* = \Delta \mathbf{x}_k \Delta \mathbf{x}_k^T - \left[\mathbf{P}_k^+ - \left(\mathbf{P}_k^- - \hat{\mathbf{Q}}_k^- \right) \right] \quad (5.13)$$

The equation shows that \mathbf{Q}^* is the residual minus the change in the *a posteriori* covariances between two consecutive time steps.⁸⁹ The measure of the process noise \mathbf{Q}^* is then combined with the current estimate $\hat{\mathbf{Q}}_k^-$ in a moving average

$$\hat{\mathbf{Q}}_k^+ = \hat{\mathbf{Q}}_k^- + \frac{1}{\gamma} \left(\mathbf{Q}^* - \hat{\mathbf{Q}}_k^- \right) \quad (5.14)$$

where γ is the window size that controls the level of expected update change and needs to be selected through a trial-error method. If γ is small, then each update is weighted heavily, but if γ is large, then each update has a small effect. The performance of the adaptive routine is very sensitive to the selection of γ , and thus should be selected for each application. Now, the discrete formulation is then placed into continuous form.

That is, if

$$\hat{\mathbf{Q}}_k^+ = \begin{bmatrix} \mathbf{Q}_{xx} & \mathbf{Q}_{x\dot{x}} \\ \mathbf{Q}_{\dot{x}x} & \mathbf{Q}_{\dot{x}\dot{x}} \end{bmatrix} \quad (5.15)$$

then, diagonalization of the process noise covariance of the velocity part can be made

$$\mathbf{q}_{\dot{x}\dot{x}} = \text{diag}(\mathbf{Q}_{\dot{x}\dot{x}}) \frac{1}{\Delta t} \quad (5.16)$$

Now redefine the estimated process noise covariance matrix

$$\hat{\mathbf{Q}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{q}_{\dot{\mathbf{x}}\dot{\mathbf{x}}} \end{bmatrix} \quad (5.17)$$

This updated estimate $\hat{\mathbf{Q}}$ is used for the state propagation between time-step k and $k + 1$. Note that the proposed adaptive algorithms highly depend on the selection of the weight factor γ . In order to provide the consistent, optimal performance of the proposed adaptive filter, we suggest an efficient calibration method in the next section.

5.4.2 Automated Calculation of Scale Factor

Now the question that comes to mind is how the scaling factor can be determined. The easiest way for deciding the scale factor is a manual trial-error process that is continued until the filter produces sub-optimal or near-optimal estimation results. However, this method costs too much in time and effort. An alternative is to use existing the numerical optimization algorithms that are well documented in several programming languages (See Ref. 92 for details).⁹² In this thesis, a derivative-free numerical optimization technique is utilized for the automated calibration of the weight scale factor. Powell⁸⁸ has used the numerical optimization method called the *Downhill Simplex* algorithm in order to tune the parameters of the process noise covariance. However, the method introduced here to decide the scale factor for the nonlinear adaptive estimator requires only function evaluations, and does not require derivatives such as the Jacobian or Hessian matrix. The advantage of the numerical method over the manual tuning is that it is an automated tool and designed to save time and effort for filter designers.

In order to apply the numerical optimization algorithm to the filter tuning, the

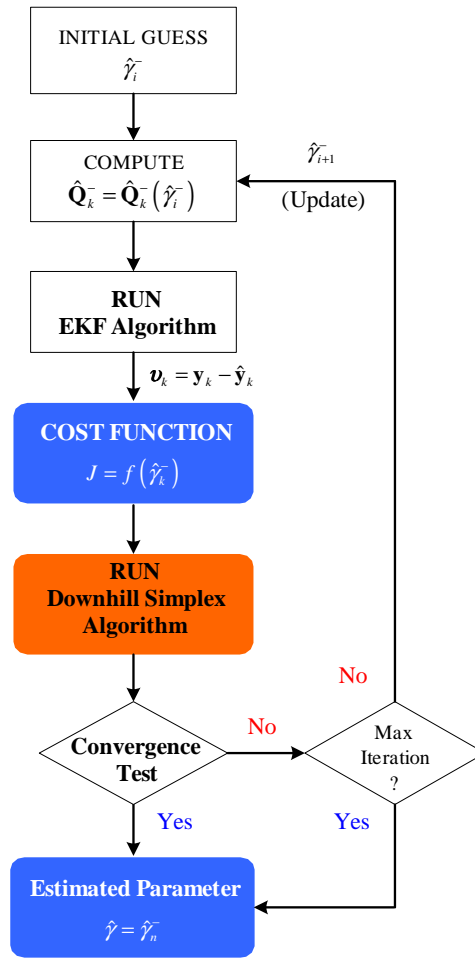


Fig. 5.1 Downhill Simplex Algorithm for Automated Parameter Tuning

tuning problem must be expressed as a numerical optimization or function minimization problem. Powell used the objective function in terms of the RMS of the state estimation errors that is the difference between the true and estimated values. The performance index J_k is used for the Monte Carlo performance simulation for each k th Monte Carlo samples. Then, the overall performance index J is obtained by

calculating the RMS of the J_k values.

$$J_k(q_{11}, \dots, q_{nn}) = \left[\frac{1}{N} \sum_{i=1}^N \{(\hat{x}_{1,i} - x_{1,i})^2 + \dots + (\hat{x}_{n,i} - x_{n,i})^2\} \right]^{1/2} \quad (5.18)$$

where $(q_{11}, q_{22}, \dots, q_{nn})$ are the parameters for representing the process covariance matrix, N is the total number of observation data points, and n is the dimension of the state vector.

The Downhill Simplex method algorithm is robust, but is best suited for a problem whose computational burden is small.⁹² In this thesis, however, the numerical Downhill Simplex method automatically calculates only a scalar weight factor, therefore the load of numerical computation can be reduced. The objective function is constructed by the innovation vector concept instead of the estimate error, which is not practical in a real application due to the absence of the true state information.

$$J_k(\gamma) = \left[\frac{1}{N} \sum_{i=1}^N \{v_{1,i}^2 + v_{2,i}^2 + \dots + v_{m,i}^2\} \right]^{1/2} \quad (5.19)$$

where γ is the weight factor, and m is the dimension of the observation vector. Note that the transient estimation result of the filtering algorithms contains large state estimation errors or the innovation errors, then the cost function can omit the first part of the transient estimation. Fig. 5.1 summarizes the procedures of the Simplex algorithms.

5.5 Adaptive Unscented Kalman Filter

This section illustrates the integration of the proposed adaptive filtering algorithms with the sigma point filters (SPFs) such as the UKF and DDF for more enhanced nonlinear filtering algorithms. Thus, the *adaptive sigma point filters* (ASPFs) lead to the *adaptive unscented Kalman filter* (AUKF) and the *adaptive divided difference filter*

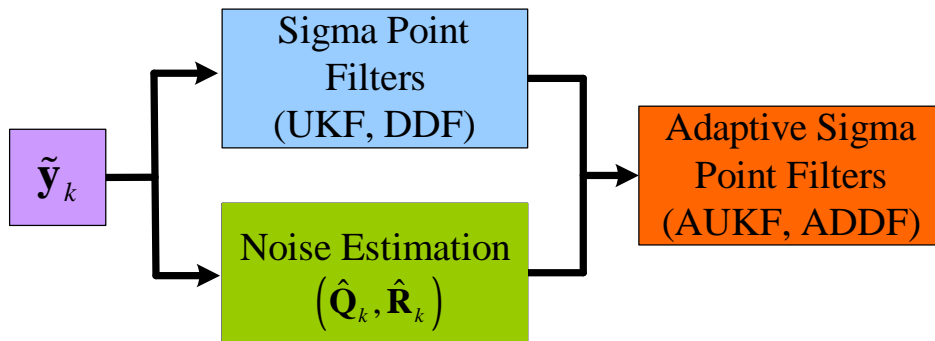


Fig. 5.2 Structure of Adaptive Sigma Point Filters

(ADDF). The objective of the integrated adaptive nonlinear filters focus is to take into account the incorrect time-varying noise statistics of dynamical systems, as well as to compensate the nonlinearity effects neglected by linearization. In Fig. 5.2 the overall concept of the adaptive sigma point filter is depicted for intuitive understanding.

5.5.1 Unscented Kalman Filter Algorithms

The unscented Kalman filter algorithms are developed for discrete-time nonlinear equations

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{w}_k, k) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k, k)\end{aligned}\tag{5.20}$$

where $\mathbf{x}_k \in \Re^{n \times 1}$ is the $n \times 1$ state vector, $\mathbf{y}_k \in \Re^{m \times 1}$ is the $m \times 1$ observation vector. $\mathbf{w}_k \in \Re^{q \times 1}$ is the $q \times 1$ state noise process vector and $\mathbf{v}_k \in \Re^{r \times 1}$ is the $r \times 1$ measurement noise vector. It is assumed that the noise vectors are uncorrelated white Gaussian processes with expected means and covariance processes with unknown covariances given by \mathbf{Q}_k and \mathbf{R}_k , respectively. The original state vector is redefined as an augmented state vector along with noise variables and an augmented covariance

matrix on the diagonal is reconstructed

$$\mathbf{x}_k^a = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{w}_k \end{bmatrix}, \mathbf{P}_k^a = \begin{bmatrix} \mathbf{P}_k & \mathbf{P}_k^{xw} \\ \mathbf{P}_k^{xw} & \mathbf{Q}_k \end{bmatrix} \quad (5.21)$$

where \mathbf{P}_k^{xw} is the correlation between the process noise and the error in the state estimate. Then, the set of $\{2(n+q)+1\}$ sigma points $\mathcal{X}_k^a \in \Re^{\{2(n+q)+1\}}$ is constructed

$$\mathcal{X}_k^a = \left[\hat{\mathbf{x}}_k^a \quad \hat{\mathbf{x}}_k^a + \sqrt{(n+q+\lambda)\mathbf{P}_k^a} \quad \hat{\mathbf{x}}_k^a - \sqrt{(n+q+\lambda)\mathbf{P}_k^a} \right] \quad (5.22)$$

where $\lambda = \alpha^2(n + \kappa) - n$ includes scaling parameters. α controls the size of the sigma point distribution, should be a small number ($0 \leq \alpha \leq 1$), and provides an extra degree of freedom to fine tune the higher order moments $\kappa = 3 - n$. As for the state propagation step, the predicted state vector $\hat{\mathbf{x}}_{k+1}^-$ and its predicted covariance \mathbf{P}_{k+1}^- are computed using the propagated sigma point vectors.

$$\mathcal{X}_{i,k+1}^x = \mathbf{f}(\mathcal{X}_{i,k}^x, \mathcal{X}_{i,k}^w, k) \quad (5.23)$$

$$\hat{\mathbf{x}}_{k+1}^- = \sum_{i=0}^{2(n+q)} W_i^{(m)} \mathcal{X}_{i,k+1}^x \quad (5.24)$$

$$\mathbf{P}_{k+1}^- = \sum_{i=0}^{2(n+q)} W_i^{(c)} \{\mathcal{X}_{i,k+1}^x - \hat{\mathbf{x}}_{k+1}^-\} \{\mathcal{X}_{i,k+1}^x - \hat{\mathbf{x}}_{k+1}^-\}^T \quad (5.25)$$

where $\mathcal{X}_{i,k}^x$ is a sigma point vector of the first n elements of $\mathcal{X}_{i,k}^a$, and $\mathcal{X}_{i,k}^w$ is a sigma point vector of the next q elements of $\mathcal{X}_{i,k}^a$, respectively.

Similarly, the predicted observation vector $\hat{\mathbf{y}}_{k+1}^-$ and its predicted covariance \mathbf{P}_{k+1}^{yy} are also calculated as

$$\mathcal{Y}_{i,k+1} = \mathbf{h}(\mathcal{X}_{i,k+1}^x, k+1) \quad (5.26)$$

$$\hat{\mathbf{y}}_{k+1}^- = \sum_{i=0}^{2(n+q)} W_i^{(m)} \mathcal{Y}_{i,k+1} \quad (5.27)$$

$$\mathbf{P}_{k+1}^{yy} = \sum_{i=0}^{2(n+q)} W_i^{(c)} \{\mathcal{Y}_{i,k+1} - \hat{\mathbf{y}}_{k+1}^-\} \{\mathcal{Y}_{i,k+1} - \hat{\mathbf{y}}_{k+1}^-\}^T \quad (5.28)$$

where $W_i^{(m)}$ is the weight for the mean and $W_i^{(c)}$ is the weight for the covariance, respectively

$$W_i^{(m)} = \begin{cases} \lambda/n + \lambda & i = 0 \\ 1/\{2(n + \lambda)\} & i = 1, \dots, 2n \end{cases} \quad (5.29)$$

$$W_i^{(c)} = \begin{cases} \lambda/(n + \lambda) + (1 - \alpha^2 + \beta) & i = 0 \\ 1/\{2(n + \lambda)\} & i = 1, \dots, 2n \end{cases} \quad (5.30)$$

β is a third parameter that makes further higher order effects to be incorporated by adding the weighting of the *zeroth* sigma point of the calculation of the covariance, and $\beta = 2$ is the optimal for Gaussian distributions. The filter gain \mathcal{K}_{k+1} is computed by

$$\mathcal{K}_{k+1} = \mathbf{P}_{k+1}^{xy} (\mathbf{P}_{k+1}^{vv})^{-1} \quad (5.31)$$

and the cross correlation matrix is determined

$$\mathbf{P}_{k+1}^{xy} = \sum_{i=0}^{2(n+q)} W_i^{(c)} \{\mathcal{X}_{i,k+1}^x - \hat{\mathbf{x}}_{k+1}^-\} \{\mathcal{Y}_{i,k+1} - \hat{\mathbf{y}}_{k+1}^-\}^T \quad (5.32)$$

The estimated state vector $\hat{\mathbf{x}}_{k+1}^+$ and updated covariance \mathbf{P}_{k+1}^+ are given by

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathcal{K}_{k+1} \mathbf{P}_{k+1}^{vv} \mathcal{K}_{k+1}^T \quad (5.33)$$

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathcal{K}_{k+1} \mathbf{v}_{k+1} \quad (5.34)$$

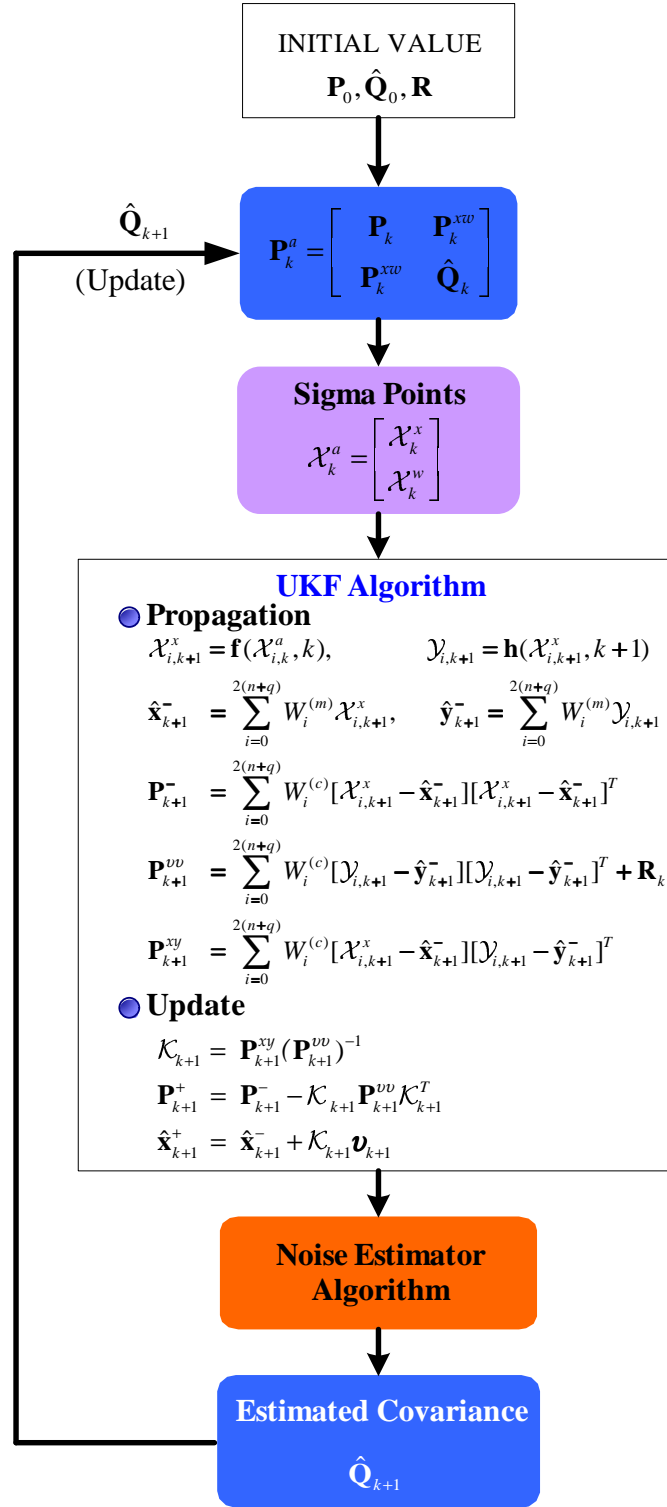


Fig. 5.3 Diagram for Adaptive Unscented Kalman Filter Algorithm

5.5.2 Adaptive Unscented Kalman Filtering

Note that for implementing the proposed adaptation algorithm into the sigma point filters the expression for the process noise covariance matrix in the predicted covariance equation should be explicit. However, the process noise covariance term in the UKF algorithm is implicitly expressed in the predicted covariance equation. Thus, the noise adaptive estimator can't be directly implemented.

There are two approaches that can integrate the proposed adaptive algorithm into the unscented Kalman filtering. The first method is to make the assumption that for the special case both the process and measurement noises are purely additive. Then, the sigma point vector \mathcal{X}_k^w for the process noise is not necessary, and the sigma point vector reduces to $\mathcal{X}_k^a = \mathcal{X}_k^x \equiv \mathcal{X}_k$. Thus, the process noise covariance can be expressed explicitly in the predicted covariance equation as

$$\mathbf{P}_{k+1}^- = \sum_{i=0}^{2n} W_i^{(c)} \left[\mathcal{X}_{i,k+1} - \hat{\mathbf{x}}_{k+1}^- \right] \left[\mathcal{X}_{i,k+1} - \hat{\mathbf{x}}_{k+1}^- \right]^T + \hat{\mathbf{Q}}_k \quad (5.35)$$

Now, the noise adaptation estimator can be directly applied to formulate the adaptive unscented Kalman filter algorithms. The second method is an implicit one where the sigma point sampling procedure is utilized. If the process noise is estimated from the noise adaptation algorithm in Eq. (5.17), the augmented covariance matrix is reconstructed at each update, and thus a new sigma point vector is generated, the *resampling* of the sigma points. Finally, the newly resampled sigma point vector is provided for the prediction and update processes at each measurement update. The overall procedure of the implicit adaptive unscented Kalman filter algorithm is depicted in Fig. 5.3.

5.6 Adaptive Divided Difference Filter

In this section, the proposed noise estimator algorithm is combined with the divided difference filter (DDF)¹² such that the integrated filtering algorithm leads to the *adaptive divided difference filter* (ADDF).

5.6.1 Divided Difference Filter Algorithms

The first-order divided difference filter (DDF1) is illustrated for general discrete-time nonlinear equations in Eq.(5.20) with the assumption that the noise vectors are uncorrelated white Gaussian process with unknown expected means and covariances

$$\begin{aligned} E\{\mathbf{w}_k\} &= \bar{\mathbf{w}}_k, \quad E\left\{[\mathbf{w}_k - \bar{\mathbf{w}}_k][\mathbf{w}_k - \bar{\mathbf{w}}_k]^T\right\} = \mathbf{Q}_k \\ E\{\mathbf{v}_k\} &= \bar{\mathbf{v}}_k, \quad E\left\{[\mathbf{v}_k - \bar{\mathbf{v}}_k][\mathbf{v}_k - \bar{\mathbf{v}}_k]^T\right\} = \mathbf{R}_k \end{aligned} \quad (5.36)$$

First, the square Cholesky factorizations are introduced

$$\mathbf{P}_0 = \mathbf{S}_x \mathbf{S}_x^T, \quad \mathbf{Q}_k = \mathbf{S}_w \mathbf{S}_w^T \quad (5.37)$$

The predicted state vector $\hat{\mathbf{x}}_{k+1}^-$ and predicted state covariance \mathbf{P}_{k+1}^- are determined by

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k, k) \quad (5.38)$$

$$\mathbf{P}_{k+1}^- = \mathbf{S}_x^-(k+1) (\mathbf{S}_x^-(k+1))^T \quad (5.39)$$

where $\mathbf{S}_x^-(k+1) = \left[\mathbf{S}_{x\hat{x}}^{(1)}(k+1) \quad \mathbf{S}_{xw}^{(1)}(k+1) \right]$ and each term is given by

$$\begin{aligned} \mathbf{S}_{x\hat{x}}^{(1)}(k+1) &= \frac{1}{2h} \{ \mathbf{f}_i(\hat{\mathbf{x}}_k + h\mathbf{s}_{x,j}, \bar{\mathbf{w}}_k) - \mathbf{f}_i(\hat{\mathbf{x}}_k - h\mathbf{s}_{x,j}, \bar{\mathbf{w}}_k) \} \\ \mathbf{S}_{xw}^{(1)}(k+1) &= \frac{1}{2h} \{ \mathbf{f}_i(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k + h\mathbf{s}_{w,j}) - \mathbf{f}_i(\hat{\mathbf{x}}_k, \bar{\mathbf{w}}_k - h\mathbf{s}_{w,j}) \} \end{aligned} \quad (5.40)$$

where $\mathbf{s}_{x,j}$ is the j th column of $\mathbf{S}_{\mathbf{x}}$ and $\mathbf{s}_{w,j}$ is the j th column of $\mathbf{S}_{\mathbf{w}}$ obtained from Eq.(5.37), respectively.

Next, the square Cholesky factorizations are performed again

$$\mathbf{P}_{k+1}^- = \mathbf{S}_{\mathbf{x}}^- \mathbf{S}_{\mathbf{x}}^{-T}, \quad \mathbf{R} = \mathbf{S}_{\mathbf{v}} \mathbf{S}_{\mathbf{v}}^T \quad (5.41)$$

The predicted observation vector $\hat{\mathbf{y}}_{k+1}^-$ and its predicted covariance are calculated in a similar fashion

$$\hat{\mathbf{y}}_{k+1}^- = \mathbf{h}(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1}, k+1) \quad (5.42)$$

$$\mathbf{P}_{k+1}^{vv} = \mathbf{S}_v(k+1) \mathbf{S}_v^T(k+1) \quad (5.43)$$

where $\mathbf{S}_v(k+1) = \left[\mathbf{S}_{y\hat{x}}^{(1)}(k+1) \mathbf{S}_{yv}^{(1)}(k+1) \right]$ and each term is given by

$$\begin{aligned} \mathbf{S}_{y\hat{x}}^{(1)}(k+1) &= \frac{1}{2h} \{ \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^- + h\mathbf{s}_{x,j}^-, \bar{\mathbf{v}}_{k+1}) - \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^- - h\mathbf{s}_{x,j}^-, \bar{\mathbf{v}}_{k+1}) \} \\ \mathbf{S}_{yv}^{(1)}(k+1) &= \frac{1}{2h} \{ \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1} + h\mathbf{s}_{v,j}) - \mathbf{h}_i(\hat{\mathbf{x}}_{k+1}^-, \bar{\mathbf{v}}_{k+1} - h\mathbf{s}_{v,j}) \} \end{aligned} \quad (5.44)$$

where $\mathbf{s}_{x,j}^-$ is the j th column of $\mathbf{S}_{\mathbf{x}}^-$ and $\mathbf{s}_{v,j}$ is the j th column of $\mathbf{S}_{\mathbf{v}}$. If the measurement noise vector is simply additive, then the innovation covariance is computed as

$$\mathbf{P}_{k+1}^{vv} = \mathbf{S}_{y\hat{x}}^{(1)}(k+1) \left(\mathbf{S}_{y\hat{x}}^{(1)}(k+1) \right)^T + \mathbf{R}_{k+1} \quad (5.45)$$

Finally, the cross correlation matrix is determined by

$$\mathbf{P}_{k+1}^{xy} = \mathbf{S}_{\mathbf{x}}^-(k+1) \left(\mathbf{S}_{y\hat{x}}^{(1)}(k+1) \right)^T \quad (5.46)$$

Meanwhile, in the update process the filter gain \mathcal{K}_{k+1} , the updated estimate state vector $\hat{\mathbf{x}}_{k+1}^+$, and updated covariance \mathbf{P}_{k+1}^+ can be computed by using the same formulas used in the UKF.

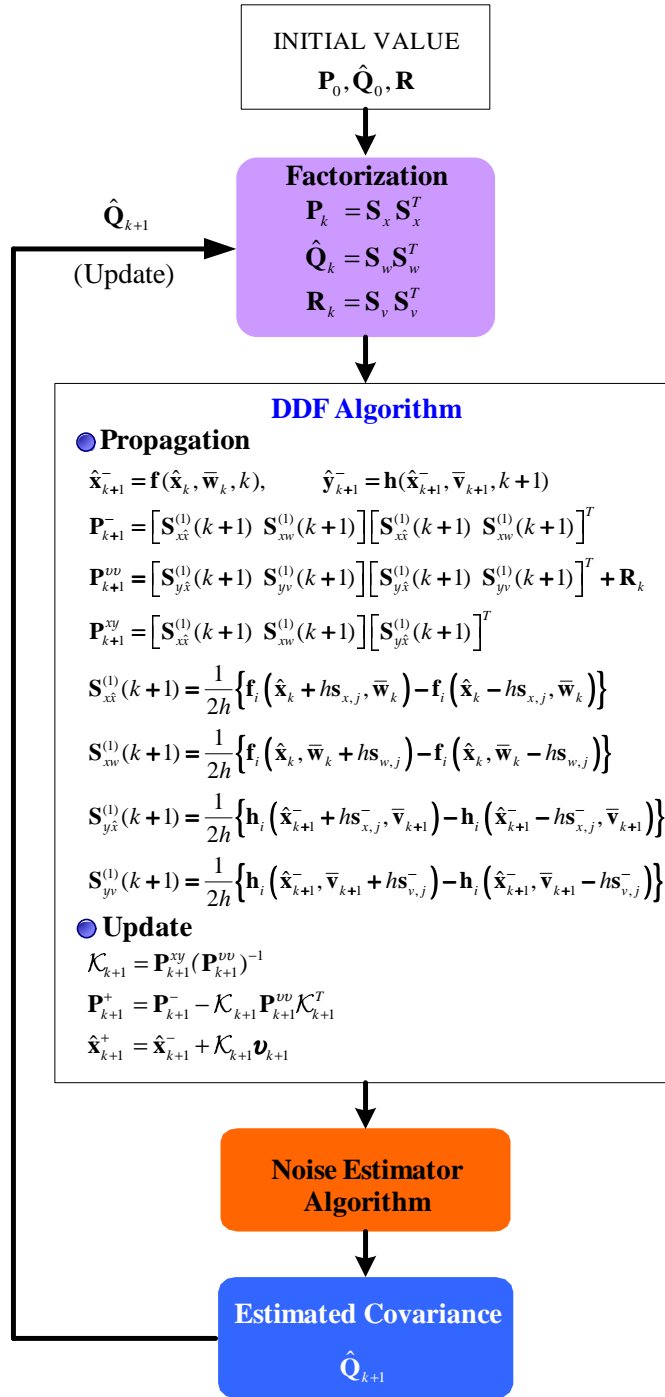


Fig. 5.4 Diagram for Adaptive Divided Difference Filter Algorithm

5.6.2 Adaptive Divided Difference Filtering

For an *adaptive divided difference filter* formulation, the method used to combine the proposed noise estimator with the DDF is to just perform the square Cholesky factorization sequentially at each time when the estimated covariance is updated from the noise adaptation. If the estimated covariance matrix is factorized at each time

$$\hat{\mathbf{Q}}_k = \mathbf{S}_{k,w} \mathbf{S}_{k,w}^T \quad (5.47)$$

then, the factorized value is delivered back to the DDF algorithm leading to an adaptive filtering structure. The overall procedure for the adaptive divided difference filter algorithm is depicted in Figure 5.4.

CHAPTER VI

ADAPTIVE MODEL COMPENSATION

In the previous section, the adaptive filtering methods which compensate for the effects of nonlinearities and modeling errors in the dynamical model were illustrated. In this section as alternative adaptive estimation algorithms, two system model compensation techniques are introduced; one is the dynamic model compensation (DMC)⁹³ and the other is the reduced dynamic compensation (RDC).⁹⁴ Both of these adaptive model compensation methods are investigated in the following sections.

6.1 Dynamic Model Compensation

The method known as dynamic model compensation (DMC) was introduced by Ingram and Myers in the early 1970's.^{93,95} DMC assumes that the spacecraft is subject to accelerations that are not included in the filter's dynamic model, but possess a random element. These are often referred to as "fictitious" accelerations,⁹⁴ but in general, the satellite will be subjected to unmodeled accelerations since any realizable dynamic model must necessarily be a finite approximation of the real dynamic process, and the values of the parameters used in the model are never perfectly known. The development of the DMC algorithm in the next section follows the derivations in Refs. 94 and 95.

6.1.1 Stochastic Acceleration

Assume that the unknown accelerations can be modeled as a first-order, stationary, Gauss-Markov process

$$\dot{\mathbf{w}} + \beta \mathbf{w} = \mathbf{u}(t) \quad (6.1)$$

where $\mathbf{w} \in \Re^{3 \times 1}$ is the vector of accelerations and $\mathbf{u}(t) \in \Re^{3 \times 1}$ is a white, Gaussian process with a mean of zero and a variance of

$$\mathbf{q}_u = \begin{bmatrix} \sigma_{u,x}^2 & 0 & 0 \\ 0 & \sigma_{u,y}^2 & 0 \\ 0 & 0 & \sigma_{u,z}^2 \end{bmatrix} \quad (6.2)$$

and by virtue of the stationary assumption, β is a constant matrix

$$\beta = \begin{bmatrix} \beta_x & 0 & 0 \\ 0 & \beta_y & 0 \\ 0 & 0 & \beta_z \end{bmatrix} = \begin{bmatrix} \frac{1}{\tau_x} & 0 & 0 \\ 0 & \frac{1}{\tau_y} & 0 \\ 0 & 0 & \frac{1}{\tau_z} \end{bmatrix} \quad (6.3)$$

where τ_i is the correlation time for a given axis. Eq. (6.1) for a scalar case (one axis) is a first-order linear differential equation, and its solution is

$$\begin{aligned} w(t) &= w_0 e^{-\beta(t-t_0)} + e^{-\beta t} \int_{t_0}^t e^{\beta T} u(T) dT \\ &= w_0 e^{-\beta(t-t_0)} + \int_{t_0}^t e^{-\beta(t-T)} u(T) dT \\ &= w_0 e^{-\beta(t-t_0)} + L(t) \end{aligned} \quad (6.4)$$

The stochastic integral $L(t)$ can't be solved by usual methods, but it can be evaluated statistically. The mean of $L(t)$ is zero since the mean of $u(t)$ is zero and its variance is obtained by

$$E \{ L^2(t) \} = E \left\{ \int_{t_0}^t \int_{t_0}^T e^{-\beta(2t-T-s)} u(T) u(s) dT ds \right\} \quad (6.5)$$

Expectation and integration are commutative linear operators,⁹³ so the expectation can be taken inside the integrals

$$E \{ L^2(t) \} = \int_{t_0}^t \int_{t_0}^T e^{-\beta(2t-T-s)} E \{ u(T) u(s) \} dT ds \quad (6.6)$$

Since $u(t)$ is uncorrelated in time

$$E \{u(T)u(s)\} = \sigma_u^2 \delta(T - s) \quad (6.7)$$

where σ_u^2 is the variance of $u(t)$, and $\delta(T - s)$ is the *Dirac delta* function where $\delta(T - s) = 0$ everywhere except at $T = s$. Thus, the integral of Eq. (6.6) is non-zero only at $T = s$, and it reduces to

$$E \{L^2(t)\} = \int_{t_0}^t e^{-2\beta(t-T)} \sigma_u^2 dT \quad (6.8)$$

If $u(t)$ is stationary over the interval, σ_u is constant and the result becomes

$$\begin{aligned} E \{L^2(t)\} &= \frac{\sigma_u^2}{2\beta} (1 - e^{-2\beta(t-t_0)}) \\ &= \frac{\tau\sigma_u^2}{2} (1 - e^{-2(t-t_0)/\tau}) \end{aligned} \quad (6.9)$$

where τ is the correlation time.

$L(t)$ is assumed to be a Gaussian process. The probability density function of a Gaussian process is completely defined by its mean and variance. Therefore, if a process can be found with the same mean and variance as $L(t)$, it will be an equivalent process. A discrete process, applicable to the discrete state estimation filter, is given by⁹⁵

$$L(t) \approx L_k = u_k \sqrt{\frac{\tau\sigma_u^2}{2} (1 - e^{-2(t-t_0)/\tau})} \quad (6.10)$$

where u_k is a discrete, Gaussian random sequence with mean and variance. The mean and variance are given by

$$E \{u_k\} = 0, \quad E \{u_k u_j\} = \delta_{k,j} \quad (6.11)$$

where $\delta_{k,j}$ is the *Kronecker delta* function. If the time interval $\Delta t = t - t_0$ is short enough such that $u(t)$ can reasonably be assumed constant over the interval, L_k will serve as a good approximation of $L(t)$. Finally, the stochastic acceleration including

both deterministic and purely random parts is given by

$$w(t) = w_0 e^{-(t-t_0)/\tau} + u_k \sqrt{\frac{\tau \sigma_u^2}{2}} (1 - e^{-2(t-t_0)/\tau}) \quad (6.12)$$

The parameters σ_u and τ determine the characteristics of the acceleration function. As $\tau \rightarrow \infty$, $w(t)$ becomes a constant, whereas if $\tau \rightarrow 0$, $w(t)$ becomes a purely random process (white noise).

6.1.2 Filtering Model

6.1.2.1 State Transition Matrix

Assume that the true real-world filter dynamic model has the following form

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a}_p(\mathbf{r}, \mathbf{v}, t) + \mathbf{w}(t) \\ \dot{\mathbf{w}} &= -\beta \mathbf{w} + \mathbf{u}(t) \end{aligned} \quad (6.13)$$

where $\mathbf{r} = [x, y, z]^T$ and $\mathbf{v} = [\dot{x}, \dot{y}, \dot{z}]^T$ are the position and velocity vectors, respectively, and \mathbf{a}_p is an acceleration function due to the two-body attraction, drag perturbation, and harmonic geopotential perturbing acceleration up to J_2 . $\mathbf{w}(t) = [w_x, w_y, w_z]^T$ represents three unknown stochastic acceleration terms. $\mathbf{u}(t)$ is a white, Gaussian process noise which presents the effects of model errors with the properties

$$E \{ \mathbf{u}(t) \} = 0, \quad E \{ \mathbf{u}(t) \mathbf{u}^T(s) \} = \mathbf{Q}_u(t) \delta(t - s) \quad (6.14)$$

Then, the vector/matrix equation in terms of the force vector \mathbf{f} , which is the time derivative of the state vector, can be written as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{u}(t) \quad (6.15)$$

where \mathbf{x} is the basic state vector for the filtering algorithm defined by

$$\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, w_x, w_y, w_z]^T \in \Re^{9 \times 1} \quad (6.16)$$

Now, the nominal or filter dynamic model in a vector/matrix form is represented by

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, t) \quad (6.17)$$

The expression for the state sensitivity matrix \mathbf{F} , defined as the partial derivative of the force vector \mathbf{f} with respect to the nominal state vector $\hat{\mathbf{x}}$, is calculated from

$$\mathbf{F} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right]_{\mathbf{x}=\hat{\mathbf{x}}} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} & \frac{\partial \dot{x}}{\partial z} & \frac{\partial \dot{x}}{\partial \dot{x}} & \frac{\partial \dot{x}}{\partial \dot{y}} & \frac{\partial \dot{x}}{\partial \dot{z}} & \frac{\partial \dot{x}}{\partial w_x} & \frac{\partial \dot{x}}{\partial w_y} & \frac{\partial \dot{x}}{\partial w_z} \\ \frac{\partial \dot{y}}{\partial x} & \dots & & & & & & & \\ \frac{\partial \dot{z}}{\partial x} & \dots & & & & & & & \\ \frac{\partial \ddot{x}}{\partial x} & \dots & & & & & & & \\ \frac{\partial \ddot{y}}{\partial x} & \dots & & & & & & & \\ \frac{\partial \ddot{z}}{\partial x} & \dots & & & & & & & \\ \frac{\partial \dot{w}_x}{\partial x} & \dots & & & & & & & \\ \frac{\partial \dot{w}_y}{\partial x} & \dots & & & & & & & \\ \frac{\partial \dot{w}_z}{\partial x} & \frac{\partial \dot{w}_z}{\partial y} & \frac{\partial \dot{w}_z}{\partial z} & \frac{\partial \dot{w}_z}{\partial \dot{x}} & \frac{\partial \dot{w}_z}{\partial \dot{y}} & \frac{\partial \dot{w}_z}{\partial \dot{z}} & \frac{\partial \dot{w}_z}{\partial w_x} & \frac{\partial \dot{w}_z}{\partial w_y} & \frac{\partial \dot{w}_z}{\partial w_z} \end{bmatrix}_{\mathbf{x}=\hat{\mathbf{x}}} \quad (6.18)$$

where the partial derivative components with respect to the position and velocity components are listed in Appendix A. The state transition matrix Φ is obtained by integrating the following matrix differential equation

$$\dot{\Phi}(t_k) = \mathbf{F}(t_k, t_0) \Phi(t_0) \quad (6.19)$$

with the initial condition, $\Phi(t_0) = \mathbf{I}$. The system of coupled, linear differential equations represented by Eq. (6.19) is integrated by a fourth-order matrix Runge-Kutta method.

The deterministic parts of the stochastic accelerations $\mathbf{w}(t)$ contribute terms to the dynamic expressions for position and velocity. Taking the x -axis as an example, the velocity component is obtained by integrating the deterministic part of Eq. (6.12) and the acceleration terms in the filter dynamic model

$$\dot{x}(t) = \dot{x}_{filter}(t) + \tau w_{x_0} (1 - e^{-(t-t_0)/\tau}) \quad (6.20)$$

Integrating the above results produces the position solution

$$x(t) = x_{filter}(t) + \tau^2 w_{x_0} (e^{-(t-t_0)/\tau} - 1) + \tau w_{x_0} (t - t_0) \quad (6.21)$$

where the subscript “*filter*” denotes the position and velocity obtained by integrating the dynamic acceleration model such as gravity plus drag.

The general definition of a state transition matrix is defined as

$$\Phi(t_k, t_0) \equiv \frac{\partial \mathbf{x}(t_k)}{\partial \mathbf{x}(t_0)} \quad (6.22)$$

Then, the state transition matrix $\Phi(t_k, t_0) \in \mathbb{R}^{9 \times 9}$ with respect to the augmented state vector $\hat{\mathbf{x}}$ takes the form

$$\Phi(t_k, t_0) = \begin{bmatrix} [\Phi_{filter}]_{6 \times 6} & [\Phi_w]_{6 \times 3} \\ [\mathbf{0}]_{3 \times 6} & [\mathbf{M}]_{3 \times 3} \end{bmatrix} \quad (6.23)$$

where $[\Phi_{filter}]_{6 \times 6}$ is the state transition matrix associated with the 6 states without the DMC, $[\Phi_w]_{6 \times 3}$ is the transition matrix of the 6 basic states with respect to the stochastic accelerations, and $[\mathbf{M}]_{3 \times 3}$ is the state transition matrix of the stochastic acceleration with respect to themselves. Then, each matrix element of Φ_w and \mathbf{M} can be found analytically by taking the partial derivatives with respect to the

corresponding state elements

$$\mathbf{\Phi}_w = \begin{bmatrix} \phi_{wp} & 0 & 0 \\ 0 & \phi_{wp} & 0 \\ 0 & 0 & \phi_{wp} \\ \phi_{wv} & 0 & 0 \\ 0 & \phi_{wv} & 0 \\ 0 & 0 & \phi_{wv} \end{bmatrix} \quad (6.24)$$

$$\mathbf{M} = \begin{bmatrix} e^{-(t-t_0)/\tau} & 0 & 0 \\ 0 & e^{-(t-t_0)/\tau} & 0 \\ 0 & 0 & e^{-(t-t_0)/\tau} \end{bmatrix} \quad (6.25)$$

where

$$\phi_{wp} = \tau^2 (e^{-(t-t_0)/\tau} - 1) + \tau (t - t_0) \quad (6.26)$$

$$\phi_{wv} = \tau (1 - e^{-(t-t_0)/\tau}) \quad (6.27)$$

Now, suppose that the augmented state vector \mathbf{x} includes the time correlation term τ such that

$$\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, w_x, w_y, w_z, \tau]^T \in \Re^{10 \times 1} \quad (6.28)$$

Then, the state transition matrix $\mathbf{\Phi}(t_k, t_0) \in \Re^{10 \times 10}$ has the following expression

$$\mathbf{\Phi}(t_k, t_0) = \begin{bmatrix} [\mathbf{\Phi}_{filter}]_{6 \times 6} & [\mathbf{\Phi}_w]_{6 \times 3} & [\mathbf{\Phi}_\tau]_{6 \times 1} \\ [\mathbf{0}]_{3 \times 6} & [\mathbf{M}]_{3 \times 3} & [\mathbf{N}]_{3 \times 1} \\ [\mathbf{0}]_{1 \times 6} & [\mathbf{0}]_{1 \times 3} & 1_{1 \times 1} \end{bmatrix} \quad (6.29)$$

where $[\mathbf{\Phi}_\tau]_{6 \times 1}$ is the transition matrix of the 6 basic states with respect to the correlation time, and $[\mathbf{N}]_{3 \times 1}$ is the transition matrix of the stochastic accelerations with

respect to the correlation time. Then, each matrix element of Φ_τ and \mathbf{M} can be found analytically

$$\Phi_\tau = \begin{bmatrix} 2\tau w_{x_0} (e^{-(t-t_0)/\tau} - 1) + w_{x_0} (t - t_0) (e^{-(t-t_0)/\tau} + 1) \\ 2\tau w_{y_0} (e^{-(t-t_0)/\tau} - 1) + w_{y_0} (t - t_0) (e^{-(t-t_0)/\tau} + 1) \\ 2\tau w_{z_0} (e^{-(t-t_0)/\tau} - 1) + w_{z_0} (t - t_0) (e^{-(t-t_0)/\tau} + 1) \\ w_{x_0} (e^{-(t-t_0)/\tau} - 1) - \frac{w_{x_0}}{\tau} (t - t_0) e^{-(t-t_0)/\tau} \\ w_{y_0} (e^{-(t-t_0)/\tau} - 1) - \frac{w_{y_0}}{\tau} (t - t_0) e^{-(t-t_0)/\tau} \\ w_{z_0} (e^{-(t-t_0)/\tau} - 1) - \frac{w_{z_0}}{\tau} (t - t_0) e^{-(t-t_0)/\tau} \end{bmatrix} \quad (6.30)$$

$$\mathbf{N} = \begin{bmatrix} \frac{w_{x_0}}{\tau^2} (t - t_0) e^{-(t-t_0)/\tau} \\ \frac{w_{y_0}}{\tau^2} (t - t_0) e^{-(t-t_0)/\tau} \\ \frac{w_{z_0}}{\tau^2} (t - t_0) e^{-(t-t_0)/\tau} \end{bmatrix} \quad (6.31)$$

6.1.2.2 Process Noise Matrix

(1) Position, Velocity, and Acceleration Terms

Now assume that a reference deterministic trajectory $\hat{\mathbf{x}}^-(t_k)$ is obtained from the propagated state estimate in the filtering propagation procedure and the deviation from the reference is defined by

$$\delta \mathbf{x}(t_k) = \mathbf{x}(t_k) - \hat{\mathbf{x}}^-(t_k) \quad (6.32)$$

where $\mathbf{x}(t_k)$ is the true state at time t_k . Then, the stochastic discrete-time linear dynamics from a linear error theory forced by process noise $\boldsymbol{\eta}(t_k)$ can be written by

$$\delta \mathbf{x}(t_k) = \Phi(t_k, t_0) \delta \mathbf{x}(t_0) + \boldsymbol{\eta}(t_k) \quad (6.33)$$

where

$$\delta \mathbf{x}(t_0) = \mathbf{x}(t_0) - \hat{\mathbf{x}}^+(t_0) \quad (6.34)$$

$\mathbf{x}(t_0)$ is the true state at time t_0 , and $\hat{\mathbf{x}}^+(t_0)$ is the estimate of the true state at time t_0 , conditioned on observations through t_0 .

The definition of the propagated covariance matrix $\mathbf{P}(t_k)$ at the current time t_k conditioned on observations through t_0 is defined by

$$\mathbf{P}(t_k) \triangleq E \left\{ [\mathbf{x}(t_k) - \hat{\mathbf{x}}^-(t_k)] [\mathbf{x}(t_k) - \hat{\mathbf{x}}^-(t_k)]^T \right\} = E \left\{ \delta \mathbf{x}(t_k) (\delta \mathbf{x}(t_k))^T \right\} \quad (6.35)$$

Substituting Eq. (6.33) into Eq. (6.35) and expanding the products results in

$$\begin{aligned} \mathbf{P}(t_k) = & \Phi(t_k, t_0) E \left\{ \delta \mathbf{x}(t_0) \delta \mathbf{x}^T(t_0) \right\} \Phi^T(t_k, t_0) + E \left\{ \boldsymbol{\eta}(t_k) \boldsymbol{\eta}^T(t_k) \right\} + \\ & \Phi(t_k, t_0) E \left\{ \delta \mathbf{x}(t_0) \boldsymbol{\eta}^T(t_k) \right\} + E \left\{ \boldsymbol{\eta}(t_k) \delta \mathbf{x}^T(t_0) \right\} \Phi^T(t_k, t_0) \end{aligned} \quad (6.36)$$

Since the $\delta \mathbf{x}(t_k)$ and $\boldsymbol{\eta}(t_k)$ are assumed to be uncorrelated

$$E \left\{ \delta \mathbf{x}(t_0) \boldsymbol{\eta}^T(t_k) \right\} = E \left\{ \boldsymbol{\eta}(t_k) \delta \mathbf{x}^T(t_0) \right\} = 0 \quad (6.37)$$

Finally, the covariance equation becomes

$$\mathbf{P}(t_k) = \Phi(t_k, t_0) \mathbf{P}(t_0) \Phi^T(t_k, t_0) + \mathbf{Q}(t_k) \quad (6.38)$$

where

$$\mathbf{P}(t_0) \equiv E \left\{ \delta \mathbf{x}(t_0) \delta \mathbf{x}^T(t_0) \right\}, \quad \mathbf{Q}(t_k) \equiv E \left\{ \boldsymbol{\eta}(t_k) \boldsymbol{\eta}^T(t_k) \right\} \quad (6.39)$$

Note that the component of $\boldsymbol{\eta}(t_k)$ due to the stochastic acceleration function has already been determined in Eq. (6.10). Translating this into vector form gives

$$\mathbf{L}_k = \begin{bmatrix} u_{x,k} \sqrt{\frac{\tau \sigma_{u,x}^2}{2} (1 - e^{-2(t-t_0)/\tau})} \\ u_{y,k} \sqrt{\frac{\tau \sigma_{u,y}^2}{2} (1 - e^{-2(t-t_0)/\tau})} \\ u_{z,k} \sqrt{\frac{\tau \sigma_{u,z}^2}{2} (1 - e^{-2(t-t_0)/\tau})} \end{bmatrix} \quad (6.40)$$

The position and velocity components of $\boldsymbol{\eta}(t_k)$ can be found by using the integral

expressions for position and velocity⁹⁵

$$\mathbf{r}(t) = \mathbf{r}(t_0) + \mathbf{v}(t_0)(t - t_0) + \int_{t_0}^t \mathbf{a}(T) (t - T) dT \quad (6.41)$$

$$\mathbf{v}(t) = \mathbf{v}(t_0) + \int_{t_0}^t \mathbf{a}(T) dT \quad (6.42)$$

The stochastic acceleration function $\mathbf{w}(t)$ consists of a deterministic component which is modeled in the filter, and a random component \mathbf{L}_k which is not modeled in the filter but contributes to $\boldsymbol{\eta}(t_k)$. The total acceleration can be expressed by

$$\mathbf{a}(t) = \mathbf{a}_m(t) + \mathbf{L}_k(t) \quad (6.43)$$

where \mathbf{a}_m are the accelerations in the filter model consisting of the perturbing acceleration \mathbf{a}_p and the deterministic part of the stochastic acceleration solution $\mathbf{w}(t)$.

Substituting this into the integral position and velocity equations gives

$$\mathbf{r}(t) = \mathbf{r}(t_0) + \mathbf{v}(t_0)(t - t_0) + \int_{t_0}^t \mathbf{a}_m(T) (t - T) dT + \int_{t_0}^t \mathbf{L}_k(T) (t - T) dT \quad (6.44)$$

$$\mathbf{v}(t) = \mathbf{v}(t_0) + \int_{t_0}^t \mathbf{a}_m(T) dT + \int_{t_0}^t \mathbf{L}_k(T) dT \quad (6.45)$$

The position and velocity components of $\boldsymbol{\eta}(t_k)$ are identified as the stochastic integral terms in these two equations. The total expression for $\boldsymbol{\eta}(t_k)$ is obtained by

$$\boldsymbol{\eta}(t_k) = \begin{bmatrix} \int_{t_0}^t \mathbf{L}_k(T) (t - T) dT \\ \int_{t_0}^t \mathbf{L}_k(T) dT \\ \mathbf{L}_k(T) \end{bmatrix} \quad (6.46)$$

Note that each of the components in these expression is a 3×1 vector, so the dimension of $\boldsymbol{\eta}(t_k)$ is 9×1 . If \mathbf{L}_k is constant over the integral $\Delta t = t - t_0$, the position component

becomes

$$\int_{t_0}^t \mathbf{L}_k(T)(t-T)dT = \mathbf{L}_k \left[tT - \frac{T^2}{2} \right]_{t_0}^t = \mathbf{L}_k \left[\left(t^2 - \frac{t^2}{2} \right) - \left(tt_0 - \frac{t_0^2}{2} \right) \right] \quad (6.47)$$

$$= \mathbf{L}_k \left[\frac{t^2}{2} - tt_0 + \frac{t_0^2}{2} \right] = \mathbf{L}_k \left[\frac{(t-t_0)^2}{2} \right] \quad (6.48)$$

$$= \mathbf{L}_k \frac{\Delta t^2}{2} \quad (6.49)$$

Thus, the expression of $\boldsymbol{\eta}(t_k)$ is approximated by

$$\boldsymbol{\eta}(t_k) = \begin{bmatrix} \Delta t^2/2 \mathbf{L}_k \\ \Delta t \mathbf{L}_k \\ \mathbf{L}_k \end{bmatrix} \quad (6.50)$$

Note that the variance of any particular component of \mathbf{L}_k is given by Eq. (6.9) and the mean of \mathbf{L}_k is zero. Assuming that the components of $\mathbf{u}(t)$ are uncorrelated and all have equal statistics, the variance of \mathbf{L}_k is defined as $E \{ \mathbf{L}_k \mathbf{L}_k^T \} \triangleq [\boldsymbol{\Lambda}]$ and can be written as

$$[\boldsymbol{\Lambda}] = \begin{bmatrix} \frac{\tau \sigma_u^2}{2} (1 - e^{-2(t-t_0)/\tau}) & 0 & 0 \\ 0 & \frac{\tau \sigma_u^2}{2} (1 - e^{-2(t-t_0)/\tau}) & 0 \\ 0 & 0 & \frac{\tau \sigma_u^2}{2} (1 - e^{-2(t-t_0)/\tau}) \end{bmatrix} \quad (6.51)$$

Finally, the discrete-time process noise covariance matrix \mathbf{Q}_k for position, velocity, and stochastic acceleration can be constructed

$$\mathbf{Q}_k = E \{ \boldsymbol{\eta}(t_k) \boldsymbol{\eta}(t_k)^T \} = \begin{bmatrix} \left[\frac{\Delta t^4}{4} \boldsymbol{\Lambda} \right] & \left[\frac{\Delta t^3}{2} \boldsymbol{\Lambda} \right] & \left[\frac{\Delta t^2}{2} \boldsymbol{\Lambda} \right] \\ \left[\frac{\Delta t^3}{2} \boldsymbol{\Lambda} \right] & [\Delta t^2 \boldsymbol{\Lambda}] & [\Delta t \boldsymbol{\Lambda}] \\ \left[\frac{\Delta t^2}{2} \boldsymbol{\Lambda} \right] & [\Delta t \boldsymbol{\Lambda}] & [\boldsymbol{\Lambda}] \end{bmatrix} \quad (6.52)$$

The presence of the correlation time τ , which is estimated in the state vector, plays a role of adaptation in dynamic model compensation. Thus, the dynamic model

compensation is interpreted as an adaptive filtering method.

(2) *Correlation Time Term*

The correlation time is modeled as a constant plus a random walk term so that its time derivative is simply equal to a random noise process

$$\dot{\tau} = u_{\tau}(t) \quad (6.53)$$

where $u_{\tau}(t)$ is a zero mean, uncorrelated, stationary, Gaussian process with variance σ_{τ}^2 . The solution is given by

$$\tau(t) = \tau_0 + \int_{t_0}^t u_{\tau}(T) dT \quad (6.54)$$

The stochastic integral is defined

$$M(t) \equiv \int_{t_0}^t u_{\tau}(T) dT \quad (6.55)$$

where $M(t)$ has a zero mean since $u_{\tau}(t)$ is zero mean, and its variance is computed by

$$\begin{aligned} E \{M^2(t)\} &= E \left\{ \int_{t_0}^t \int_{t_0}^t u_{\tau}(T) u_{\tau}(s) dT ds \right\} \\ &= \int_{t_0}^t \int_{t_0}^t E \{u_{\tau}(T) u_{\tau}(s)\} dT ds \end{aligned} \quad (6.56)$$

Since $u_{\tau}(t)$ is uncorrelated

$$E \{u_{\tau}(T) u_{\tau}(s)\} = \sigma_{\tau}^2 \delta(T - s) \quad (6.57)$$

where $\delta(T - s)$ is the *Dirac delta* function. The integral reduces to

$$E \{M^2(t)\} = \int_{t_0}^t \sigma_{\tau}^2 dT \quad (6.58)$$

Since $u_\tau(t)$ is a stationary process, σ_τ is constant, and thus the integral becomes

$$E \{M^2(t)\} = \sigma_\tau^2(t - t_0) \quad (6.59)$$

A discrete process with the same mean and variance is given by

$$M_k = u_{\tau,k} \sigma_\tau \sqrt{(t - t_0)} \quad (6.60)$$

where $u_{\tau,k}$ is a discrete Gaussian sequence with

$$E \{u_{\tau,k}\} = 0, \quad E \{u_{\tau,k} u_{\tau,j}\} = \delta_{k,j} \quad (6.61)$$

$\delta_{k,j}$ is the *Kronecker delta* function. The value of $u_{\tau,k}$ is assumed constant over the interval $\Delta t = t - t_0$, which is short enough for this to be a reasonable assumption, M_k will be a good approximation of $M(t)$. The stochastic model of the correlation time is calculated by

$$\tau(t) = \tau_0 + u_{\tau,k} \sigma_\tau \sqrt{(t - t_0)} \quad (6.62)$$

The contribution to the process noise matrix is given by the variance

$$q_\tau = E \{M_k^2\} = \sigma_\tau^2(t - t_0) \quad (6.63)$$

(3) *Complete Form of the Process Noise Matrix*

The discrete-time process noise covariance matrix $\mathbf{Q}_k \in \Re^{10 \times 10}$ for the process noise terms due to position, velocity, stochastic acceleration, and correlation time is

constructed by

$$\mathbf{Q}_k = \begin{bmatrix} \left[\frac{\Delta t^4}{4} \mathbf{\Lambda} \right]_{3 \times 3} & \left[\frac{\Delta t^3}{2} \mathbf{\Lambda} \right]_{3 \times 3} & \left[\frac{\Delta t^2}{2} \mathbf{\Lambda} \right]_{3 \times 3} & [\mathbf{0}]_{3 \times 1} \\ \left[\frac{\Delta t^3}{2} \mathbf{\Lambda} \right]_{3 \times 3} & [\Delta t^2 \mathbf{\Lambda}]_{3 \times 3} & [\Delta t \mathbf{\Lambda}]_{3 \times 3} & [\mathbf{0}]_{3 \times 1} \\ \left[\frac{\Delta t^2}{2} \mathbf{\Lambda} \right]_{3 \times 3} & [\Delta t \mathbf{\Lambda}]_{3 \times 3} & [\mathbf{\Lambda}]_{3 \times 3} & [\mathbf{0}]_{3 \times 1} \\ [\mathbf{0}]_{1 \times 3} & [\mathbf{0}]_{1 \times 3} & [\mathbf{0}]_{1 \times 3} & [q_\tau]_{1 \times 1} \end{bmatrix} \quad (6.64)$$

where $\Delta t = t - t_0$

6.2 Reduced Dynamic Tracking

Several works have employed the model compensation method known as reduced dynamic tracking (RDT), which is similar to that of the DMC method.^{94,96} Unfortunately, only a summary of the RDT method is provided, and no specific explanation and derivation steps are provided. Thus, in this section, we first derive the RDT formulation and compare with the DMC method for the computational analysis in detail.

6.2.1 Modified Stochastic Acceleration

Suppose that the unknown accelerations can be modeled as a modified, first-order, stationary, Gauss-Markov process

$$\dot{\mathbf{w}} + \beta \mathbf{w} = \sqrt{2\beta} \mathbf{u}(t) \quad (6.65)$$

where $\mathbf{w} \in \Re^{3 \times 1}$ is the vector of accelerations and $\mathbf{u}(t) \in \Re^{3 \times 1}$ is a white, Gaussian process with a mean of zero and a variance of

$$\mathbf{q}_u = \begin{bmatrix} \sigma_{u,x}^2 & 0 & 0 \\ 0 & \sigma_{u,y}^2 & 0 \\ 0 & 0 & \sigma_{u,z}^2 \end{bmatrix} \quad (6.66)$$

and β is assumed to have a constant matrix

$$\beta = \begin{bmatrix} \beta_x & 0 & 0 \\ 0 & \beta_y & 0 \\ 0 & 0 & \beta_z \end{bmatrix} = \begin{bmatrix} \frac{1}{\tau_x} & 0 & 0 \\ 0 & \frac{1}{\tau_y} & 0 \\ 0 & 0 & \frac{1}{\tau_z} \end{bmatrix} \quad (6.67)$$

where τ_i is the correlation time for a given axis. For a scalar case (one axis), a first-order linear differential equation and its solution is calculated by

$$\begin{aligned} w(t) &= w_0 e^{-\beta(t-t_0)} + e^{-\beta t} \int_{t_0}^t e^{\beta T} \sqrt{2\beta} u(T) dT \\ &= w_0 e^{-\beta(t-t_0)} + \int_{t_0}^t e^{-\beta(t-T)} \sqrt{2\beta} u(T) dT \\ &= w_0 e^{-\beta(t-t_0)} + L(t) \end{aligned} \quad (6.68)$$

The statistical mean of $L(t)$ is zero since the mean of $u(t)$ is zero and its variance is obtained by

$$E \{L^2(t)\} = E \left\{ \int_{t_0}^t \int_{t_0}^T e^{-\beta(2t-T-s)} 2\beta u(T) u(s) dT ds \right\} \quad (6.69)$$

The expectation can be taken inside the integrals and becomes

$$E \{L^2(t)\} = \int_{t_0}^t \int_{t_0}^T e^{-\beta(2t-T-s)} 2\beta E \{u(T)u(s)\} dT ds \quad (6.70)$$

Since $u(t)$ is uncorrelated in time

$$E \{u(T)u(s)\} = \sigma_u^2 \delta(T-s) \quad (6.71)$$

where σ_u^2 is the variance of $u(t)$, and $\delta(T - s)$ is the *Dirac delta* function where $\delta(T - s) = 0$ everywhere except at $T = s$. Thus, the integral of Eq. (6.70) is non-zero only at $T = s$, and it reduces to

$$E \{L^2(t)\} = \int_{t_0}^t e^{-2\beta(t-T)} 2\beta \sigma_u^2 dT \quad (6.72)$$

If $u(t)$ is stationary over the interval, σ_u is constant and the result becomes

$$\begin{aligned} E \{L^2(t)\} &= \sigma_u^2 (1 - e^{-2\beta(t-t_0)}) \\ &= \sigma_u^2 (1 - e^{-2(t-t_0)/\tau}) \end{aligned} \quad (6.73)$$

where τ is the correlation time.

If $L(t)$ is assumed to be a Gaussian process, then the process can be approximated in terms of the same mean and variance as $L(t)$ leading to a discrete process L_k expressed by

$$L(t) \approx L_k = u_k \sqrt{\sigma_u^2 (1 - e^{-2(t-t_0)/\tau})} \quad (6.74)$$

where u_k is a discrete, Gaussian random sequence with mean and variance. The mean and variance are given by

$$E \{u_k\} = 0, \quad E \{u_k u_j\} = \delta_{k,j} \quad (6.75)$$

where $\delta_{k,j}$ is the *Kronecker delta* function. If the time interval $\Delta t = t - t_0$ is short enough such that $u(t)$ can reasonably be assumed constant over the interval, L_k will be a good approximation of $L(t)$. Finally, the stochastic acceleration $w(t)$ including both deterministic and purely random parts is given by

$$w(t) = w_0 e^{-(t-t_0)/\tau} + u_k \sqrt{\sigma_u^2 (1 - e^{-2(t-t_0)/\tau})} \quad (6.76)$$

The parameters σ_u and τ determine the characteristics of the acceleration function.

Note that RDT can be characterized as a subset of DMC with a simplified,

modified process noise matrix. The significant difference between the DMC and RDT lies in the factor τ . First, consider the discrete-time process noise L_k in RDT case, as $\tau \rightarrow \infty$ for a finite σ_i and Δt , L_k becomes zero, whereas if $\tau \rightarrow 0$ for a finite σ_i and Δt , L_k reduces to $u_k \sigma_i$ which is a purely random process. For the DMC case, as $\tau \rightarrow \infty$ for a finite σ_i and Δt , L_k goes to $u_k \sqrt{\sigma_i^2 \Delta t}$. As τ increases and L_k approaches this limiting value, $w(t)$ loses its dependence on τ . Meanwhile, the deterministic parts of the acceleration functions are the same for both the DMC and RDT methods. Hence, in the DMC case, as the correlation time become infinite, the acceleration takes a random walk characteristic.

6.2.2 Filtering Model

6.2.2.1 State Transition Matrix

In this RDC method, it is assumed that the true real-world filter dynamic model takes the following form

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a}_p(\mathbf{r}, \mathbf{v}, t) \\ \dot{\mathbf{w}} &= -\beta \mathbf{w} + \sqrt{2\beta} \mathbf{u}(t)\end{aligned}\tag{6.77}$$

where $\mathbf{r} = [x, y, z]^T$ and $\mathbf{v} = [\dot{x}, \dot{y}, \dot{z}]^T$ are the position and velocity vectors, respectively, and \mathbf{a}_p is an acceleration function due to the two-body attraction, drag perturbation, and harmonic geopotential perturbing acceleration up to J_2 . $\mathbf{w}(t) = [w_x, w_y, w_z]^T$ represents three unknown stochastic acceleration terms. $\mathbf{u}(t)$ represents the effects of the model errors and is a white, Gaussian process noise with the properties

$$E\{\mathbf{u}(t)\} = 0, \quad E\{\mathbf{u}(t)\mathbf{u}^T(s)\} = \mathbf{Q}_u(t)\delta(t-s)\tag{6.78}$$

Then, the vector/matrix equation in terms of the acceleration vector \mathbf{f} , which is

the time derivative of the state vector, can be expressed by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \boldsymbol{\eta}(t) \quad (6.79)$$

where the process noise $\boldsymbol{\eta}(t)$ is equal to $\sqrt{2\boldsymbol{\beta}}\mathbf{u}(t)$, and \mathbf{x} is the basic state vector for the filtering algorithm defined by

$$\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, w_x, w_y, w_z]^T \in \Re^{9 \times 1} \quad (6.80)$$

The nominal or filter dynamic model in a vector/matrix form is represented by

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, t) \quad (6.81)$$

The state transition matrix is calculated by integrating the matrix differential equation

$$\dot{\boldsymbol{\Phi}}(t_k) = \mathbf{F}(t_k, t_0)\boldsymbol{\Phi}(t_0), \quad \boldsymbol{\Phi}(t_0) = \mathbf{I} \quad (6.82)$$

where the state sensitivity matrix, \mathbf{F} , was obtained in Eq. (6.18).

According to the general definition of a state transition matrix defined in Eq. (6.22), the state transition matrix $\boldsymbol{\Phi}(t_k, t_0) \in \Re^{9 \times 9}$ has the form

$$\boldsymbol{\Phi}(t_k, t_0) = \begin{bmatrix} [\boldsymbol{\Phi}_{filter}]_{6 \times 6} & [\boldsymbol{\Phi}_{wp}]_{6 \times 3} \\ [\mathbf{0}]_{3 \times 6} & [\mathbf{M}]_{3 \times 3} \end{bmatrix} \quad (6.83)$$

where $[\boldsymbol{\Phi}_{filter}]_{6 \times 6}$ is the state transition matrix associated with the 6 states without the DMC, $[\boldsymbol{\Phi}_{wp}]_{6 \times 3}$ is the transition matrix of the six basic states with respect to the stochastic accelerations, and $[\mathbf{M}]_{3 \times 3}$ is the state transition matrix of the stochastic acceleration with respect to themselves and its diagonal matrix component with the i th element is given by

$$m_i = e^{-(t-t_0)/\tau_i}, \quad i = 1, 2, 3 \quad (6.84)$$

Note that the state transition matrix given by Eq. (6.83) is analogous to that in Eq. (6.29) except for the exclusion of the terms associated with the correlation time τ in the state vector.

6.2.2.2 Process Noise Matrix

(1) *Position, Velocity, and Acceleration Terms*

The stochastic discrete-time linear dynamics from a linear error theory forced by process noise $\boldsymbol{\eta}(t_k)$ can be written by

$$\delta \mathbf{x}(t_k) = \boldsymbol{\Phi}(t_k, t_0) \delta \mathbf{x}(t_0) + \boldsymbol{\eta}(t_k) \quad (6.85)$$

and the covariance equation becomes

$$\mathbf{P}(t_k) = \boldsymbol{\Phi}(t_k, t_0) \mathbf{P}(t_0) \boldsymbol{\Phi}^T(t_k, t_0) + \mathbf{Q}(t_k) \quad (6.86)$$

where

$$\mathbf{P}(t_0) \equiv E \{ \delta \mathbf{x}(t_0) \delta \mathbf{x}^T(t_0) \}, \quad \mathbf{Q}(t_k) \equiv E \{ \boldsymbol{\eta}(t_k) \boldsymbol{\eta}^T(t_k) \} \quad (6.87)$$

The component of $\boldsymbol{\eta}(t_k)$ due to the stochastic acceleration function has already been determined in Eq. (6.74). Formulating this into vector form leads to

$$\mathbf{L}_k = \begin{bmatrix} u_{x,k} \sqrt{\sigma_{u,x}^2 (1 - e^{-2(t-t_0)/\tau})} \\ u_{y,k} \sqrt{\sigma_{u,y}^2 (1 - e^{-2(t-t_0)/\tau})} \\ u_{z,k} \sqrt{\sigma_{u,z}^2 (1 - e^{-2(t-t_0)/\tau})} \end{bmatrix} \quad (6.88)$$

where the approximate discrete-time process \mathbf{L}_k is a 3×1 vector, so the dimension of $\boldsymbol{\eta}(t_k)$ is 9×1 . It is noted from the true dynamic model in Eq. (6.77) that in the RDC algorithm the random part of the stochastic acceleration in Eq. (6.76) does not affect the position and velocity components. Therefore, the expression of $\boldsymbol{\eta}(t_k)$ in the

RDC is approximated by

$$\boldsymbol{\eta}(t_k) = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{L}_k \end{bmatrix} \quad (6.89)$$

The mean of \mathbf{L}_k is zero, and the variance of \mathbf{L}_k is given by Eq. (6.73). Assuming that the components of $\mathbf{u}(t)$ are uncorrelated and all have equal statistics, the variance of \mathbf{L}_k is defined as $E \{ \mathbf{L}_k \mathbf{L}_k^T \} \triangleq [\boldsymbol{\Lambda}]$ computed by

$$[\boldsymbol{\Lambda}] = \begin{bmatrix} \sigma_u^2 (1 - e^{-2(t-t_0)/\tau}) & 0 & 0 \\ 0 & \sigma_u^2 (1 - e^{-2(t-t_0)/\tau}) & 0 \\ 0 & 0 & \sigma_u^2 (1 - e^{-2(t-t_0)/\tau}) \end{bmatrix} \quad (6.90)$$

Finally, the discrete-time process noise covariance matrix \mathbf{Q}_k for position, velocity, and stochastic acceleration can be constructed

$$\mathbf{Q}_k = E \{ \boldsymbol{\eta}(t_k) \boldsymbol{\eta}^T(t_k) \} = \begin{bmatrix} [\mathbf{0}]_{3 \times 3} & [\mathbf{0}]_{3 \times 3} & [\mathbf{0}]_{3 \times 3} \\ [\mathbf{0}]_{3 \times 3} & [\mathbf{0}]_{3 \times 3} & [\mathbf{0}]_{3 \times 3} \\ [\mathbf{0}]_{3 \times 3} & [\mathbf{0}]_{3 \times 3} & [\boldsymbol{\Lambda}]_{3 \times 3} \end{bmatrix} \quad (6.91)$$

RDC can be characterized as a subset of DMC with a simplified process noise matrix. It can be seen that RDC only provides variance terms for the stochastic acceleration components in the process noise matrix of \mathbf{Q}_k . There are no position, and velocity covariance terms. In some cases, with a sufficiently large σ_u , this simplified process noise matrix can play the similar role of the DMC approach in the sense of adaptive model compensation.⁹⁷

CHAPTER VII

APPLICATIONS TO AEROSPACE SYSTEMS

The purpose of this section is to apply the proposed adaptive/nonlinear filtering methods to the aerospace system problems, real-time satellite orbit estimation and autonomous navigation. In the real-time satellite orbit estimation, a ground-based sensor that provides the range, azimuth, and elevation observations is utilized for the measurements with designated sensor accuracies. In the GPS navigation application, both static and dynamic solutions are represented by using the GPS pseudorange and range-rate observations.

7.1 Orbit and Parameter Estimation

7.1.1 Introduction

The evolution of orbit determination began hundreds of years ago with Kepler (c.1610) and Legendre (c.1750).⁹⁸ Gauss (c.1810) gave it a firm analytical and computational basis. During the mid 1800s, Gauss made significant contributions in many areas of mathematics, including statistics, probability theory and estimation of dynamical systems. He invented the technique of deterministic least-squares and applied it to a preliminary orbit determination problem with telescope measurements. Gauss's least-squares method is a corner stone for the current computational method for orbit estimation used by Air Force Space Command (AFSPC).⁹⁹ Many improvements and innovations took place between the original foundations and current theories, but the fundamental principles are the same.

Orbit determination generally consists of two major parts: first, the initial or preliminary orbit determination from a minimum set of observations, and, secondly,

performing estimation to improve the orbit using many observations. Initial orbit determination is defined as the process of determining an initial orbit for a satellite. The estimation process of taking observations and forming an updated state vector and covariance matrix is called differential correction. Propagation between tracks is accomplished by either integrating numerically the differential equations of motion of the space object, or by applying an analytic solution. In some applications the state of the earth-orbiting satellite is represented in a compressed, higher-order polynomial form as alternative to the numerical and analytical techniques.¹⁰⁰ The process of predicting the state of a vehicle is referred to as generating an ephemeris. An ephemeris for a space object consists of the position and velocity components for the spacecraft as a function of time. If forces acting on the spacecraft are known and uncertainties of the forces can be neglected, the motion of the satellite is determined by six parameters. These six can be the position and velocity or a set of six orbit elements at some epoch. However, the deterministic models that assume the system's dynamics is exact can not describe the motion correctly for highly accurate missions. In fact any quantity like uncertainty due to the unknown accelerations in drag and/or solar radiation pressure affects the motion of the spacecraft.¹⁰¹ The unknown acceleration is modeled as a first-order, stationary Gauss Markov stochastic process. In reality the orbit determination (OD) with a filtering procedure may be used to obtain precise estimates of the state of the spacecraft along with calibration of the station clocks, radar biases and geophysical constants.

In this section, we will provide an illustration of the algorithms of the orbit estimation process. The underlying approach is stochastic because observations for orbit determinations include measurement noise in the real world environment. First, the motion of a spacecraft under perturbing accelerations is investigated and modeled by a system of differential equations. The state vector is frequently referred to as the

solution of this system. To predict the state vector of the system in the future we need to provide the initial conditions for the state vector. For accurate orbit prediction, the best estimated initial state and covariance values are essential.^{102,103} For the best initial estimates the nonlinear least-squares differential correction (NLSDC) method can be utilized for the first step of the orbit estimation process. However, it requires a batch of observations from sensors and an *a priori* state in order to produce best initial estimates. Instead of the batch least-squares method, a simple and efficient method that requires only three position vectors, such as Herrick-Gibbs (HG) initial orbit determination algorithm,¹⁰⁴ can be employed to determine the initial orbit. In this dissertation, an extended HG algorithm is proposed to generate not only the position and velocity information but also the state covariance matrix. After obtaining the best estimated state vector and/or covariance matrix, these quantities will be propagated through numerical integration and linear error propagation method over a desired time span, respectively. On the other hand, nonlinear sequential estimation methods, such as the extended Kalman filter, are used for real-time estimation of the spacecraft orbit. In the next section, we show some practical examples for better appreciation of these procedures.

7.1.2 Equations of Orbital Motion

In this section, we present the equations of satellite motion subject to perturbations. The governing equations of motion for a near-earth satellite perturbed by the aspherical earth perturbations and atmospheric drag uncertainty take the following form with position \mathbf{r} and velocity \mathbf{v} with their corresponding initial conditions $\mathbf{r}(t_o)$ and $\mathbf{v}(t_o)$

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_g + \mathbf{a}_d\end{aligned}\tag{7.1}$$

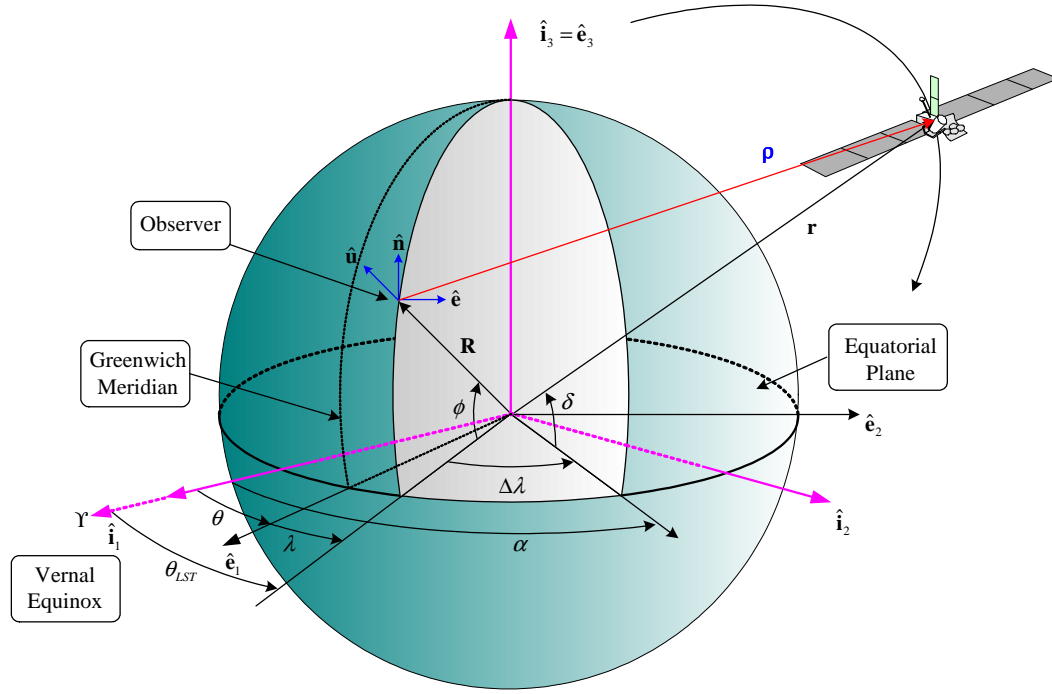


Fig. 7.1 Description of the Geometry of Ground-Based Sensor

where \mathbf{a}_g and \mathbf{a}_d denote the perturbative acceleration terms due to non-spherical earth perturbations and atmospheric drag respectively. r and μ stand for the magnitude of the instantaneous position vector and the product of the gravitational constant and the mass of the earth, respectively. ECI rectangular coordinates are chosen as the reference inertial frame for illustration of the equations of motion. In the rectangular coordinate system in Fig. 7.1, the inertial x axis is the direction of the vernal equinox, the z axis is the normal to the equatorial plane and positive toward north, and the y axis completes a right-handed system. It is convenient to introduce a six-dimensional state vector \mathbf{x} defined by

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \end{bmatrix} \quad (7.2)$$

then, the equation of motion in Eq. (7.1) can be expressed in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \quad (7.3)$$

The solution of the first-order differential equations in Eq. (7.3) involves six constants, which could be the epoch state $\mathbf{x}(t_o)$.

7.1.2.1 Gravitational Perturbation

If the gravitational potential is modeled as a sixth-order aspherical potential function,¹⁰⁵ then it is expressed by

$$\begin{aligned} U = \frac{\mu}{r} & \left[1 - \frac{J_2}{2} \left(\frac{r_e}{r} \right)^2 (3 \sin^2 \delta - 1) \right. \\ & - \frac{J_3}{2} \left(\frac{r_e}{r} \right)^3 (5 \sin^3 \delta - 3 \sin \delta) \\ & - \frac{J_4}{8} \left(\frac{r_e}{r} \right)^4 (35 \sin^4 \delta - 30 \sin^2 \delta + 3) \\ & - \frac{J_5}{8} \left(\frac{r_e}{r} \right)^5 (63 \sin^5 \delta - 70 \sin^3 \delta + 15 \sin \delta) \\ & \left. - \frac{J_6}{16} \left(\frac{r_e}{r} \right)^6 (231 \sin^6 \delta - 315 \sin^4 \delta + 105 \sin^2 \delta - 5) \right] \end{aligned} \quad (7.4)$$

where J_i is i th harmonic coefficient, μ is the earth's gravitational parameter, r is the satellite's orbital radius, r_e is the earth's equatorial radius, δ is the geocentric latitude. The acceleration due to gravity is derived by the gradient of this potential function

$$\mathbf{a}_g = \nabla U = \begin{bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{\partial U}{\partial z} \end{bmatrix} \quad (7.5)$$

If only the second spherical harmonic J_2 is adopted from Eq. (7.4), the perturbation acceleration $\mathbf{a}_g = a_{gx}\hat{\mathbf{i}} + a_{gy}\hat{\mathbf{j}} + a_{gz}\hat{\mathbf{k}}$ along a set of inertial rectangular coordinates about the mass center of the earth with the position vector $\mathbf{r} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}}$

is given by

$$a_{gx} = -\frac{3}{2}J_2 \left(\frac{\mu}{r^2}\right) \left(\frac{r_e}{r}\right)^2 \left\{1 - 5\left(\frac{z}{r}\right)^2\right\} \left(\frac{x}{r}\right) \quad (7.6)$$

$$a_{gy} = -\frac{3}{2}J_2 \left(\frac{\mu}{r^2}\right) \left(\frac{r_e}{r}\right)^2 \left\{1 - 5\left(\frac{z}{r}\right)^2\right\} \left(\frac{y}{r}\right) \quad (7.7)$$

$$a_{gz} = -\frac{3}{2}J_2 \left(\frac{\mu}{r^2}\right) \left(\frac{r_e}{r}\right)^2 \left\{3 - 5\left(\frac{z}{r}\right)^2\right\} \left(\frac{z}{r}\right) \quad (7.8)$$

where $r_e = 6378.165\text{km}$, $J_2 = 0.001082616$.

Note that in this thesis the perturbing accelerations due to the earth's oblateness J_2 in Eqs. (7.6) ~ (7.8) are used for the filter dynamic model, and the aspherical perturbing accelerations up to J_6 are used to generate the true trajectory of a user satellite in GPS navigation application.

7.1.2.2 Drag Perturbation

Atmospheric drag causes a significant perturbation on low-earth- orbit (LEO) satellites, and errors in the drag model can lead to significant errors in the orbit determination and prediction of the satellite motion.¹⁰⁴ The instantaneous drag acceleration due to the atmospheric density is assumed to be opposed to the direction of motion and proportional to the atmospheric density ρ and the velocity squared

$$\mathbf{a}_d = -\frac{1}{2B^*} \rho \|\mathbf{v}_{rel}\| \mathbf{v}_{rel} \quad (7.9)$$

$$B^* = \frac{m_s}{C_d A_s} \quad (7.10)$$

where B^* is the ballistic drag coefficient, C_d is the aerodynamic drag coefficient, m_s is the mass of the satellite, and A_s is the cross sectional area of the satellite in a plane normal to the relative velocity vector \mathbf{v}_{rel} , which is formulated in a set of inertial

rectangular coordinates by

$$\mathbf{v}_{rel} = \begin{bmatrix} \dot{x} + \omega_e y \\ \dot{y} - \omega_e x \\ \dot{z} \end{bmatrix} \quad (7.11)$$

For the precise calculation of drag accelerations knowledge of the atmospheric density as a function of position and time is required. Numerous density models have been developed over the past few decades including the Harris-Priester (HP) model, the Jacchia-Roberts (JR) model, the Mass Spectrometer/Incoherent Scatter (MSIS) model, and a simple exponential model (EM).¹⁰⁶ The MSIS formulation includes density perturbations including solar particle flux heating, annual variations, semi-annual variations, diurnal variations, semi-diurnal variations, and so on. Time-varying models like the Jacchia-Roberts and MSIS models provide accurate data, but they demand high computational power. On the other hand, the simplest model is the exponential model that requires lower computational load. It is used in this dissertation to demonstrate the performance of the proposed nonlinear filtering methods. The same methods can be used with the more complex methods in the real world applications.

For a reference atmospheric density the simple exponential function can be employed, and it is assumed to rotate at the same angular rate of the earth ω_e , which leads to

$$\rho(r) = \rho_0 \exp \left\{ \frac{-(r - r_0)}{H} \right\} \quad (7.12)$$

where ρ_0 is the reference density and H is the scale height at the reference radius r_0 .

Note that the standard exponential density model can be modified to include perturbing variations. For example, the atmospheric density with the diurnal bulge

variation is given by¹⁰⁶

$$\rho = \rho_0 \exp \left\{ \frac{-(r - r_0)}{H} \right\} \left(1 + \cos^4 \frac{\phi}{2} \right) \quad (7.13)$$

where ϕ is the geocentric angle between the satellite and the apex of the diurnal bulge.

7.1.2.3 Stochastic Drag Model

There are non-gravitational forces to be taken into consideration for precise orbit determination. The primary one is the atmospheric drag and ignoring this perturbation for a low-earth orbit will result in significant errors. The uncertainties in the drag acceleration can be attributed to three separate effects: (a) errors in the atmospheric density model, (b) errors in the ballistic coefficient or drag coefficient, and (c) errors in the satellite relative velocity.¹⁰⁷ There are methods taking into account the measurement uncertainty expressed by the standard deviation estimate of the orbital element, and the uncertainty of the force model arising due to density model errors.^{106,108} For example, if the atmospheric drag coefficient C_d or ballistic coefficient β^* is estimated with the satellite state vector, the errors due to the atmospheric scaling factor can be compensated.¹⁰⁹

In this section, the standard deterministic drag model for an orbiting object is replaced by a stochastic drag model that has a deterministic part of the drag acceleration due to the standard exponential form of atmosphere plus a stochastic acceleration part based on three first-order stationary Gauss-Markov processes.⁸⁶

The instantaneous acceleration due to drag is assumed to be opposed to the direction of motion and proportional to the air density ρ and velocity squared, which is explained in Eq. (7.1). Assume that the unknown accelerations in the drag model

can be modeled as a first-order, stationary, Gauss-Markov process

$$\dot{\mathbf{w}} + \boldsymbol{\beta}\mathbf{w} = \mathbf{u}(t) \quad (7.14)$$

where $\mathbf{w}(t) \in \Re^{3 \times 1}$ is the vector of accelerations and $\mathbf{u}(t) \in \Re^{3 \times 1}$ is a zero-mean white, Gaussian noise vector with the covariance matrix

$$\mathbf{q}_u = \begin{bmatrix} \sigma_{u,1}^2 & 0 & 0 \\ 0 & \sigma_{u,2}^2 & 0 \\ 0 & 0 & \sigma_{u,3}^2 \end{bmatrix} \quad (7.15)$$

By virtue of the stationary assumption, $\boldsymbol{\beta}$ is a constant matrix

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & 0 & 0 \\ 0 & \beta_2 & 0 \\ 0 & 0 & \beta_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\tau_1} & 0 & 0 \\ 0 & \frac{1}{\tau_2} & 0 \\ 0 & 0 & \frac{1}{\tau_3} \end{bmatrix} \quad (7.16)$$

where τ_i is the correlation time. The purpose of the three Gauss-Markov parameters (w_i , $i = 1, 2, 3$) is to represent density perturbations due to orbit period, half daily, daily, and monthly or seasonal density variations, that are associated with the selection of the time correlation (τ_i , $i = 1, 2, 3$).

The stochastic exponential density $\rho(r)$ is assumed to be modeled as

$$\rho(r) = \rho_p \exp\{-k(r - r_p)\} + \sum_{i=1}^3 w_i \quad (7.17)$$

where ρ_p and r_p are the nominal density and distance at perigee, and $1/k$ is the density scale height H . The exponential term in the above equation is the deterministic term of the drag acceleration, whereas the w_i are stochastic variables which are the solution obtain from the i th first-order stationary Gauss-Markov processes. The solution of the stochastic Gauss-Markov differential equation is composed of the deterministic

and random parts and is shown in Eq. (6.4). For a scalar case the covariance is computed by

$$E\{w(t)w(t)^T\} = \sigma_w^2(t) = w_o^2 e^{-2(t-t_o)/\tau} + \frac{\tau\sigma_u^2}{2}[1 - 2e^{-2(t-t_o)/\tau}] \quad (7.18)$$

If we choose $w_o^2 = \tau\sigma_u^2$, then

$$\sigma_w^2(t) = E\{w^2(t)\} = \frac{\tau\sigma_u^2}{2} = \text{constant}, \forall t \geq t_o \quad (7.19)$$

As τ approaches zero, then $w(t)$ becomes a pure Gaussian random process (white noise), whereas if τ goes to infinite, $w(t)$ becomes a constant.

7.1.3 Observation and Tracking Model

Orbit estimation of an artificial satellite or space object requires measurements that are related to the satellite's position and velocity. These observations are provided from various sensor systems that measure the properties of some scalar quantity, such as electromagnetic wave propagation between the transmitter and the receiver. Modern space-based systems, such as the Global Positioning System (GPS), provide the position of a GPS receiver. This section illustrates commonly used tracking systems, which are incorporated into the spacecraft orbit estimation and prediction.¹¹⁰

7.1.3.1 Radar Tracking

To update any existing information of the state vector of a satellite, observations must be collected. North American Air Defense Command (NORAD), which has the responsibility for tracking all man-made objects in earth orbit, uses the Space Surveillance Network (SSN) to collect these observations.⁹⁸ This network is comprised of radar sensors for near-earth tracking (below around 6,000 km altitude) and optical sensors for deep-space objects (above 6,000 km altitude). These sensors are

geographically distributed around the world to provide global coverage. Typical observations from a radar site might include azimuth, elevation, range, and range rate, while optical sensors usually provide angles only, such as azimuth and elevation or right ascension and declination. In this study, a radar site (PPE Cape Code NE) located in Cape Code is used for the observations. Data coming from the sensor site (a radar site) used in this work for the orbit determination includes range, azimuth, and elevation.

The general process is to find the sensor site's position and velocity vectors, find the satellite's state vectors relative to the site (ENU), which is referred to the Topocentric-Horizon coordinates, and then transform the position and velocity vectors into the inertial frame. With Fig. 7.1, we establish the basic equations for simulation of the measurements consisting of object range, elevation, and azimuth relative to the earth fixed system, the Topocentric-Horizon frame. The basic equations that govern earth-based tracking are the relationships between the satellite position state, the sensor position, and the observer-to-satellite range vector. This relationship is evident from the geometry of the observation of Fig. 7.1.

The inertial position vector \mathbf{r} is expressed by

$$\mathbf{r} = \mathbf{R} + \boldsymbol{\rho} \quad (7.20)$$

where $\boldsymbol{\rho}$ is the range vector and \mathbf{R} is the radar site vector. The equations relating the observations to the states are relatively straight forward, but highly non-linear. The first step is to consider the site's position described by, for example, declination, ϕ , longitude, λ , and radius. Once we know the site's location, the satellite's position and velocity vector in the topocentric coordinate system is easily obtained by the

following relationship

$$\begin{pmatrix} \rho_x(t) \\ \rho_y(t) \\ \rho_z(t) \end{pmatrix} = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} - \begin{pmatrix} R \cos \phi \cos \theta(t) \\ R \cos \phi \sin \theta(t) \\ R \sin \phi \end{pmatrix} \quad (7.21)$$

$$\begin{pmatrix} \dot{\rho}_x(t) \\ \dot{\rho}_y(t) \\ \dot{\rho}_z(t) \end{pmatrix} = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{pmatrix} - w_e \begin{pmatrix} -R \cos \phi \sin \theta(t) \\ R \cos \phi \cos \theta(t) \\ 0 \end{pmatrix} \quad (7.22)$$

where $R = \|\mathbf{R}\|$, and $\theta(t)$ is the local sidereal time (LST) of the observer location (see Appendix C for details).

From elementary vector algebra, the range is given by

$$\rho(t) = \sqrt{\rho_x(t)^2 + \rho_y(t)^2 + \rho_z(t)^2} \quad (7.23)$$

$$= \sqrt{[x(t) - R \cos \phi \cos \theta(t)]^2 + [y(t) - R \cos \phi \sin \theta(t)]^2 + [z(t) - R \sin \phi]^2}$$

$$\dot{\rho}(t) = \sqrt{\dot{\rho}_x(t)^2 + \dot{\rho}_y(t)^2 + \dot{\rho}_z(t)^2} \quad (7.24)$$

Now, the “up”, “east”, and “north” components of the range vector, $\boldsymbol{\rho}$, are expressed by

$$\boldsymbol{\rho} = \rho_u \hat{\mathbf{u}} + \rho_e \hat{\mathbf{e}} + \rho_n \hat{\mathbf{n}} \quad (7.25)$$

Conversion from the Inertial to the “up”, “east”, and “north” components is made by performing the transformation

$$\begin{pmatrix} \hat{\mathbf{u}} \\ \hat{\mathbf{e}} \\ \hat{\mathbf{n}} \end{pmatrix} = [\mathbf{C}] \begin{pmatrix} \hat{\mathbf{i}} \\ \hat{\mathbf{j}} \\ \hat{\mathbf{k}} \end{pmatrix} \quad (7.26)$$

where the transformation matrix \mathbf{C} is

$$[\mathbf{C}] = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.27)$$

Then, the relationship between the range vector in the Topocentric coordinate and the ECI is obtained by

$$\begin{bmatrix} \rho_u \\ \rho_e \\ \rho_n \end{bmatrix} = [\mathbf{C}] \begin{bmatrix} x - R \cos \phi \cos \theta \\ y - R \cos \phi \sin \theta \\ z - R \sin \phi \end{bmatrix} \quad (7.28)$$

Each component of the range vector in the radar site is expressed by

$$\begin{aligned} \rho_u &= \cos \phi \cos \theta [\rho_x] + \cos \phi \sin \theta [\rho_y] + \sin \phi [\rho_z] \\ \rho_e &= -\sin \theta [\rho_x] + \cos \theta [\rho_y] \\ \rho_n &= -\sin \phi \cos \theta [\rho_x] - \sin \phi \sin \theta [\rho_y] + \cos \phi [\rho_z] \end{aligned} \quad (7.29)$$

Thus, the range is calculated by

$$\rho = \sqrt{\rho_u^2 + \rho_e^2 + \rho_n^2} \quad (7.30)$$

Now, the range-rate equation is calculated by utilizing the range measurement equation in Eq. (7.28) by taking the time derivative

$$\frac{d\rho}{dt} = \frac{\partial \rho}{\partial \rho_u} \dot{\rho}_u + \frac{\partial \rho}{\partial \rho_e} \dot{\rho}_e + \frac{\partial \rho}{\partial \rho_n} \dot{\rho}_n \quad (7.31)$$

After manipulation, the range-rate equation reduces to

$$\frac{d\rho}{dt} = \frac{\{\rho_u \dot{\rho}_u + \rho_e \dot{\rho}_e + \rho_n \dot{\rho}_n\}}{\rho} \quad (7.32)$$

The next step is to define elevation, (el), and azimuth, (az). We measure eleva-

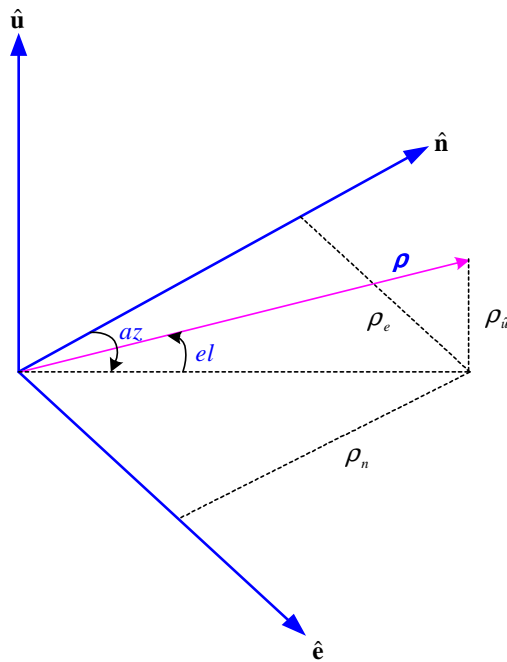


Fig. 7.2 Description of Radar Site for Satellite Observation

tion from the horizon of the site to the satellite and azimuth clockwise from north, that is, azimuth varies from 0 to 360°, whereas elevation varies between -90° and 90°, which is illustrated in Fig. 7.2. This convention is also valid for space-based sensors.

From the geometry of the topocentric radar site in Fig. 7.2, the unique determination of the elevation is obtained by

$$az = \tan^{-1} \left(\frac{\rho_e}{\rho_n} \right) \quad (7.33)$$

and the elevation is computed by

$$el = \tan^{-1} \left(\frac{\rho_u}{\sqrt{\rho_e^2 + \rho_n^2}} \right) \quad (7.34)$$

The relationship between the observation vector \mathbf{y} (range, azimuth, and elevation) and the state vector \mathbf{x} of the dynamic system can be represented in a vec-

tor/matrix form

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, t) \quad (7.35)$$

where $\mathbf{x} = [\mathbf{r}^T \mathbf{v}^T]^T$ is the six-dimensional state vector.

7.1.3.2 Laser Tracking

Satellite laser ranging (SLR) is a technique for precisely measuring the range between a laser station and a satellite that is equipped with retro-reflectors.¹¹⁰ SLR systems provides highly accurate range measurements by determining the turn-around light time of laser pulses transmitted to a satellite and returned by a retro-reflector. Depending on the distance and the resulting strength of the returned signal (along with the attitude and location of the retro reflector on the satellite), accuracies of several centimeters may be achieved.

Modern laser transmitters use a solid-state pulsed laser that applies neodymium as a lasing impurity in a lattice of yttrium aluminum garnet. This allows the generation of green laser light with a wavelength of 532 nm and ultra-short pulses of 30-200 ps width that are repeated at a rate of 5-10 Hz.¹¹⁰ When a laser pulse is transmitted by a telescope, a discriminator starts a time interval counter for initialization of a range measurement. The laser pulse then propagates through the atmosphere until it is reflected by a retro-reflector array onboard a satellite. When the pulse is received at the telescope, a high-speed photodetector stops the time interval counter with a time granularity of less than 20 ps, equivalent to a one-way range precision of better than 3 mm. The half difference of the counter stop and start time multiplied by the velocity of light hence gives an unambiguous average one-way range. The measurement is time-tagged with an accuracy of better than a microsecond, when a rubidium or cesium atomic clock is applied that is regularly synchronized by a GPS time receiver.

The precision of modern SLR systems is usually given as the root-mean-square of the single-shot accuracy over a single pass and is in the order of 5-50 mm. To further reduce the data scatter, normal points are formed at the laser stations by averaging individual range measurements over a several-minute data interval. Due to the high accuracy of SLR data, geodetic applications in the fields of gravity field determination, and Earth rotation parameter estimation are the major applications of SLR. In addition, the development of precise satellite force models, and the calibration of other tracking devices significantly benefit from SLR.

It is noted that laser tracking, in contrast to radar tracking, does not allow autonomous tracking of satellites, but depends on the availability of high-precision *a priori* orbit elements for laser pointing. Furthermore, the use of SLR for regular tracking is restricted due to its dependence on the weather at the laser stations and to the denser operations schedule of the ground segments.

7.1.4 Initial Orbit Determination

The motion of an object is governed by a system of differential equations. The state vector is referred to as the solution of this system. Since the initial conditions determine the solution to the differential conditions we need to determine the precise initial conditions that best approximate the motion of the satellite.

The first-time development of the six orbital elements or state of an earth orbiting satellite by using a set of tracking measurements is commonly referred to as initial or preliminary orbit determination (IOD), which is different from batch filters such as the nonlinear differential correction (NDC), in that the IOD doesn't require the *a priori* state information that is needed for batch filters to start the batch initialization.¹¹¹ Both methods, IOD and NDC, however, can be used to provide an initial state estimate for the recursive filters.

There are several methods for determining the orbital elements from observations.⁹⁸ In this section, the Herrick-Gibbs method, which uses three position vectors for the initial orbit determination, is described.

7.1.4.1 Herrick-Gibbs Method

A typical sensor site's observation of a pass by a satellite usually results in many observations. The Herrick-Gibbs method uses three of these observations (position vectors) to obtain an initial estimate of the orbital elements from a single pass of a satellite through a radar site. It uses the *1st* and *3rd* position vectors to provide an estimate of the velocity at the time of the *2nd* (middle) vector, which yields the position and velocity at the time of the *2nd* vector. This position and velocity is then used as the *a priori* estimate for the differential correction process. The second vector is generally at the mid-point of the pass.⁹⁸ The following derivations are based on Ref. 98.

The angle separation between the position vectors is of interest because this method depends on geometry. Let \mathbf{z}_{23} be the normal vector from the second and third position vector.

$$\mathbf{z}_{23} = \mathbf{r}_2 \times \mathbf{r}_3 \quad (7.36)$$

The coplanar condition is defined from the given initial vectors $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$. The vectors are coplanar if \mathbf{z}_{23} is perpendicular to \mathbf{r}_1

$$\mathbf{z}_{23} \circ \mathbf{r}_1 = 0 \quad (7.37)$$

The coplanar angle is given by

$$\alpha = 90^\circ - \cos \left\{ \frac{\mathbf{z}_{23} \circ \mathbf{r}_1}{|\mathbf{z}_{23}| |\mathbf{r}_1|} \right\} \quad (7.38)$$

If the vectors are coplanar, α will take on a value of zero. Typical values for real world data from a sensor site yield results that may vary by as much as 2° or 3° .

To begin the procedure for calculating the middle velocity, expand the position vector as a Taylor series about the middle time, t_2 . Then, the form of the Taylor series becomes

$$\mathbf{r}(t) = \mathbf{r}_2 + \dot{\mathbf{r}}_2(t - t_2) + \frac{1}{2}\ddot{\mathbf{r}}_2(t - t_2)^2 + \frac{1}{3!}\mathbf{r}_2^{(3)}(t - t_2)^3 + \frac{1}{4!}\mathbf{r}_2^{(iv)}(t - t_2)^4 + \dots \quad (7.39)$$

where $\mathbf{r}_2^{(i)}$ is the i th-order derivative of the position vector \mathbf{r}_2 . Let's define a notation for the time difference as

$$\Delta t_{ij} = t_i - t_j \quad (7.40)$$

Ignoring terms higher than fourth order gives

$$\mathbf{r}_1 = \mathbf{r}_2 + \dot{\mathbf{r}}_2\Delta t_{12} + \frac{1}{2}\ddot{\mathbf{r}}_2\Delta t_{12}^2 + \frac{1}{3!}\mathbf{r}_2^{(3)}\Delta t_{12}^3 + \frac{1}{4!}\mathbf{r}_2^{(iv)}\Delta t_{12}^4 \quad (7.41)$$

$$\mathbf{r}_3 = \mathbf{r}_2 + \dot{\mathbf{r}}_2\Delta t_{32} + \frac{1}{2}\ddot{\mathbf{r}}_2\Delta t_{32}^2 + \frac{1}{3!}\mathbf{r}_2^{(3)}\Delta t_{32}^3 + \frac{1}{4!}\mathbf{r}_2^{(iv)}\Delta t_{32}^4 \quad (7.42)$$

The goal is to find the middle velocity vector. First, eliminate the second order derivative by multiplying the first equation with $-\Delta t_{32}^2$ and add it to the second order equation multiplied by $-\Delta t_{12}^2$

$$\begin{aligned} \mathbf{r}_3\Delta t_{12}^2 - \mathbf{r}_1\Delta t_{32}^2 &= \mathbf{r}_2(\Delta t_{12}^2 - \Delta t_{32}^2) + \dot{\mathbf{r}}_2(\Delta t_{12}^2\Delta t_{32} - \Delta t_{32}^2\Delta t_{12}) + \\ &\quad \frac{\mathbf{r}_2^{(3)}}{6}(\Delta t_{12}^2\Delta t_{32}^4 - \Delta t_{32}^2\Delta t_{12}^3) + \\ &\quad \frac{\mathbf{r}_2^{(iv)}}{24}(\Delta t_{12}^2\Delta t_{32}^4 - \Delta t_{32}^2\Delta t_{12}^4) \end{aligned} \quad (7.43)$$

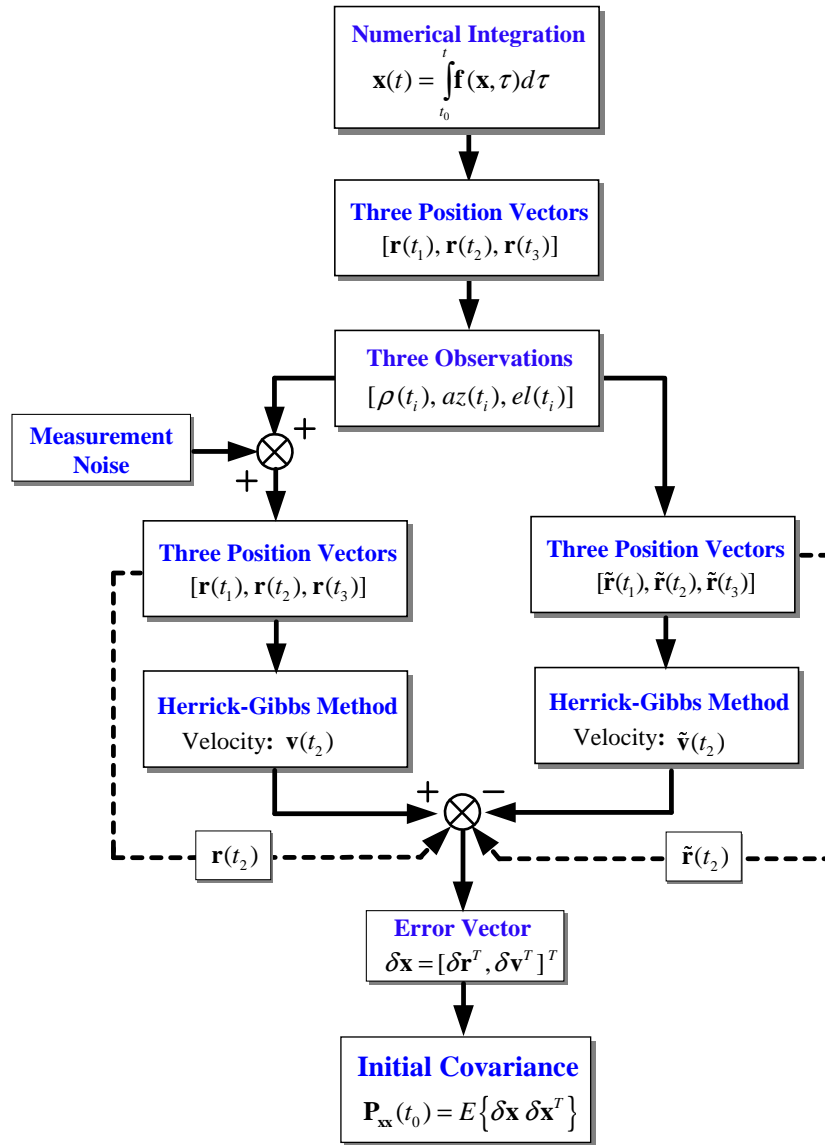


Fig. 7.3 Herrick-Gibbs Method for Initial State and Covariance Information

Now,

$$\Delta t_{12}^2 \Delta t_{32} - \Delta t_{32}^2 \Delta t_{12} = \Delta t_{12} \Delta t_{32} (\Delta t_{12} - \Delta t_{32}) = \Delta t_{12} \Delta t_{32} \Delta t_{13} \quad (7.44)$$

$$\Delta t_{12}^2 \Delta t_{32}^3 - \Delta t_{32}^2 \Delta t_{12}^3 = \Delta t_{12}^2 \Delta t_{32}^2 (\Delta t_{12} - \Delta t_{32}) = \Delta t_{12}^2 \Delta t_{32}^2 \Delta t_{31} \quad (7.45)$$

$$\Delta t_{12}^2 \Delta t_{32}^4 - \Delta t_{32}^2 \Delta t_{12}^4 = \Delta t_{12}^2 \Delta t_{32}^2 (\Delta t_{32}^2 - \Delta t_{12}^2) = \Delta t_{12}^2 \Delta t_{32}^2 \{\Delta t_{32}^2 - \Delta t_{12}^2\} \quad (7.46)$$

After some manipulation, the velocity is obtained⁹⁸

$$\mathbf{v}_2 = -d_1 \mathbf{r}_1 + d_2 \mathbf{r}_2 + d_3 \mathbf{r}_3 \quad (7.47)$$

where the coefficient d_i is given by

$$d_i = \tau_i + \frac{\gamma_i}{r_i^3}, \quad i = 1, 2, 3 \quad (7.48)$$

and

$$\tau_1 = \frac{\Delta t_{23}}{\Delta t_{12} \Delta t_{13}}, \quad \tau_3 = \frac{\Delta t_{12}}{\Delta t_{23} \Delta t_{13}}, \quad \tau_2 = \tau_1 - \tau_3 \quad (7.49)$$

$$\gamma_1 = \frac{\mu}{12} \Delta t_{23}, \quad \gamma_3 = \frac{\mu}{12} \Delta t_{12}, \quad \gamma_2 = \gamma_1 - \gamma_3 \quad (7.50)$$

Note that the Herrick-Gibbs algorithm can be utilized in order to calculate the *a priori* information of an initial state covariance as well as the position and velocity information. The outputs from this algorithm are inputs to the differential correction (DC) and/or the sequential estimation. Therefore, the Herrick-Gibbs algorithm can play the role of a batch filter for the refined orbit estimation. The systematic diagram for the calculation of an initial state covariance is illustrated in Fig. 7.3.

7.1.5 Orbit and Parameter Estimation

In this section, to obtain the best estimate of the state vector of an object that contains error sources, such as process noise and measurement noise, two different

estimation methods are described: (1) Gauss least-squares differential correction, and (2) suboptimal nonlinear filtering.

7.1.5.1 Satellite Visibility Analysis

Consider the orbit estimation of a near-earth satellite using radar observations that consist of range, elevation and azimuth with a known ground station. For simulation study it is necessary to set a criterion for the availability of observations from a radar at the known ground station before applying specific estimation schemes. For the satellite visibility check, a minimum allowable elevation cut-off angle is established. The overall procedures are described in Fig. 7.4.

First, the given orbital parameters of a satellite at some specified time are transformed into the corresponding state vector and the elevation (EL) of the satellite is computed. Assumed that 10 degrees is set as the minimum allowable or threshold value. If EL is less than 10 (deg), the prediction equations are used in order to carry out pure predictions until EL exceeds 10 (deg). Once EL crosses 10 (deg), estimation begins and this mode is continued until EL again becomes less than 10 (deg). Then, the problem enters the realm of prediction.

7.1.5.2 Gauss Least-Squares Differential Correction

The nonlinear least-squares (NLS) method is utilized as a batch filter to estimate the state of a satellite and ballistic coefficient based on noisy observations of range, azimuth, and elevation relative to a radar site.⁹⁹ The NLS algorithm was explained in Fig. 3.1, thus this section focuses on developing specific orbit estimation algorithms based on a designated filter dynamic model.

The objective of the initial orbit determination is to provide for the initial state estimates of the position and velocity, or some unknown parameters from the obser-

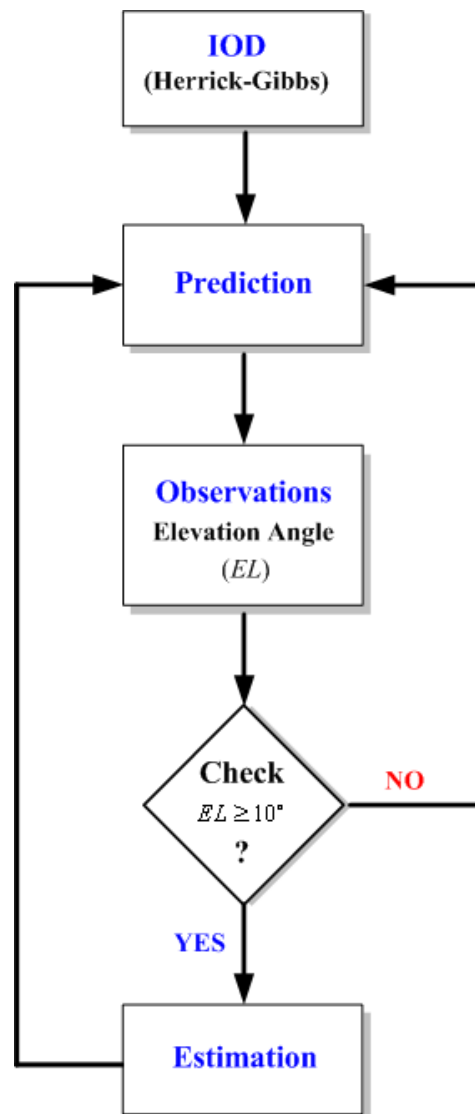


Fig. 7.4 Diagram for Satellite Visibility Check

vations. We include the ballistic coefficient in the state vector since it is generally not known and it is very difficult to model. The filter dynamic model includes the two-body equations of motion plus perturbations due to the earth's oblateness, J_2 , and atmospheric drag, and each acceleration equation was illustrated in Eqs. (7.6)~(7.8) and Eq. (7.9), respectively

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a}_m(\mathbf{r}, \mathbf{v}, \mathbf{p}, t) \\ \dot{\mathbf{p}} &= \mathbf{0}\end{aligned}\tag{7.51}$$

where \mathbf{r} and \mathbf{v} are the position and velocity vectors, respectively, and \mathbf{a}_m is an acceleration function due to the two-body attraction, drag perturbation, and the harmonic geopotential perturbing acceleration up to J_2 . \mathbf{p} includes the system parameters such as the drag coefficient, and gravitational constant.

The first-order system equations can be expressed for simplicity as a vector/matrix equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{p}, t)\tag{7.52}$$

where the state vector $\mathbf{x} = [\mathbf{r}^T \ \mathbf{v}^T]^T$ includes the position and velocity components.

Since the equations of the satellite motion and observations are nonlinear, the GLSDC orbit estimation starts with linearizing the system and measurement equations.

Linearization of System Model

The system solution for state propagation can be obtained by integrating

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^t \mathbf{f}(\mathbf{x}, \mathbf{p}, t) dt\tag{7.53}$$

Differentiating $\mathbf{x}(t)$ with respect to $\mathbf{x}(t_0)$ leads to

$$\left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_0)} \right] = \mathbf{I} + \int_{t_0}^t \left\{ \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}(t)} \right] \left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_0)} \right] + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{p}} \right] \left[\frac{\partial \mathbf{p}}{\partial \mathbf{x}(t_0)} \right] \right\} dt \quad (7.54)$$

Since the partial of the parameter \mathbf{p} with respect to the initial state $\mathbf{x}(t_0)$ is equal to zero

$$\left[\frac{\partial \mathbf{p}}{\partial \mathbf{x}(t_0)} \right] = \mathbf{0} \quad (7.55)$$

it reduces to

$$\left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_0)} \right] = \mathbf{I} + \int_{t_0}^t \left\{ \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}(t)} \right] \left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_0)} \right] \right\} dt \quad (7.56)$$

Now, differentiating the above equation with respect to time t provides

$$\frac{d}{dt} \left\{ \left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_0)} \right] \right\} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}(t)} \right] \left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_0)} \right] \quad (7.57)$$

which becomes the differential state transition matrix equation in the form

$$\dot{\Phi}(t, t_0) = \mathbf{F} \Phi(t, t_0), \quad \Phi(t_0, t_0) = \mathbf{I} \quad (7.58)$$

where $\Phi(t, t_0)$ is the state transition matrix defined by

$$\Phi(t, t_0) = \left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_0)} \right] \quad (7.59)$$

and the Jacobian matrix \mathbf{F} is

$$\mathbf{F} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right] \quad (7.60)$$

Now, differentiating $\mathbf{x}(t)$ with respect to \mathbf{p} gives

$$\left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{p}} \right] = \left[\frac{\partial \mathbf{x}(t_0)}{\partial \mathbf{p}} \right] + \int_{t_0}^t \left\{ \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}(t)} \right] \left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{p}} \right] + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{p}} \right] \right\} dt \quad (7.61)$$

where

$$\left[\frac{\partial \mathbf{x}(t_0)}{\partial \mathbf{p}} \right] = \mathbf{0} \quad (7.62)$$

Then, the partial derivative becomes

$$\left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{p}} \right] = \int_{t_0}^t \left\{ \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}(t)} \right] \left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{p}} \right] + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{p}} \right] \right\} dt \quad (7.63)$$

Differentiation of the above equation with respect to time t leads to

$$\frac{d}{dt} \left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{p}} \right] = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}(t)} \right] \left[\frac{\partial \mathbf{x}(t)}{\partial \mathbf{p}} \right] + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{p}} \right] \quad (7.64)$$

which can be further defined by

$$\dot{\Psi}(t, t_0) = \mathbf{F} \Psi(t, t_0) + \mathbf{B} \quad (7.65)$$

where $\Psi(t)$ is the sensitivity matrix defined by

$$\Psi(t) = \left[\frac{\partial \mathbf{x}}{\partial \mathbf{p}} \right] \quad (7.66)$$

and the term \mathbf{B} is the Jacobian matrix employed by

$$\mathbf{B} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{p}} \right] \quad (7.67)$$

Linearization of Measurement Model

Assume that range (ρ), azimuth (az), and elevation (el) are available as measurements from a sensor. Then, the measurement model is expressed by

$$\tilde{\mathbf{y}} = \mathbf{h}(\mathbf{x}, \mathbf{p}, \mathbf{k}, t) + \mathbf{v}(t) \quad (7.68)$$

where the elements of \mathbf{k} are the other parameters such as biases, and $\mathbf{v}(t)$ is measurement noise, which is assumed to be white Gaussian with zero mean and known

covariance matrix

$$\begin{aligned} E \{ \mathbf{v}(t) \} &= \mathbf{0} \\ E \{ \mathbf{v}(t) \mathbf{v}(s)^T \} &= \mathbf{R}(t) \delta(t - s) \end{aligned} \quad (7.69)$$

where $\mathbf{R}(t) = \mathbf{R}(t)^T \geq 0$ is positive semi-definite. The partial derivative of the measurement equation is obtained by

$$\mathbf{H} = \left[\frac{\partial \mathbf{h}(t)}{\partial \{ \mathbf{x}_0, \mathbf{p}, \mathbf{k} \}} \right] = \left[\frac{\partial \mathbf{h}(t)}{\partial \mathbf{x}(t)} \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_0)} : \frac{\partial \mathbf{h}(t)}{\partial \mathbf{x}(t)} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} : \frac{\partial \mathbf{h}(t)}{\partial \mathbf{x}(t)} \frac{\partial \mathbf{x}}{\partial \mathbf{k}} \right] \quad (7.70)$$

and it can be further expressed by using the definition of the state transition

$$\mathbf{H} = \left[\frac{\partial \mathbf{h}(t)}{\partial \mathbf{x}(t)} \boldsymbol{\Phi}(t, t_0) : \frac{\partial \mathbf{h}(t)}{\partial \mathbf{x}(t)} \boldsymbol{\Psi}(t) : \frac{\partial \mathbf{h}(t)}{\partial \mathbf{k}} \right] \quad (7.71)$$

The components of the Jacobian matrix \mathbf{H} are formulated by

$$\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}(t_0)} \right] = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}(t)} \right] \left[\frac{\partial \mathbf{x}}{\partial \mathbf{x}(t_0)} \right] = \mathbf{G}(t) \boldsymbol{\Phi}(t, t_0) \quad (7.72)$$

where

$$\mathbf{G} = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}(t)} \right] \quad (7.73)$$

Finally, the GLSDC orbit estimation algorithm is summarized in Fig. 7.5.

7.1.5.3 Nonlinear Filtering Algorithm

In this subsection, the dynamic state solution of the satellite orbit estimation is based on the utilization of nonlinear filters due to the nonlinearity in the system and measurement equations. The basic state vector for the filtering algorithm in the orbit estimation is defined by

$$\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, C_d]^T \in \Re^{7 \times 1} \quad (7.74)$$

where C_d is included to allow adjustment of the dynamic model uncertainty, and to compensate for the dynamic modeling errors. The true real-world dynamical equa-

GAUSS LEAST-SQUARES DIFFERENTIAL CORRECTION

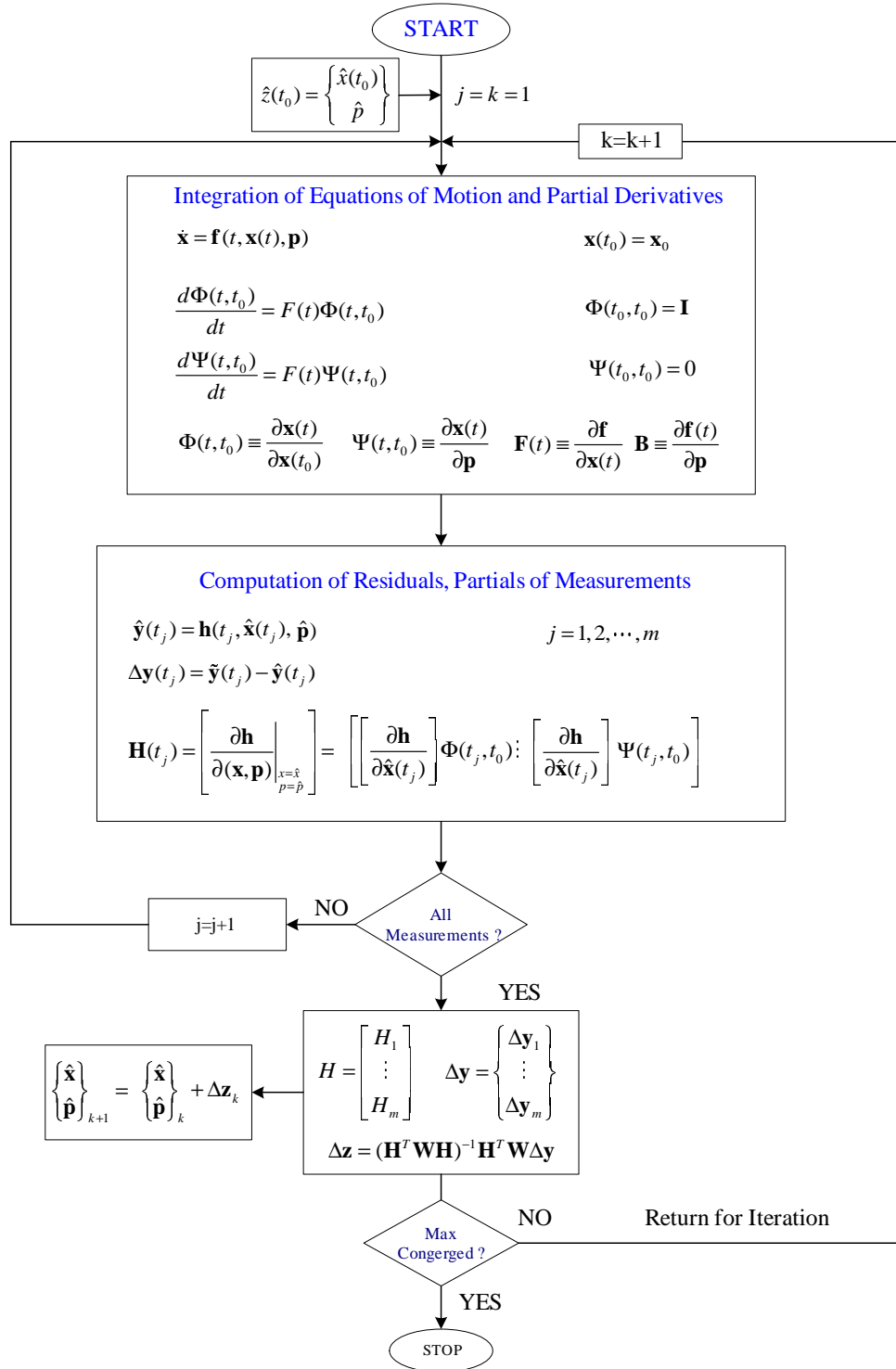


Fig. 7.5 Diagram for GLSDC Orbit Estimation

tions of motion are assumed to be

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a}_m(\mathbf{r}, \mathbf{v}, \mathbf{p}, t) + \mathbf{w}(t) \\ \dot{\mathbf{p}} &= \mathbf{0}\end{aligned}\tag{7.75}$$

where \mathbf{r} and \mathbf{v} are the position and velocity vectors, respectively, and \mathbf{a}_m is an acceleration function due to the two-body attraction, drag perturbation, and the harmonic geopotential perturbing acceleration up to J_2 . The constant parameter \mathbf{p} consists of the drag coefficient. $\mathbf{w}(t)$ is a white, Gaussian process noise which represents the effects of model errors with the properties

$$E\{\mathbf{w}(t)\} = 0, \quad E\{\mathbf{w}(t)\mathbf{w}^T(s)\} = \mathbf{Q}(t)\delta(t-s)\tag{7.76}$$

Then, the vector/matrix equation in terms of the force vector \mathbf{f} , which is the time derivative of the state vector, can be written as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{w}(t)\tag{7.77}$$

where the state vector \mathbf{x} is the augmented state vector expressed in Eq. (7.74), and $\mathbf{w}(t)$ has the state noise covariance matrix $\mathbf{Q}(t)$ given by

$$\mathbf{Q}(t) = \sigma_w^2 [\mathbf{I}_{3 \times 3}]\tag{7.78}$$

Now, consider the nominal or filter dynamic model in a vector/matrix form represented by

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, t)\tag{7.79}$$

The expression for the state sensitivity matrix \mathbf{F} , defined as the partial derivative

of the force vector \mathbf{f} with respect to the nominal state vector $\hat{\mathbf{x}}$, is calculated from

$$\mathbf{F} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} & \frac{\partial \dot{x}}{\partial z} & \frac{\partial \dot{x}}{\partial \dot{x}} & \frac{\partial \dot{x}}{\partial \dot{y}} & \frac{\partial \dot{x}}{\partial \dot{z}} & \frac{\partial \dot{x}}{\partial C_d} \\ \frac{\partial \dot{y}}{\partial x} & \dots & & & & & \\ \frac{\partial \dot{z}}{\partial x} & \dots & & & & & \\ \frac{\partial \ddot{x}}{\partial x} & \dots & & & & & \\ \frac{\partial \ddot{y}}{\partial x} & \dots & & & & & \\ \frac{\partial \ddot{z}}{\partial x} & \dots & & & & & \\ \frac{\partial \dot{C}_d}{\partial x} & \frac{\partial \dot{C}_d}{\partial y} & \frac{\partial \dot{C}_d}{\partial z} & \frac{\partial \dot{C}_d}{\partial \dot{x}} & \frac{\partial \dot{C}_d}{\partial \dot{y}} & \frac{\partial \dot{C}_d}{\partial \dot{z}} & \frac{\partial \dot{C}_d}{\partial C_d} \end{bmatrix} \quad (7.80)$$

where the non-zero terms of this matrix are listed in Appendix A. The state transition matrix Φ is obtained by integrating the following matrix differential equation

$$\dot{\Phi}(t_k) = \mathbf{F}(t_k, t_0)\Phi(t_0), \quad \Phi(t_0) = \mathbf{I} \quad (7.81)$$

The integration for the state transition matrix is made by using the Runge-Kutta numerical method.

Suppose that a reference trajectory $\hat{\mathbf{x}}^-(t_k)$ is obtained from the propagated state estimate in the filtering propagation procedure. Then, the stochastic discrete-time linear dynamics from the linear error theory forced by process noise $\boldsymbol{\eta}(t_k)$ can be written by

$$\delta \mathbf{x}(t_k) = \Phi(t_k, t_0)\delta \mathbf{x}(t_0) + \boldsymbol{\eta}(t_k) \quad (7.82)$$

where

$$\delta \mathbf{x}(t_0) = \mathbf{x}(t_0) - \hat{\mathbf{x}}^+(t_0) \quad (7.83)$$

$\mathbf{x}(t_0)$ is the true state at time t_0 , and $\hat{\mathbf{x}}^+(t_0)$ is the estimate of the true state at time t_0 , conditioned on observations through t_0 . Then, from the definition of the propagated covariance matrix $\mathbf{P}(t_k)$ at the current time t_k conditioned on observations through

t_0 in Eq. (6.35), the covariance equation becomes

$$\mathbf{P}(t_k) = \mathbf{\Phi}(t_k, t_0)\mathbf{P}(t_0)\mathbf{\Phi}^T(t_k, t_0) + \mathbf{Q}(t_k) \quad (7.84)$$

where

$$\mathbf{P}(t_0) \equiv E\{\delta\mathbf{x}(t_0)\delta\mathbf{x}^T(t_0)\}, \quad \mathbf{Q}(t_k) \equiv E\{\boldsymbol{\eta}(t_k)\boldsymbol{\eta}^T(t_k)\} \quad (7.85)$$

The discrete-time process noise covariance matrix for the position, velocity, and drag coefficient can be constructed by

$$\mathbf{Q}(t_k) = \begin{bmatrix} \frac{\Delta t^3 \sigma_w^2}{3} [\mathbf{I}_{3 \times 3}] & \frac{\Delta t^2 \sigma_w^2}{2} [\mathbf{I}_{3 \times 3}] & \mathbf{0}_{1 \times 1} \\ \frac{\Delta t^2 \sigma_w^2}{2} [\mathbf{I}_{3 \times 3}] & \Delta t \sigma_w^2 [\mathbf{I}_{3 \times 3}] & \mathbf{0}_{1 \times 1} \\ \mathbf{0}_{1 \times 1} & \mathbf{0}_{1 \times 1} & \mathbf{0}_{1 \times 1} \end{bmatrix} \quad (7.86)$$

7.1.6 Error Sources and Compensation

If the dynamic model and measurements were perfect, then the orbit determination would be easily accomplished without complex estimation procedures. However, the real world is not perfect since errors arise from numerous sources. In the next section, error sources in the application of the orbit estimation are investigated.

7.1.6.1 Measurement Error

All observations obtained from various ground-based or on-board sensors are easily influenced by errors which come from many sources;⁹⁸ for example, receiver noise and sensor calibration. To solve the limitations of the sensor coverage, sensors are located throughout the world to provide more frequent measurements. Measurement errors can be quantified by using the error covariance matrix, \mathbf{R} , which indicates the variation in the observations about their true value. Measurement errors can be divided into two main categories: noise and biases. Bias is a slowly varying constant

offset from the true value, whereas noise is a statistical indication of the random variation which is scattered about the observed mean value. If we know the feature of the biases, they can be subtracted from each observation. The measurement noise can also be incorporated into the estimation process, provided its statistics are known.

When it comes to the time-varying errors due to inaccuracy of time, all clocks suffer from white noise in their frequency so that tracking systems which rely on a transmitting and receiving clock are subject to this error.

7.1.6.2 Mathematical Models

The next main error source comes from the mathematical dynamic model. Even if the dynamic model is precise with complex atmosphere (Jacchia density model) and gravitational models, it is not perfect. The process noise $\mathbf{v}(t)$ is used to represent the error in the mathematical modeling of the system dynamics. Process noise accounts for each of these errors. After assuming that these are white noises we can interpret the characteristics of process noises using a statistical method. However, in the real world, they are correlated with time, that is, not white noises, but colored noises or non-Gaussian noises. One of the advantages of the particle filtering methods is the ability of handling non-Gaussian noises. In this work, only Gaussian noises are discussed and incorporated into systems and measurements.

7.2 Autonomous GPS Navigation

The basic principle of GPS positioning and navigation is to obtain the distance between the user vehicle and each visible GPS satellite by measuring the time that a signal travels from the satellite to the user.¹¹² The user's three unknown position parameters can thus be estimated from four such distances, accounting for an ad-

ditional unknown that is the receiver's clock offset Δt from the GPS system time. This is illustrated for the pseudorange positioning, where the signal traveling time is determined by the time shift required for a match between the received code from the satellite and the receiver replica using the start time information. Each satellite also broadcasts its clock offset, Δt in the figure relative to the GPS system time so that only four unknowns need to be determined. It is clear that at least four satellites have to be in sight. If more satellites are visible, the positioning errors due to the measurement errors can be reduced. For example, even though four satellites are only needed for an instantaneous fix, one can provide a fix over a period of time and any GPS measurements can be used to provide an orbit update. In this section, GPS navigation solutions are presented in terms of point positioning and dynamic filtering.

7.2.1 GPS Navigation Systems

GPS is the abbreviation for Navigation Satellite Timing and Ranging/Global Positioning System-NAVSTAR/GPS. It is a satellite based navigation system with worldwide coverage that allows a user access to accurate positioning and timing information. GPS has been described as the most significant development for safe and efficient navigation and surveillance of air and spacecraft since the introduction of radio navigation 50 years ago. It has had a widespread impact, which continues to grow due to its high accuracy, global availability, and low cost.¹¹³

7.2.1.1 GPS Systems

The GPS system consists of three subsystems²⁷ (i) Space Segment (Satellite System): Broadcast position and time signals as well as other messages (ii) User Segment (GPS Receivers): Receive signal to determine position, time and so on. (iii) Ground Control Segment: Update information disseminated by each satellite, monitor satellite health,

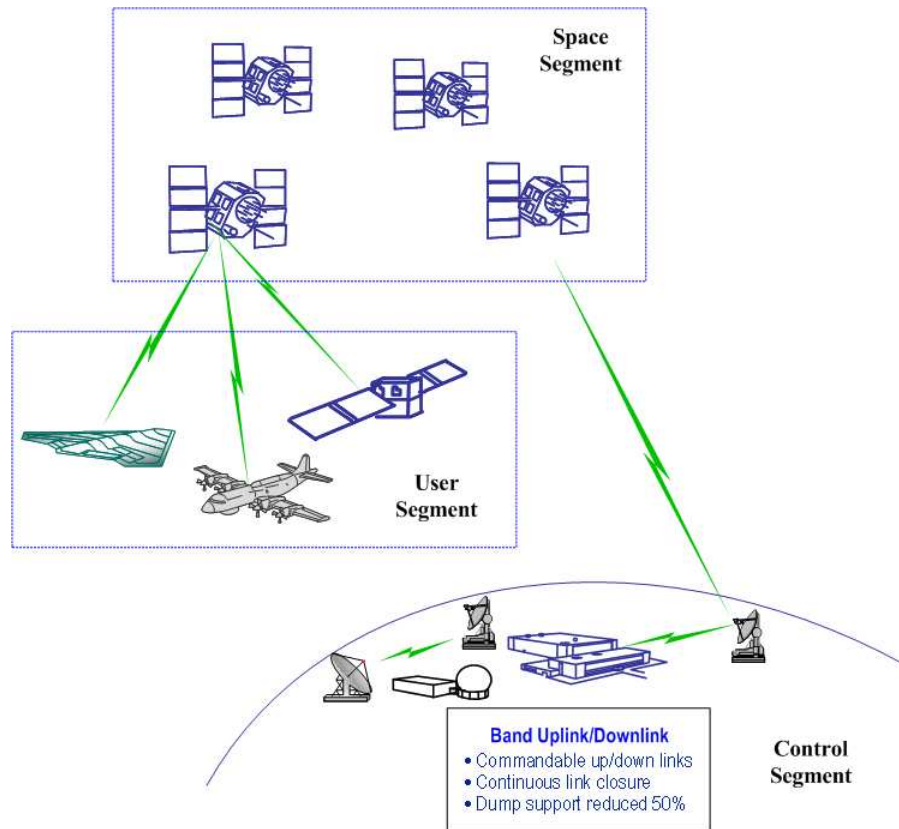


Fig. 7.6 Diagram for the GPS Segments

control satellites. The master station is located at Colorado Springs and there exist several updating stations and monitor stations.¹¹⁴ These concepts are illustrated in Fig. 7.6. For more details on the GPS system see Refs. 113 and 114.

7.2.1.2 GPS Satellite Constellation

The current constellation consists of 24 satellites in six 12-hour orbit planes (4 in each) inclined at 55° to the equatorial plane at a nominal altitude of 20,200 *km*. With this constellation, a user anywhere on the globe at any time of the day will receive signals from at least four, at most eleven, and most often six to eight satellites. The GPS constellation is illustrated in Fig. 7.7.



Fig. 7.7 The GPS Satellite Constellation

7.2.1.3 GPS Measurement Models

(1) *GPS Signals*

The signal broadcast by each satellite consists of three elements; a carrier, a unique spread-spectrum code of pseudorandom numbers, and a binary data message. It includes the satellite's current position, the time when the signal is sent, and satellite's health information, etc. Each satellite broadcasts signals on two carriers: the major carrier L1 with frequency $f_1 = 1575.42 \text{ MHz}$, and the secondary carrier L2 with frequency $f_2 = 1227.60 \text{ MHz}$. Two pseudorandom noise (PRN) codes are modulated on the two base carriers. The first code is the coarse acquisition code (C/A code) which is available for civilian use with the Standard Positioning Service (SPS). The second code is the precision code (P code or Y code) for military use and designated as the Precision Positioning Service (PPS). The P-code is modulated on both carriers L1 and L2, whereas the C/A code is modulated upon only L1.

In addition to the PRN codes, a data message on each GPS satellite is modulated

on the carriers consisting of status information, satellite clock bias, and satellite ephemeris. A detailed signal description is found in Ref. 113.

The three types of measurements can be obtained from GPS receivers based on the GPS signals.

- *Pseudorange Measurements*: These are derived from the PRN codes
- *Carrier Phase Measurements*: These are obtained by measuring the phase of the incoming carrier (L1 and/or L2), and the range to a satellite can be computed by measuring an ambiguous number of cycles
- *Doppler Measurements*: The derivative of the carrier phase measurement is the Doppler shift due to the relative motion between the receiver and the GPS satellite

(2) *Pseudorange Measurement*

The basic equation for GPS Positioning is the following *pseudorange equation* for n visible satellites

$$\rho_i = r_i + c\Delta t + w_i = \psi_i + w_i, \quad i = 1, 2, \dots, n \quad (7.87)$$

where ρ_i is the pseudorange between the user and satellite i , $\psi_i = r_i + c\Delta t$ noiseless pseudorange to satellite i , w_i is the measurement error for satellite i , c is the speed of light 3×10^8 m/s, Δt is the timing error (receiver clock offset from GPS time). r_i is the true range to satellite i and is expressed by

$$r_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2} \quad (7.88)$$

where (X_i, Y_i, Z_i) are the coordinates of GPS satellite i , and (x, y, z) are coordinates of user.

The pseudorange measurement error w_i includes errors due to selective availability (SA), ionospheric and tropospheric refraction, multipath effect, satellites position errors, and receiver noise, etc.

The policy of the selective availability (SA) intentionally introduces two types of errors into the C/A code for signal degradation so as to prevent unauthorized use such as precision guidance of weapons: erroneous or imprecise values of the satellite ephemeris parameters, which are often referred to as orbit SA, and an erroneous drift of the GPS satellite time, known as dither SA. They affect primarily the pseudorange and delta range measurements, respectively. The SA policy has been discontinued since May 1, 2000. The prevention of the hostile use of GPS is accomplished through other measures, such as selective and local denial of GPS signals.¹¹⁵

The pseudorange measurements are based on determining the time shift (or phase difference) of the C/A and P codes. Measurements can also be made based on the phase and/or frequency shift between the receiver-replicated and the actually received carriers, leading to the delta range, integrated Doppler and/or carrier phase measurements.

(3) *Doppler Measurement*

The range rate between the GPS satellite and the user vehicle can be derived from the Doppler shifted GPS carrier signal. The GPS Doppler measurements measure the Doppler shift in carrier frequency due to the relative motion of the GPS satellite and the user, as well as the frequency error or drift of the satellite and user clocks. The Doppler shift caused by satellite and user motion is the projection of the relative velocity along the line-of-sight direction scaled by the transmitted frequency $L_1 = 1575.42\text{MHz}$ divided by the speed of light. The user's receiver contains an oscillator which produces the reference carrier frequency, and this signal is used to produce the best frequency with the received, Doppler shifted signal. This best frequency

measured in the receiver can be expressed as

$$D_i = - \left(\frac{\mathbf{v}_i - \mathbf{v}_u}{c} \circ \frac{\mathbf{r}_i - \mathbf{r}_u}{\|\mathbf{r}_i - \mathbf{r}_u\|} \right) L_1 \quad (7.89)$$

Now, the Doppler can be converted into a pseudorange rate measurement and can be written by

$$\dot{\rho}_i = (\mathbf{v}_i - \mathbf{v}_u) \circ \frac{\mathbf{r}_i - \mathbf{r}_u}{\|\mathbf{r}_i - \mathbf{r}_u\|} + f + v_{\dot{\rho}_i} \quad (7.90)$$

where f is the receiver clock drift in m/s , and $v_{\dot{\rho}_i}$ is the range rate observation error. The range rate observed by the user is called pseudorange rate.

A measurement of the range rate may be made by counting the number of Doppler cycles over a short period, which is known as the delta range measurement. Alternatively, if the Doppler count is kept running, then a so-called integrated Doppler, accumulated delta range, or continuous carrier phase measurement is formed. Since frequency is the variation rate of phase, both of these measurements actually amount to measuring the carrier phase.

(4) *Carrier Phase Measurement*

The noisy carrier phase measurement of each satellite is

$$\phi = \frac{\psi}{\lambda} + n_\phi + \omega_\phi \quad (7.91)$$

where n_ϕ is the integer ambiguity and ω_ϕ is the carrier phase measurement error. The integer ambiguity n_ϕ appears because it is impossible to distinguish two carriers that differ only in phase with an integer cycle and thus only fractional-cycle phase difference is measurable. Its determination is the crux to the utilization of the carrier phase measurements. The measurement error ω_ϕ is made up of time-correlated errors and white measurement noise. It includes errors caused by atmospheric refraction, satellite ephemeris error, and the receiver measurement noise.

Note that the pseudorange and carrier-phase measurements in the absence of integer ambiguity obey two equations of basically the same form. Since the carrier frequency is much higher than that of the codes, the carrier-phase measurements would be much more accurate than the pseudorange measurements if the integer ambiguity could be completely removed, which is unfortunately impossible. Also, continuous tracking of the carrier is required in the carrier-phase measurements. Interruption may occur in reality due to receiver failure or external disturbance, resulting in what is known as a cycle slip. Note, however, that this ambiguity does not affect the delta-range and range rate measurements.¹¹⁵

7.2.1.4 GPS Positioning Accuracy

(1) *Dilution of Precision*

Dilution of Precision (DOP) is the phenomenon that occurs when an object is localized based on distance measurements along lines of sight (LOS) from the object to reference points, when these LOS have small angles between them. In such a case, due to the ill-conditioning of the estimation problem, the uncertainty in a direction perpendicular to the LOS is significantly larger than along the LOS.

The current GPS satellite constellation guarantees that four to eleven satellites are visible for a user anywhere on the globe. In general, there are more equations than unknowns. Thus the least-squares solution of the equations are found.

Linearizing the pseudorange equation yields

$$\begin{aligned} z_1 &= H_1 x + v_1 \\ &\vdots \\ z_n &= H_n x + v_n \end{aligned} \tag{7.92}$$

In vector and matrix form, it is can be written as

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (7.93)$$

If the measurement error vector \mathbf{v} has the covariance \mathbf{R} , then the LS estimator yields

$$\hat{\mathbf{x}} = [\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{z} \quad (7.94)$$

$$\mathbf{P}_x = [\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \quad (7.95)$$

If the measurement errors are uncorrelated and have approximately equal variance σ^2 , then $\mathbf{R} = \mathbf{I}\sigma^2$ and

$$\hat{\mathbf{x}} = [\mathbf{H}^T \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{z} \quad (7.96)$$

$$\mathbf{P}_x = [\mathbf{H}^T \mathbf{H}]^{-1} \sigma^2 \quad (7.97)$$

where \mathbf{H} depends only on the relative geometry of the satellites and the user.

The geometric DOP (GDOP) matrix is defined by¹¹⁶

$$[\mathbf{H}^T \mathbf{H}]^{-1} \triangleq \begin{bmatrix} E_{DOP}^2 & & & \\ & N_{DOP}^2 & & \\ & & V_{DOP}^2 & \\ & & & T_{DOP}^2 \end{bmatrix} \quad (7.98)$$

where the off-diagonal terms are omitted, and

$$\begin{aligned} E_{DOP}^2 &= \text{east DOP} \\ N_{DOP}^2 &= \text{north DOP} \\ V_{DOP}^2 &= \text{vertical DOP} \\ T_{DOP}^2 &= \text{time DOP} \end{aligned} \quad (7.99)$$

The scalar position DOP value is given by¹¹⁶

$$\text{Position DOP} = \sqrt{\text{tr} [(\mathbf{H}_3^{-1} \mathbf{H}_3)^{-1}]} \quad (7.100)$$

where \mathbf{H}_3 consists of the position part of \mathbf{H} . The position DOP is an instantaneous measure of the observability condition and a scaling factor that relates the actual positioning accuracy and the measurement error in one direction.

(2) GPS Positioning Accuracy

The current nominal 1σ magnitudes and decorrelation time constants of the major components contribute to the GPS measurement errors. The time constants are the parameters corresponding to the correlation time of an autocorrelated (colored) random process. The most commonly used measures of accuracy for vertical, horizontal, and 3-D positioning are the 2σ errors¹¹⁶

$$\begin{aligned} 2\sigma V_{DOP} &= \text{two times } \sigma \times \text{vertical DOP} \\ 2\sigma H_{DOP} &= 2\sigma \sqrt{E_{DOP}^2 + N_{DOP}^2} \\ 2\sigma P_{DOP} &= 2\sigma \sqrt{E_{DOP}^2 + N_{DOP}^2 + V_{DOP}^2} \end{aligned} \quad (7.101)$$

where $H_{DOP} = \text{horizontal DOP}$, $P_{DOP} = \text{positional DOP}$. However, the common term *2drms* stands for twice distance rsm, rather than two-dimensional rms, although the distance is measured in a 2-D space, the horizontal plane.

Note that 2drms is not equivalent to a 95% error for a 2-D distribution. It actually corresponds to a 98% error for a circular Gaussian distribution.

The GPS satellite constellation was designed to minimize GDOP. The upper limit for P_{DOP} was 10. The worldwide daylong median values are $P_{DOP} = 2.0$ and $H_{DOP} = 1.2$. For a normal situation $H_{DOP} \leq 4$.

Note that there are no published specifications for velocity estimates. They are obtained internally by the GPS receiver by filtering the measurements.

7.2.1.5 Coordinate Transformation

The satellite position signals are in the World Geodetic Systems of 1984 (WGS-84) coordinate system. Usually, it is necessary to express the positioning solution in the navigation reference frame, and thus a transformation is needed. By performing a series of linear translational and rotational transformations, a linear transformation from the WGS-84 system to the navigation reference coordinate system (east, north, up) can be obtained by²⁷

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = C_{WGS}^N \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{WGS-84}} - C_{WGS0}^N \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (7.102)$$

where

$$C_{WGS}^N = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\sin \phi \cos \lambda & -\sin \phi \sin \lambda & \cos \phi \\ \cos \phi \cos \lambda & \cos \phi \sin \lambda & \sin \phi \end{bmatrix} \quad (7.103)$$

$$C_{WGS0}^N = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\sin \phi \cos \lambda & -\sin \phi \sin \lambda & \cos \phi \\ \cos \phi \cos \lambda & \cos \phi \sin \lambda & \sin \phi \end{bmatrix} \quad (7.104)$$

and (ϕ, λ) are the local reference latitude and longitude, respectively. (x_0, y_0, z_0) are the coordinates of the origin of the navigation reference coordinate system in the WGS-84 system.

7.2.2 Navigation Solution

There are two basic solution strategies for deriving user state estimates from the pseudorange and range rate measurements. One is the static or kinematic solution and the other is the dynamic solution. If four simultaneous measurements from distinct

GPS satellites are available, the user's state can be determined. Four simultaneous pseudorange measurements are required to solve for the four components of user position and clock bias. With these position and bias solutions, four range rate measurements produce a solution of the user velocity and clock drift components.

The GPS nonlinear measurement equations can be solved by employing either closed form solutions^{117,118} or iterative techniques¹¹⁴ based on linearization for static solutions. The general problem of the existence and uniqueness of the position and bias in terms of a closed form solution is addressed in Ref. 118. This work provides the basis for the geometric algebraic solution used in this work. When properly formulated and implemented, the geometric solution is stable and fast, but its accuracy is limited by considering geometry and measurement noise.⁹⁶ Note that as alternative point solution the least-squares differential correction method can be utilized to produce the user positioning solutions. The geometric algebraic solution does not utilize a dynamic model to describe the evolution of the state and hence does not produce any covariance information or permit the propagation of the state estimate beyond the current observation time. However, the solutions from the geometric method can be used for the initial *a priori* estimates of user position, velocity, clock bias, and clock drift to the dynamic state solution process.

Any existing nonlinear filter such as EKF could be used for the dynamic solution. The use of a dynamic model in order to propagate the state and covariance estimates makes the dynamic solution more tolerant to poor viewing geometry than the geometric solution technique.⁹⁶ However, the weakness of the dynamic state estimation is stability in that it is susceptible to saturation and the subsequent divergence of the state estimate. As the number of observations increases, the magnitude of the covariance of the state estimate tends to decrease. Saturation occurs when the state covariance matrix becomes so small or nearly singular that the filter essentially ig-

nores observational data and simply generates state solutions based on its dynamic model.

7.2.3 Kinematic Navigation Solutions

7.2.3.1 Geometric State Solution

The geometric algorithm used is based on a study of the existence and uniqueness of GPS geometric solution.^{118,119}

(1) *Position and Clock Bias Solution*

Assume n GPS satellites located in m -space at the position \mathbf{x}_i , $i = 1, 2, \dots, n$, in our system ($n=4$, $m=3$). The coordinate system is chosen so that the origin coincides with the position of one of the GPS satellites, so that $\mathbf{x}_1 = 0$. The location of the user is \mathbf{x}_u . The distance between the user to the origin is

$$r_u = \|\mathbf{x}_u\| \quad (7.105)$$

The distance between the i th GPS satellite and user is expressed by

$$D_i = \|\mathbf{D}_i\| = \|\mathbf{x}_i - \mathbf{x}_u\| \quad (7.106)$$

The vector of pseudorange measurement to the n GPS satellite is given by

$$\mathbf{p} = \mathbf{D} - \mathbf{I}bc \quad (7.107)$$

where $\mathbf{I} = n \times 1$ column vector, b is the user's clock bias with respect to the GPS time scale, and c is the speed of light. The vector of range differences is formed by subtracting the first pseudorange from the others

$$\mathbf{d} = [\mathbf{z}] \mathbf{p} \quad (7.108)$$

where $[\mathbf{z}] = [-\mathbf{I}]$, and each component has the dimension, $\mathbf{d} \in \Re^{n-1}$, $[\mathbf{z}] \in \Re^{(n-1) \times n}$. The i th range difference can be obtained by

$$d_i = D_i - D_1 \quad (7.109)$$

With the definitions in the following

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}, \quad and \quad \boldsymbol{\gamma} = \begin{bmatrix} d_2^2 - r_2^2 \\ \vdots \\ d_n^2 - r_n^2 \end{bmatrix} \quad (7.110)$$

where $r_i = \|\mathbf{x}_i\|$, $r_u = \|\mathbf{x}_u\|$. The user position solution can be obtained for a non-singular $\mathbf{\Gamma}$ (More details in Ref. 119)

$$\mathbf{x}_u = -\mathbf{\Gamma}^{-1} \left(\frac{\boldsymbol{\gamma}}{2} + \mathbf{d}r_u \right) \quad (7.111)$$

where the value of r_u is computed from

$$r_u^\pm = \frac{(\mathbf{d}^T \mathbf{\Gamma}^{-T} \mathbf{\Gamma}^{-1} \boldsymbol{\gamma}) \pm \left[(\mathbf{d}^T \mathbf{\Gamma}^{-T} \mathbf{\Gamma}^{-1} \boldsymbol{\gamma})^2 + (\boldsymbol{\gamma}^T \mathbf{\Gamma}^{-T} \mathbf{\Gamma}^{-1} \boldsymbol{\gamma}) (1 - \mathbf{d}^T \mathbf{\Gamma}^{-T} \mathbf{\Gamma}^{-1} \mathbf{d}) \right]^{1/2}}{2(1 - \mathbf{d}^T \mathbf{\Gamma}^{-T} \mathbf{\Gamma}^{-1} \mathbf{d})} \quad (7.112)$$

If $\mathbf{d}^T \mathbf{\Gamma}^{-T} \mathbf{\Gamma}^{-1} \mathbf{d} = 1$, then

$$r_u = \frac{-\boldsymbol{\gamma}^T \mathbf{\Gamma}^{-T} \mathbf{\Gamma}^{-1} \boldsymbol{\gamma}}{4\mathbf{d}^T \mathbf{\Gamma}^{-T} \mathbf{\Gamma}^{-1} \boldsymbol{\gamma}} \quad (7.113)$$

Substitution of Eq. (7.112) back into Eq. (7.111) gives the two user positions, \mathbf{x}_u^+ and \mathbf{x}_u^- . One, both, or neither of these potential user positions may be real and yield a positive r_u . Once the user position is known, the clock bias can be obtained from Eq. (7.107). The conditions for existence and uniqueness of the user solution with the condition are discussed in Ref. 119. However, it is pointed out that for users in low Earth orbits, the user position solution is typically unique, yielding one positive real root in Eq. (7.112).

To eliminate the ambiguity problem in the solution r_u , a redundant pseudorange measurement is used in determining the user's position. Since n is now five, $n = m + 2$, the resulting $\mathbf{\Gamma}$ matrix is now 4×3 , and the generalized inverse $\mathbf{\Gamma}^*$ is used in place of $\mathbf{\Gamma}^{-1}$ in Eqs. (7.112) and (7.111)

$$\mathbf{\Gamma}^* = (\mathbf{\Gamma}^T \mathbf{\Gamma})^{-1} \mathbf{\Gamma}^T \quad (7.114)$$

(2) *Velocity and Clock Drift Solution*

In order to complete the description of the user vehicle's state, velocity and clock drift must also be determined, but the solution given in Ref. 119 only considered user position and clock bias. Referring to Refs. 119 and 120, the user velocity and clock drift solutions are described in the following. Assuming that the user satellite's GPS receiver measures the beat frequency from the incoming Doppler shifted signal as well as pseudorange, a solution for velocity and clock drift can be derived. The pseudorange to the i th GPS satellite can be written

$$p_i = \sqrt{(x_{s,i} - x_u)^2 + (y_{s,i} - y_u)^2 + (z_{s,i} - z_u)^2} - bc \quad (7.115)$$

where the subscript "s" denotes the coordinates of the GPS satellite, and the subscript "u" denotes the coordinates of the user satellite. Taking the time derivative yields

$$\dot{p}_i = \frac{1}{\rho_i} [(x_{s,i} - x_u)(\dot{x}_{s,i} - \dot{x}_u) + (y_{s,i} - y_u)(\dot{y}_{s,i} - \dot{y}_u) + (z_{s,i} - z_u)(\dot{z}_{s,i} - \dot{z}_u)] - \dot{bc} \quad (7.116)$$

Rearranging results in

$$\Delta x_i \dot{x}_{s,i} + \Delta y_i \dot{y}_{s,i} + \Delta z_i \dot{z}_{s,i} - \rho_i \dot{p}_i = \Delta x_i \dot{x}_u + \Delta y_i \dot{y}_u + \Delta z_i \dot{z}_u + \rho_i \dot{bc} \quad (7.117)$$

where $\Delta x_i = (x_{s,i} - x_u)$, $\Delta y_i = (y_{s,i} - y_u)$, and $\Delta z_i = (z_{s,i} - z_u)$. Then, the n

equations can be rewritten in form of matrix and matrix

$$\begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 & \rho_1 c \\ \Delta x_2 & \Delta y_2 & \Delta z_2 & \rho_2 c \\ \vdots & \vdots & \vdots & \vdots \\ \Delta x_n & \Delta y_n & \Delta z_n & \rho_n c \end{bmatrix} \begin{bmatrix} \dot{x}_u \\ \dot{y}_u \\ \dot{z}_u \\ \dot{b} \end{bmatrix} = \begin{bmatrix} \Delta x_1 \dot{x}_{s,1} + \Delta y_1 \dot{y}_{s,1} + \Delta z_1 \dot{z}_{s,1} - \rho_1 \dot{p}_1 \\ \Delta x_2 \dot{x}_{s,2} + \Delta y_2 \dot{y}_{s,2} + \Delta z_2 \dot{z}_{s,2} - \rho_2 \dot{p}_2 \\ \vdots \\ \Delta x_n \dot{x}_{s,n} + \Delta y_n \dot{y}_{s,n} + \Delta z_n \dot{z}_{s,n} - \rho_n \dot{p}_n \end{bmatrix} \quad (7.118)$$

The positions and velocities of the GPS satellites are known from their ephemeris message, the position of the user is known from the geometric solution given by Eqs. (7.112) and (7.111). If the range rates from the GPS satellites are known from the beat frequency measurements, the vector/matrix equation above can be solved for the unknown user velocity and clock drift vector. If $n > 4$, the system is over determined, the generalized inverse is used to obtain the best least-squares solution.

7.2.3.2 Least-Squares Solution

In this section, the navigation solution based on the least-squares method is illustrated by using the GPS pseudorange and range range rate measurements at a single time.

For each satellite tracked by the receiver, the measurement equations should be linearized around the prediction $\hat{\mathbf{x}}_{k|k-1}$, then the vector/matrix equation is formed

$$\Delta \boldsymbol{\rho} = \mathbf{H} \Delta \mathbf{x} + \mathbf{v} \quad (7.119)$$

If n satellites are visible and $\mathbf{x} = [x, y, z, bc]^T$, then

$$\mathbf{H} = \begin{bmatrix} -h_1^x & -h_1^y & -h_1^z & 1 \\ -h_2^x & -h_2^y & -h_2^z & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -h_n^x & -h_n^y & -h_n^z & 1 \end{bmatrix} \quad (7.120)$$

$$\Delta\boldsymbol{\rho} = \begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_2 \\ \vdots \\ \Delta\rho_n \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_{\rho_1} \\ v_{\rho_2} \\ \vdots \\ v_{\rho_n} \end{bmatrix} \quad (7.121)$$

where $[-h_i^x, -h_i^y, -h_i^z]$ is the user-to-satellite- i LOS vector, and \mathbf{v} is the pseudorange measurement noise.

If the measurement error vector \mathbf{v} has zero mean and covariance \mathbf{R} , then the weighted least-squares (WLS) estimator yields

$$\Delta\hat{\mathbf{x}} = [\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{R}^{-1} \Delta\boldsymbol{\rho} \quad (7.122)$$

where the covariance of the user state is calculated by

$$\mathbf{P}_x = [\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \quad (7.123)$$

Note that the estimate correction $\Delta\hat{\mathbf{x}}$ is usually iterated until the change in the estimate is sufficiently smaller than a specified threshold value, and it requires more computational time than the algebraic geometric solution. The advantage of the solution from the LS estimator is that since the geometric matrix \mathbf{H} depends on the line-of-sight unit vector it is not very sensitive to errors in the observation position. Moreover, the covariance information is available in the LS estimator solution.

7.2.4 Dynamic Navigation Solution

Although the geometric state solution offers certain advantages, its performance is limited due to the effects of viewing geometry and measurement noise. Furthermore, it requires a minimum of four pseudorange and range rate measurements at each solution time since each solution is computed independently from the previous one. The

dynamic solution, on the other hand, propagates the estimated state forward from one measurement update time to the next by means of a system dynamic model. It does not necessarily require measurements from four GPS satellites, and can compute a state update with even a single measurement. However, estimation accuracy generally improves when more measurements are used per update. Hence, the dynamic solution is less affected by poor viewing geometry than the geometric solution.¹²⁰

7.2.4.1 State Noise Compensation Method

The dynamic state solution of the GPS navigation is based on the utilization of the extended Kalman filter due to the nonlinearity in the measurement equation. The basic state vector for the filtering algorithm in the GPS orbit navigation is defined by

$$\mathbf{x} = \left[x, y, z, \dot{x}, \dot{y}, \dot{z}, \mu, J_2, C_d, bc, \dot{bc} \right]^T \in \mathbb{R}^{11 \times 1} \quad (7.124)$$

where c is the speed of light, b is the user clock bias. The dynamic parameters, μ , J_2 , and C_d are included to allow adjustment of the dynamic model uncertainty and compensate for dynamic modeling errors. The true real-world dynamical equations of motion are assumed to have the following form

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a}_m(\mathbf{r}, \mathbf{v}, \mathbf{p}, t) + \mathbf{w}(t) \\ \dot{\mathbf{p}} &= \mathbf{0} \\ \dot{\mathbf{d}} &= \mathbf{0} \end{aligned} \quad (7.125)$$

where \mathbf{r} and \mathbf{v} are the position and velocity vectors, respectively, and \mathbf{a}_m is an acceleration function due to the two-body attraction, drag perturbation, and harmonic geopotential perturbing acceleration. The constant parameter \mathbf{p} is composed of three dynamic parameters, and \mathbf{d} represents the clock bias and clock drift. $\mathbf{w}(t)$ is a white,

Gaussian process noise which presents the effects of model errors with the properties

$$E \{ \mathbf{w}(t) \} = 0, \quad E \{ \mathbf{w}(t) \mathbf{w}^T(s) \} = \mathbf{Q}(t) \delta(t - s) \quad (7.126)$$

Then, the vector/matrix equation in terms of the force vector \mathbf{f} can be written as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{w}(t) \quad (7.127)$$

where $\mathbf{w}(t)$ has the 3×3 state noise covariance matrix $\mathbf{Q}(t)$ given by

$$\mathbf{Q}(t) = \sigma_w^2 [\mathbf{I}_{3 \times 3}] \quad (7.128)$$

(1) *State Transition Matrix*

The nominal or filter dynamic model in a vector/matrix form is represented by

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, t) \quad (7.129)$$

The expression for the state sensitivity matrix \mathbf{F} , defined as the partial derivative of the force vector \mathbf{f} with respect to the nominal state vector $\hat{\mathbf{x}}$, is represented by

$$\mathbf{F} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} & \frac{\partial \dot{x}}{\partial z} & \frac{\partial \dot{x}}{\partial \dot{x}} & \frac{\partial \dot{x}}{\partial \dot{y}} & \frac{\partial \dot{x}}{\partial \dot{z}} & \frac{\partial \dot{x}}{\partial \mu} & \frac{\partial \dot{x}}{\partial J_2} & \frac{\partial \dot{x}}{\partial C_d} & \frac{\partial \dot{x}}{\partial bc} & \frac{\partial \dot{x}}{\partial bc} \\ \frac{\partial \dot{y}}{\partial x} & \dots & & & & & & & & & \frac{\partial \dot{y}}{\partial bc} \\ \frac{\partial \dot{z}}{\partial x} & \dots & & & & & & & & & \frac{\partial \dot{z}}{\partial bc} \\ \frac{\partial \ddot{x}}{\partial x} & \dots & & & & & & & & & \frac{\partial \ddot{x}}{\partial bc} \\ \frac{\partial \ddot{y}}{\partial x} & \dots & & & & & & & & & \frac{\partial \ddot{y}}{\partial bc} \\ \frac{\partial \ddot{z}}{\partial x} & \dots & & & & & & & & & \frac{\partial \ddot{z}}{\partial bc} \\ \frac{\partial \ddot{\mu}}{\partial x} & \dots & & & & & & & & & \frac{\partial \ddot{\mu}}{\partial bc} \\ \frac{\partial \dot{J}_2}{\partial x} & \dots & & & & & & & & & \frac{\partial \dot{J}_2}{\partial bc} \\ \frac{\partial \dot{C}_d}{\partial x} & \dots & & & & & & & & & \frac{\partial \dot{C}_d}{\partial bc} \\ \frac{\partial \dot{bc}}{\partial x} & \dots & & & & & & & & & \frac{\partial \dot{bc}}{\partial bc} \\ \frac{\partial \ddot{bc}}{\partial x} & \frac{\partial \ddot{bc}}{\partial y} & \frac{\partial \ddot{bc}}{\partial z} & \frac{\partial \ddot{bc}}{\partial \dot{x}} & \frac{\partial \ddot{bc}}{\partial \dot{y}} & \frac{\partial \ddot{bc}}{\partial \dot{z}} & \frac{\partial \ddot{bc}}{\partial \mu} & \frac{\partial \ddot{bc}}{\partial J_2} & \frac{\partial \ddot{bc}}{\partial C_d} & \frac{\partial \ddot{bc}}{\partial bc} & \frac{\partial \ddot{bc}}{\partial bc} \end{bmatrix} \quad (7.130)$$

where the non-zero terms of this matrix are listed in Appendix A. The state transition matrix Φ is obtained by integrating the following matrix differential equation

$$\dot{\Phi}(t_k) = \mathbf{F}(t_k, t_0)\Phi(t_0), \quad \Phi(t_0) = \mathbf{I} \quad (7.131)$$

The covariance equation is obtained from the definition of the propagated covariance matrix $\mathbf{P}(t_k)$ at the current time t_k conditioned on observations through t_0 in Eq. (6.35)

$$\mathbf{P}(t_k) = \Phi(t_k, t_0)\mathbf{P}(t_0)\Phi^T(t_k, t_0) + \mathbf{Q}(t_k) \quad (7.132)$$

where $\mathbf{P}(t_0) = E\{\delta\mathbf{x}(t_0)\delta\mathbf{x}^T(t_0)\}$, and $\mathbf{Q}(t_k)$ is computed in the following.

(2) Process Noise Covariance Matrix

The discrete-time process noise covariance matrix for position, velocity, and pa-

rameters can be constructed

$$\mathbf{Q}_w(t_k) = \begin{bmatrix} \frac{\Delta t^3 \sigma_w^2}{3} \mathbf{I}_{3 \times 3} & \frac{\Delta t^2 \sigma_w^2}{2} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \frac{\Delta t^2 \sigma_w^2}{2} \mathbf{I}_{3 \times 3} & \Delta t \sigma_w^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (7.133)$$

Now, suppose that the user clock drift is modeled as a constant term plus a random walk. The time derivative is given by

$$\ddot{b}(t) = u_d(t) \quad (7.134)$$

where $u_d(t)$ is a zero mean, uncorrelated, stationary, Gaussian process with variance σ_u^2 . Then, the solution is obtained by

$$\dot{b}(t) = \dot{b}_0 + \int_{t_0}^t u_d(T) dT \quad (7.135)$$

The stochastic integral is defined by

$$D(t) \equiv \int_{t_0}^t u_d(T) dT \quad (7.136)$$

Then, an approximate, discrete solution to the stochastic integral is given by

$$D_k = u_{d,k} \sigma_d \sqrt{(t - t_0)} \quad (7.137)$$

where $u_{d,k}$ is a discrete Gaussian sequence with a zero mean and unity variance. Then, the stochastic model of the clock drift is calculated by

$$\dot{b}(t) = \dot{b}_0 + u_{d,k} \sigma_d \sqrt{(t - t_0)} \quad (7.138)$$

The contribution to the process noise matrix is given by the variance

$$q_d = E \{ D_k^2 \} = \sigma_d^2 (t - t_0) \quad (7.139)$$

The time derivative of the user clock bias is modeled as a constant, deterministic drift plus a zero mean, uncorrelated, stationary, Gaussian noise component

$$\dot{b}_b = \dot{b}_{\text{det}} + u_b(t) \quad (7.140)$$

where $u_b(t)$ is a zero mean, uncorrelated, stationary, Gaussian process with variance σ_d^2 . Then, an approximate, discrete solution to the corresponding stochastic integral is given by

$$C_k = u_{b,k} \sigma_b \sqrt{(t - t_0)} \quad (7.141)$$

where $u_{b,k}$ is a discrete Gaussian sequence with a zero mean and unity variance. The resulting clock bias stochastic model is obtained by

$$b(t) = b_0 + \dot{b}_{\text{det}}(t - t_0) + u_{b,k} \sigma_b \sqrt{(t - t_0)} \quad (7.142)$$

The contribution to the process noise matrix is given by the variance

$$q_b = E \{ C_k^2 \} = \sigma_b^2 (t - t_0) \quad (7.143)$$

Finally, the discrete-time process noise covariance matrix $\mathbf{Q}(t_k)$ due to position, velocity, three parameters, clock bias, and clock drift is expressed by

$$\mathbf{Q}(t_k) = \begin{bmatrix} \frac{\Delta t^3 \sigma_w^2}{3} \mathbf{I}_{3 \times 3} & \frac{\Delta t^2 \sigma_w^2}{2} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 0 & 0 \\ \frac{\Delta t^2 \sigma_w^2}{2} \mathbf{I}_{3 \times 3} & \Delta t \sigma_w^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 0 & 0 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 0 & 0 \\ 0 & 0 & 0 & q_b & 0 \\ 0 & 0 & 0 & 0 & q_d \end{bmatrix} \quad (7.144)$$

where $\Delta t = t - t_0$.

(3) *Linearized Measurement Equation*

The state estimation filter can use both pseudorange and range rate measure-

ments. The observation and state relationships for these are given by

$$y_p = \rho - bc \quad (7.145)$$

$$y_d = \frac{1}{\rho} [(x_s - x_u)(\dot{x}_s - \dot{x}_u) + (y_s - y_u)(\dot{y}_s - \dot{y}_u) + (z_s - z_u)(\dot{z}_s - \dot{z}_u)] - \dot{bc} \quad (7.146)$$

where the p subscript denotes pseudorange, d denotes range rate (Doppler), s denotes the GPS satellite, u denotes the user satellite, and ρ is the range between the GPS and user satellites. The corresponding partial derivatives of the observation equation with respect to the state are expressed by

$$\mathbf{H}_p = \frac{\partial y_p}{\partial \mathbf{x}} = [h_{1,p}, h_{2,p}, h_{3,p}, 0, 0, 0, 0, 0, 0, -1, 0] \quad (7.147)$$

$$\mathbf{H}_d = \frac{\partial y_d}{\partial \mathbf{x}} = [h_{1,d}, h_{2,d}, h_{3,d}, h_{4,d}, h_{5,d}, h_{6,d}, 0, 0, 0, 0, -1] \quad (7.148)$$

where each component is given by

$$\begin{aligned} h_{1,p} &= \frac{-(x_s - x_u)}{\rho} \\ h_{2,p} &= \frac{-(y_s - y_u)}{\rho} \\ h_{3,p} &= \frac{-(z_s - z_u)}{\rho} \end{aligned} \quad (7.149)$$

$$\begin{aligned} h_{1,d} &= \frac{(x_s - x_u)^2(\dot{x}_s - \dot{x}_u)}{\rho^3} + \frac{(x_s - x_u)(y_s - y_u)(\dot{y}_s - \dot{y}_u)}{\rho^3} + \frac{(x_s - x_u)(z_s - z_u)(\dot{z}_s - \dot{z}_u)}{\rho^3} - \frac{(\dot{x}_s - \dot{x}_u)}{\rho} \\ h_{2,d} &= \frac{(y_s - y_u)^2(\dot{y}_s - \dot{y}_u)}{\rho^3} + \frac{(y_s - y_u)(x_s - x_u)(\dot{x}_s - \dot{x}_u)}{\rho^3} + \frac{(y_s - y_u)(z_s - z_u)(\dot{z}_s - \dot{z}_u)}{\rho^3} - \frac{(\dot{y}_s - \dot{y}_u)}{\rho} \\ h_{3,d} &= \frac{(z_s - z_u)^2(\dot{z}_s - \dot{z}_u)}{\rho^3} + \frac{(z_s - z_u)(x_s - x_u)(\dot{x}_s - \dot{x}_u)}{\rho^3} + \frac{(z_s - z_u)(y_s - y_u)(\dot{y}_s - \dot{y}_u)}{\rho^3} - \frac{(\dot{z}_s - \dot{z}_u)}{\rho} \\ h_{4,d} &= h_{1,p} \\ h_{5,d} &= h_{2,p} \\ h_{6,d} &= h_{3,p} \end{aligned} \quad (7.150)$$

Note that in the case of multiple measurements at each update time, the rows of the total sensitive partial matrix \mathbf{H} consist of the appropriate \mathbf{H}_p and/or \mathbf{H}_d row matrices. If m visible satellites are available, and n state vectors are estimated, then

the total sensitive partial matrix is formulated by

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{p,1} \\ \mathbf{H}_{d,1} \\ \vdots \\ \mathbf{H}_{p,m} \\ \mathbf{H}_{d,m} \end{bmatrix} = \Re^{2m \times n} \quad (7.151)$$

Usually, using pseudorange and range rate measurements together provides better performance than using pseudorange measurements alone.

7.2.4.2 Dynamic Model Compensation Method

The deterministic parameter of the stochastic acceleration can be added to the state vector and estimated in the conventional way to provide some measure of compensation for accelerations that are unknown and unmodeled in the dynamic model. Since the parameter τ is a deterministic constant value in the acceleration function, it can be added to the state vector and estimated for an optimal value. Assuming that τ is the same for all three axes, the augmented state vector contains 15 parameters

$$\mathbf{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \mu & J_2 & C_d & bc & \dot{bc} & w_x & w_y & w_z & \tau \end{bmatrix}^T \in \Re^{15 \times 1} \quad (7.152)$$

The deterministic parts of the stochastic accelerations contribute terms to the dynamic expressions for position and velocity.

(1) *State Transition Matrix*

The state transition matrix $\Phi(t_k, t_0)$ and process noise matrix \mathbf{Q}_k for the DMC formulation are obtained in Eq. (6.29) and Eq. (6.64), respectively. Since the augmented state vector \mathbf{x} includes the system parameters, user clock bias, and drift terms, then the complete state transition matrix $\Phi(t_k, t_0) \in \Re^{15 \times 15}$ has the following

expression

$$\Phi(t_k, t_0) = \begin{bmatrix} [\Phi_{filter}]_{11 \times 11} & [\Phi_w]_{11 \times 3} & [\Phi_\tau]_{11 \times 1} \\ \mathbf{0}_{3 \times 11} & [\mathbf{M}]_{3 \times 3} & [\mathbf{N}]_{3 \times 1} \\ \mathbf{0}_{1 \times 11} & \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (7.153)$$

where $[\Phi_{filter}]$ is the state transition matrix associated with the 11 states without the DMC which include the position, velocity, system parameters, user clock bias, and clock drift, $[\Phi_w]$ is the transition matrix of the 11 basic states with respect to the stochastic accelerations, $[\Phi_\tau]$ is the transition matrix of the 11 basic states with respect to the correlation time τ . $[\mathbf{M}]$ is the transition matrix of the stochastic accelerations with respect to themselves, which is computed in Eq. (6.25), and $[\mathbf{N}]$ is the transition matrix of the stochastic accelerations with respect to the correlation time given in Eq. (6.31), respectively

$$\mathbf{M} = \begin{bmatrix} e^{-(t-t_0)/\tau} & 0 & 0 \\ 0 & e^{-(t-t_0)/\tau} & 0 \\ 0 & 0 & e^{-(t-t_0)/\tau} \end{bmatrix} \quad (7.154)$$

$$\mathbf{N} = \begin{bmatrix} \frac{w_{x0}}{\tau^2}(t-t_0)e^{-(t-t_0)/\tau} \\ \frac{w_{y0}}{\tau^2}(t-t_0)e^{-(t-t_0)/\tau} \\ \frac{w_{z0}}{\tau^2}(t-t_0)e^{-(t-t_0)/\tau} \end{bmatrix} \quad (7.155)$$

The elements of Φ_w , and Φ_τ , can be found analytically by

$$\Phi_w = \begin{bmatrix} \phi_{wp} & 0 & 0 \\ 0 & \phi_{wp} & 0 \\ 0 & 0 & \phi_{wp} \\ \phi_{wv} & 0 & 0 \\ 0 & \phi_{wv} & 0 \\ 0 & 0 & \phi_{wv} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (7.156)$$

$$\Phi_\tau = \begin{bmatrix} 2\tau w_{x_0} (e^{-(t-t_0)/\tau} - 1) + w_{x_0} (t - t_0) (e^{-(t-t_0)/\tau} + 1) \\ 2\tau w_{y_0} (e^{-(t-t_0)/\tau} - 1) + w_{y_0} (t - t_0) (e^{-(t-t_0)/\tau} + 1) \\ 2\tau w_{z_0} (e^{-(t-t_0)/\tau} - 1) + w_{z_0} (t - t_0) (e^{-(t-t_0)/\tau} + 1) \\ w_{x_0} (e^{-(t-t_0)/\tau} - 1) - \frac{w_{y_0}}{\tau} (t - t_0) e^{-(t-t_0)/\tau} \\ w_{y_0} (e^{-(t-t_0)/\tau} - 1) - \frac{w_{y_0}}{\tau} (t - t_0) e^{-(t-t_0)/\tau} \\ w_{z_0} (e^{-(t-t_0)/\tau} - 1) - \frac{w_{z_0}}{\tau} (t - t_0) e^{-(t-t_0)/\tau} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (7.157)$$

where

$$\phi_{wp} = \tau^2 (e^{-(t-t_0)/\tau} - 1) + \tau (t - t_0) \quad (7.158)$$

$$\phi_{uv} = \tau (1 - e^{-(t-t_0)/\tau}) \quad (7.159)$$

(2) *Process Noise Matrix*

The process noise covariance matrix for the DMC was expressed by Eq. (6.64) in the previous chapter, which is associated with the covariance elements due to the position, velocity, and three stochastic accelerations, and correlation time terms. The discrete-time process noise covariance terms due to the system parameters, user clock bias, and clock drift were given in Eq. (7.144) from the previous SNC section. Thus, the complete process noise covariance terms due to position, velocity, three system parameters, clock bias, clock drift, three stochastic accelerations, and correlation time can be constructed by

$$\mathbf{Q}_k = \begin{bmatrix} \left[\frac{\Delta t^4}{4} \mathbf{\Lambda} \right]_{3 \times 3} & \left[\frac{\Delta t^3}{2} \mathbf{\Lambda} \right]_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \left[\frac{\Delta t^2}{2} \mathbf{\Lambda} \right] & \mathbf{0}_{3 \times 1} \\ \left[\frac{\Delta t^3}{2} \mathbf{\Lambda} \right]_{3 \times 3} & [\Delta t^2 \mathbf{\Lambda}]_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & [\Delta t \mathbf{\Lambda}] & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & [q_b]_{1 \times 1} & 0_{1 \times 1} & \mathbf{0}_{1 \times 3} & 0_{1 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0_{1 \times 1} & [q_d]_{1 \times 1} & \mathbf{0}_{1 \times 3} & 0_{1 \times 1} \\ \left[\frac{\Delta t^2}{2} \mathbf{\Lambda} \right]_{3 \times 3} & [\Delta t \mathbf{\Lambda}]_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & [\mathbf{\Lambda}]_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0_{1 \times 1} & 0_{1 \times 1} & \mathbf{0}_{1 \times 3} & [q_\tau]_{1 \times 1} \end{bmatrix} \quad (7.160)$$

where $\Delta t = t - t_0$.

(iii) *Linearized Measurement Equation*

Using both pseudorange and range rate measurements, the corresponding partial derivatives of the observation equation with respect to the augmented state $\mathbf{x} \in \mathbb{R}^{15 \times 1}$ are expressed by

$$\begin{aligned} \mathbf{H}_p &= [h_{1,p}, h_{2,p}, h_{3,p}, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0] \\ \mathbf{H}_d &= [h_{1,d}, h_{2,d}, h_{3,d}, h_{4,d}, h_{5,d}, h_{6,d}, 0, 0, 0, 0, -1, 0, 0, 0, 0] \end{aligned} \quad (7.161)$$

where each component of $h_{i,p}$ and $h_{i,d}$ can be found in Eq. (7.149) and Eq. (7.150), respectively. For the case of multiple measurements at each update time, the rows of the total sensitive partial matrix \mathbf{H} consist of the appropriate \mathbf{H}_p and/or \mathbf{H}_d row matrices. If m visible satellites are available, and n state vector are estimated, then the total sensitive partial matrix is formulated by

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{p,1} \\ \mathbf{H}_{d,1} \\ \vdots \\ \mathbf{H}_{p,m} \\ \mathbf{H}_{d,m} \end{bmatrix} = \Re^{2m \times n} \quad (7.162)$$

CHAPTER VIII

SIMULATION RESULTS

8.1 Performance Criteria

The criteria for judging the performance of the proposed nonlinear/adaptive filters are the magnitude of the residuals and their statistics. If the measurement residuals or the state estimation errors are sufficiently small and consistent with their statistics, then the filter is trusted to be operating consistently. In other words, the most common way for testing the consistency of the filtering results is to depict the estimated state errors with the 3-*sigma* bound given by $\pm 3\sqrt{\mathbf{P}_k^+}$. If the errors lie within the bound, the estimation result is believed to be consistent and reliable. Instead of the state estimation errors, the measurement innovation vector can also be used for the filter performance analysis. If the measurement residuals lie within the 2-*sigma* bound, $\pm 2\sqrt{\mathbf{P}_{k+1}^{vv}}$, it indicates the 95% confidence of the estimation results.

In this study, the average root mean square (RMS) value is also used for the quantitative performance comparison of the proposed filters. The RMS error is defined by

$$\varepsilon_i(k) = \sqrt{\frac{1}{N} \sum_{j=1}^N [x_{i,j}(k) - \hat{x}_{i,j}(k)]^2}$$

where N is the number of Monte-Carlo runs, j denotes the j th simulation run, and i represents the i th component of the state vector $\mathbf{x}(k)$ and its current estimate vector $\hat{\mathbf{x}}(k)$.

However, it is not enough to strictly judge the optimality of the filtering algorithms. Thus, three additional methods for a quantitative analysis of the filtering optimality are proposed. Two methods are based on the *mean-square error* (MSE) concept and the third is the *innovation-based whiteness*.

8.1.1 Posterior Cramér-Rao Lower Bound

A closed form analytic solution to this optimal filtering problem is not tractable in general, and in practical applications nonlinear filtering is represented by an approximated and suboptimal filtering algorithm. Despite the absence of a closed form solution, the best achievable error performance for nonlinear filtering can be assessed by considering lower bounds on the mean squared error (MSE). Lower bounds give an indication of performance limitations, and it can be used to determine whether imposed performance is realistic or not.

As explained in Chapter 4, a commonly used lower bound is the Cramér-Rao lower bound (CRLB), which is defined to be the inverse of the Fisher information matrix and provides a lower bound on the performance of any unbiased estimator of an unknown parameter vector. This provides a powerful tool that has been used to assess the performance of unbiased estimators of parameters for deterministic dynamical motion. In the case of uncertain dynamical motion the posterior Cramér-Rao lower bound (PCRLB) has been used to determine performance bounds for recursive Bayesian estimators of the uncertain target state.⁶⁸ Determining PCRLBs represents an even more challenging problem. The reason is that for calculating the Fisher information matrix, it is necessary to consider both the effect of measurement uncertainty as well as uncertainty in the random state.

The general PCRLB formulation for the nonlinear Gaussian filtering problem is given by taking the inverse of the posterior Fisher information matrix equation (see Ref. 84 for details). The Riccati-like recursive Fisher information matrix equation is expressed by

$$\mathbf{J}_{k+1} = \mathbf{D}_k^{22} - \mathbf{D}_k^{21} (\mathbf{J}_k + \mathbf{D}_k^{11})^{-1} \mathbf{D}_k^{12} \quad (8.1)$$

where

$$\mathbf{D}_k^{11} = E \left\{ [\nabla_{\mathbf{x}_k} \mathbf{f}_k^T(\mathbf{x}_k)] \mathbf{Q}_k^{-1} [\nabla_{\mathbf{x}_k} \mathbf{f}_k^T(\mathbf{x}_k)]^T \right\} \quad (8.2)$$

$$\mathbf{D}_k^{12} = -E \left\{ [\nabla_{\mathbf{x}_k} \mathbf{f}_k^T(\mathbf{x}_k)] \mathbf{Q}_k^{-1} \right\} = (\mathbf{D}_k^{21})^T \quad (8.3)$$

$$\mathbf{D}_k^{22} = \mathbf{Q}_k^{-1} + E \left\{ [\nabla_{\mathbf{x}_{k+1}} \mathbf{h}_{k+1}^T(\mathbf{x}_{k+1})] \mathbf{R}_{k+1}^{-1} [\nabla_{\mathbf{x}_{k+1}} \mathbf{h}_{k+1}^T(\mathbf{x}_{k+1})]^T \right\} \quad (8.4)$$

8.1.2 Optimality Analysis

The EKF assumes that the deviations of the model state trajectory from the actual trajectory are small. However, errors due to the truncated terms in the linearization of the dynamic and observation models can lead to bias estimation and instability. Many methods have been developed to mitigate the nonlinearity effects. One of the methods that directly estimates the bias term by augmenting it into the extended state vector is simply a joint estimation through the extended Kalman filter.¹²¹ In this paper, however, we focus on measuring consistency and/or biasness indirectly instead of directly estimating biases. For this analysis a simple, but efficient, methodology is introduced.

Optimality of the nonlinear filtering guarantees that the state estimation errors from the actual trajectory are small and the model estimates are unbiased.¹²² However, truncated errors due to the neglected terms in the approximation of the nonlinear models cause biased estimation leading to non-Gaussian *a posteriori* densities. Thus, measuring the nonlinearity can be an alternative way for checking the filtering optimality.

First, the mean square error (MSE) of the estimate $\hat{\mathbf{x}}_k$, is defined as

$$MSE \{ \mathbf{e}_k \} \equiv E \left\{ [\mathbf{e}_k - E \{ \mathbf{e}_k \}] [\mathbf{e}_k - E \{ \mathbf{e}_k \}]^T \right\} \quad (8.5)$$

where the error vector $\mathbf{e}_k \in \Re^{n \times 1}$ is given by the difference between the true and

estimated (or predicted) state vectors

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k \quad (8.6)$$

If the estimate is biased, then the mean square error is written by²⁷

$$MSE \{ \mathbf{e}_k \} = \mathbf{P}_k + \mathbf{b}_k \mathbf{b}_k^T \quad (8.7)$$

where \mathbf{b}_k is the bias obtained by taking the expected value of the estimate error, given by

$$\mathbf{b}_k \equiv E \{ \mathbf{e}_k \} \quad (8.8)$$

When the estimate $\hat{\mathbf{x}}_k$ is not biased, the expected value of the estimate errors is determined from the covariance

$$E \{ \mathbf{e}_k \mathbf{e}_k^T \} = \mathbf{P}_k \quad (8.9)$$

Note that for unbiased estimate cases the MSE is reduced to the consistency analysis test.¹²¹ Now, with the unbiased estimate assumption, the expectation of the quadratic value is introduced by

$$E \{ \mathbf{e}_k^T \mathbf{A} \mathbf{e}_k \} = tr [\mathbf{A} E \{ \mathbf{e}_k \mathbf{e}_k^T \}] = tr [E \{ \mathbf{e}_k \mathbf{e}_k^T \} \mathbf{A}] \quad (8.10)$$

If the matrix \mathbf{A} is the inverse state covariance matrix \mathbf{P}_k^{-1} , then the expected value reduces to

$$E \{ \mathbf{e}_k^T \mathbf{A} \mathbf{e}_k \} = tr [\mathbf{P}_k^{-1} \mathbf{P}_k] = tr [\mathbf{I}_n] = n \quad (8.11)$$

which means that the expected normalized error squared should be equal to the dimension of the state error vector n . This fact can be utilized to check a degree of nonlinearity in the prediction and update of the state and covariance used in various

filters and estimators. The optimality index τ_k is defined by

$$\tau_k \equiv \frac{1}{\sqrt{n}} \left\{ [\mathbf{x}_k - \hat{\mathbf{x}}_k]^T \mathbf{P}_k^{-1} [\mathbf{x}_k - \hat{\mathbf{x}}_k] \right\}^{1/2} \quad (8.12)$$

where \mathbf{P}_k^{-1} is the updated covariance matrix and $\hat{\mathbf{x}}_k$ is the updated state vector. If the value of the optimality index τ_k is much greater than unity ($\tau_k \gg 1$) then the effect of nonlinearity is believed to be severe, but in contrast if τ_k is close to unity, it is trusted that the nonlinearity is small and ignorable. Simultaneously, those conditions can tell the optimality of the performance of the nonlinear filters. If nonlinear filters produce unbiased estimates with consistent covariances, then the estimated results should make the optimality index close to unity ($\tau_k \simeq 1$).

Note that when τ_k becomes much larger than unity, the process noise matrix \mathbf{Q}_k is increased so that the optimality index can reduced to a reasonable value close to unity. Therefore, this method has the combined feature of optimality analysis and adaptive model compensation. This method is utilized for the development of the adaptive filtering approach proposed by Jazwinski.⁹⁰

8.1.3 Whiteness Test

An innovation-based consistency test is based on the fact that the quantities available for judging filter performance are the magnitude of the residuals (innovation vector) and their expected statistics (zero-mean and whiteness).^{1,90} Therefore, the statistical consistency of the measurement innovation vector is vital for verifying filter optimality. The whiteness test of the measurement innovation \mathbf{v}_k , which are j steps apart, from a single run is derived by computing the time-averaged autocorrelation²⁷

$$\bar{\rho}_l(j) = \frac{1}{\sqrt{m}} \sum_{k=1}^N v_l(k) v_l(k+j) \left[\sum_{k=1}^N v_l(k)^2 \sum_{k=1}^N v_l(k+j)^2 \right]^{-1/2} \quad \text{for } l = 1, \dots, m \quad (8.13)$$

where m is the dimension of the innovation vector, and N is the number of the observation data points. If the innovations are zero-mean and white, then $\bar{\rho}_l(j)$ becomes zero mean with variance $1/N$ for N large enough.

8.2 Orbit and Parameter Estimation

Two simulation examples are illustrated; The first example is the estimation of the spacecraft in a low Earth orbit with application of the presented nonlinear filtering methods, but there is no adaptive filtering or model compensation approaches in the first one. The other example shows the advantages of the adaptive process noise compensation, and the performance of the non-adaptive nonlinear filters and the adaptive nonlinear filters are compared. The true trajectory, observations, and filter dynamic model are all the same in the two examples, but the applied estimation methods are different.

8.2.1 Generation of True Trajectory

The satellite under consideration has the following orbit parameters in Table 8.1 at epoch, defined to be simulation time zero. The perturbing acceleration function

Table 8.1 Classical Orbital Elements

Orbital Elements	Values
a	6778.136 (km)
e	1.0×10^{-5}
i	51.6 (deg)
Ω	25.0 (deg)
ω	30.0 (deg)
M	0.0 (deg)

consists of a gravitational component due to the J_2 zonal perturbation, and a drag

component that is described by the exponential density function. The unmodeled or neglected accelerations are compensated by adding process noises into the deterministic system equation such that the true system model has the following form expressed by first order differential state forms

$$\dot{\mathbf{r}} = \mathbf{v} \quad (8.14)$$

$$\dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_g + \mathbf{a}_d + \mathbf{w}(t) \quad (8.15)$$

where \mathbf{r} and \mathbf{v} are the position and velocity vectors, respectively, and \mathbf{a}_g is an acceleration function due to the J_2 zonal harmonics, and \mathbf{a}_d is the drag perturbation. $\mathbf{w}(t) \in \Re^{3 \times 1}$ is a white, Gaussian process noise which presents the effects of model errors with the properties

$$E\{\mathbf{w}(t)\} = 0, \quad E\{\mathbf{w}(t)\mathbf{w}^T(s)\} = \mathbf{Q}(t)\delta(t-s) \quad (8.16)$$

The simulated true orbit for the user satellite is generated by numerical integration of the acceleration function by means of a fourth-order Runge-Kutta method.

8.2.2 Generation of Observations

A ground-based radar tracking station was used to take observations and the location of the sensor was selected to be Eglin Air Force Base whose location is at 30.2316° latitude and 86.2347° longitude. An observation consists of range, azimuth, and elevation angles and the measurement errors were considered to be Gaussian random processes with zero means and variances

$$\sigma_{range} = 25.0 \text{ m}, \quad \sigma_{azimuth} = 0.015^\circ, \quad \sigma_{elevation} = 0.015^\circ$$

The observation track length is 120 seconds with observations every five seconds, and each observation consists of a range, azimuth, and elevation measurement. For

the generation of the observations, Eq. (7.30), Eq. (7.33), and Eq. (7.34) were used, respectively.

In order to determine the satellite visibility tracking capability analysis, an elevation cut-off angle is set with the threshold value 15° . The overall procedures were already described in Fig. 7.4. Tracks available for observations are separated by 12, and 24 hour time gaps, thus estimated outputs from the first track are propagated up to the next available measurements. Herein, real-time OD is defined as completing the calculations required to perform the OD measurement update prior to acquiring the next measurement.

8.2.3 Filter Dynamic Model

Based on the true system model in Eq. (8.14), the nominal or filter dynamic model in a vector/matrix form is represented by

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, t) \quad (8.17)$$

where $\hat{\mathbf{x}} \in \Re^{7 \times 1}$ consists of the estimated position, velocity, and drag coefficient components, respectively. This ephemeris generation is referred to as reference trajectory generation.

Once, the state transition matrix Φ is obtained by integrating the matrix differential equation in Eq. (7.84), the propagation of the state covariance matrix is calculated from the discrete-time covariance equation in Eq. (7.86). The discrete-time process noise covariance matrix for position, velocity, and drag coefficient is constructed by

$$\mathbf{Q}(t_k) = \begin{bmatrix} \frac{\Delta t^3 \sigma_w^2}{3} \mathbf{I}_{3 \times 3} & \frac{\Delta t^2 \sigma_w^2}{2} \mathbf{I}_{3 \times 3} & \mathbf{0}_{1 \times 1} \\ \frac{\Delta t^2 \sigma_w^2}{2} \mathbf{I}_{3 \times 3} & \Delta t \sigma_w^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{1 \times 1} \\ \mathbf{0}_{1 \times 1} & \mathbf{0}_{1 \times 1} & \mathbf{0}_{1 \times 1} \end{bmatrix} \quad (8.18)$$

where the variance of the process noise σ_w^2 is obtained by selecting the variance of the process noise $\mathbf{w}(t)$. In continuous systems, propagation of the covariance matrix is computed by integrating the differential Lyapunov equation with an initial estimated covariance matrix.

8.2.4 Simulation Results and Performance Analysis

Two simulation examples are illustrated; one is the orbit estimation with the presented nonlinear filtering methods in the absence of adaptive filtering or model compensation methods. The other scenario utilizes the adaptive nonlinear filters, and the adaptive filters are compared with the standard nonlinear filters.

8.2.4.1 Nonlinear Estimation

In this section, the performance of the proposed nonlinear filters, the EKF, UKF, and DDF is demonstrated through simulation examples using the realistic system and observation models. Fig. 8.1 illustrates the orbit estimation strategy used in this study. For establishing accurate estimate conditions a few measurements are used to perform an initial guess of the state of the satellite. In this study, the Herrick-Gibbs algorithm,¹⁰⁴ which was explained in Eq. (7.47), is adopted to obtain the initial orbit estimate.

The *solve-for* state vector $\mathbf{x} \in \mathbb{R}^{6 \times 1}$ consists of the position, and velocity. The state vector for the UKF is augmented with the process noise terms, $\mathbf{x}^a = [\mathbf{r}^T \mathbf{v}^T \mathbf{w}^T]^T \in \mathbb{R}^{6+3}$. The parameters used in the UKF are the scaling factors associated with the scaled unscented transformation. $\beta = 2$ is set to capture the higher order (fourth) terms in the Taylor series expansion, κ provides an extra degree of freedom to fine tune the higher order moments of the approximation. If \mathbf{x} is a Gaussian distribution, then $\kappa = 3 - n$ is used for multi-dimensional systems, and $\alpha = 1/\sqrt{n}$ is chosen to

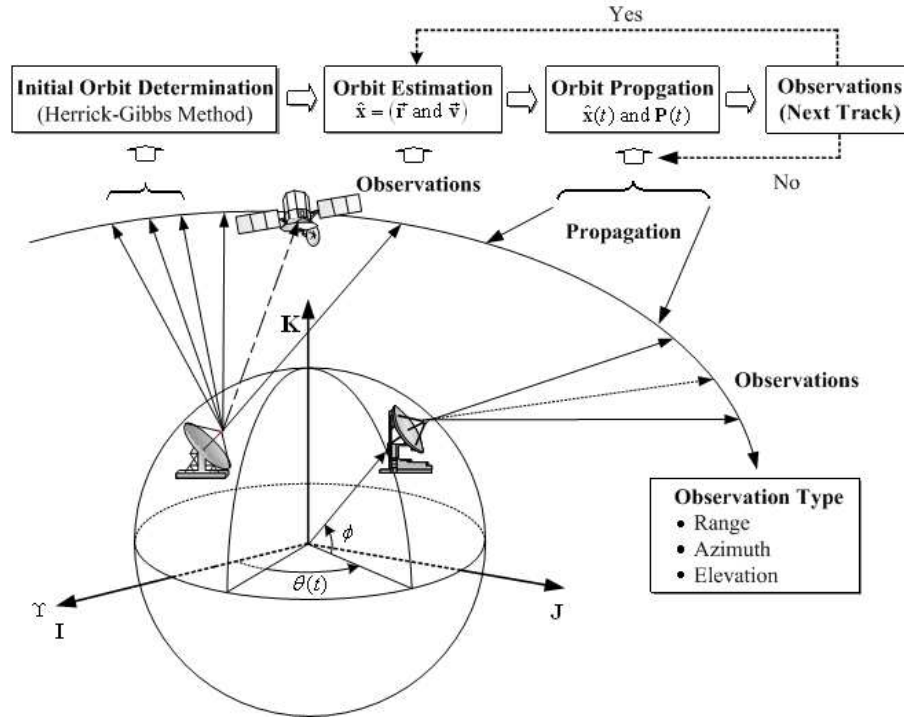


Fig. 8.1 Diagram for Orbit Estimation Strategy

make the sample distance independent of the state size. The interval length $h = \sqrt{3}$ is set for a Gaussian distribution in the DDF.

The initial covariance $\mathbf{P}_0 \in \mathbb{R}^{6 \times 6}$ used for the filters is assumed to be diagonal such that the position variances are 1 km^2 , the velocity variances are $5 \times 10^{-3} \text{ km}^2/\text{s}^2$.

$$\mathbf{P}_0 = \text{diag}([1 \ 1 \ 1 \ 5 \times 10^{-3} \ 5 \times 10^{-3} \ 5 \times 10^{-3}])$$

The process noise for the dynamic model errors needs to be added to the acceleration terms so that it can adjust the convergence properties. In this study, however, the value for $\mathbf{Q}(t)$ is set rather than adjusted in order to model the realistic environment as close as possible. For instance, the acceleration due to J_2 is approximately 10^{-5} km/sec^2 , and the truncated or ignored perturbing accelerations are roughly of order J_2^2 . Therefore, in the orbit scenario model, the process noise matrix takes the

values

$$\mathbf{Q}(t) = \text{diag} \left(\begin{bmatrix} 0 & 0 & 0 & 10^{-16} & 10^{-16} & 10^{-16} \end{bmatrix} \right)$$

Note that since the process noise covariance $\mathbf{Q}(t)$ comes from a continuous-time dynamical system model, it needs to be converted into the discrete-time form of the covariance \mathbf{Q}_k through an approximate numerical integration scheme.²⁷ In this work, the discrete-time process noise matrix introduced in Eq. (8.18) can be used such that the variance of the velocity variance component $\sigma_w^2 = 10^{-16}$ in the continuous-time system is substituted for the discrete-time one

$$\mathbf{Q}_k = \begin{bmatrix} \frac{\Delta t^3}{3} 10^{-16} [\mathbf{I}_{3 \times 3}] & \frac{\Delta t^2}{2} 10^{-16} [\mathbf{I}_{3 \times 3}] \\ \frac{\Delta t^2}{2} 10^{-16} [\mathbf{I}_{3 \times 3}] & \Delta t 10^{-16} [\mathbf{I}_{3 \times 3}] \end{bmatrix} \quad (8.19)$$

The simulation result in Fig. 8.2 shows the average magnitude of the position and velocity estimate errors generated by each filter through a Monte-Carlo simulation consisting of 30 runs. As can be seen, the advantage of the SPKFs over the EKF in this case is not obvious, which indicates that the effect of nonlinearity on the filters is negligible with the small initial state errors along with the small process noises over the short track length. This is the expected result because nonlinearity is negligible over the short arc, and setting $\mathbf{Q}(t)$ to zero in the filters should obtain theoretically similar results with the sequential least-squares estimation of the state and covariance. To verify the above results optimality or consistency is investigated by using the index τ_k in Figures 8.3 and 8.4. Fig. 8.3 shows the state estimation performance of the filters in terms of the value τ_k , where the error \mathbf{e}_k is the state estimation error between the true \mathbf{x}_k and estimated $\hat{\mathbf{x}}_k^+$ and \mathbf{P}_k^{-1} is the inverse of the updated state covariance matrix \mathbf{P}_k^+ . As expected the values of the optimality index τ_k vary around the value 1.0 in Fig. 8.3, which means that the proposed filters provided accurate estimate values with small bias error during the first observation span. Fig. 8.4 is the consistency

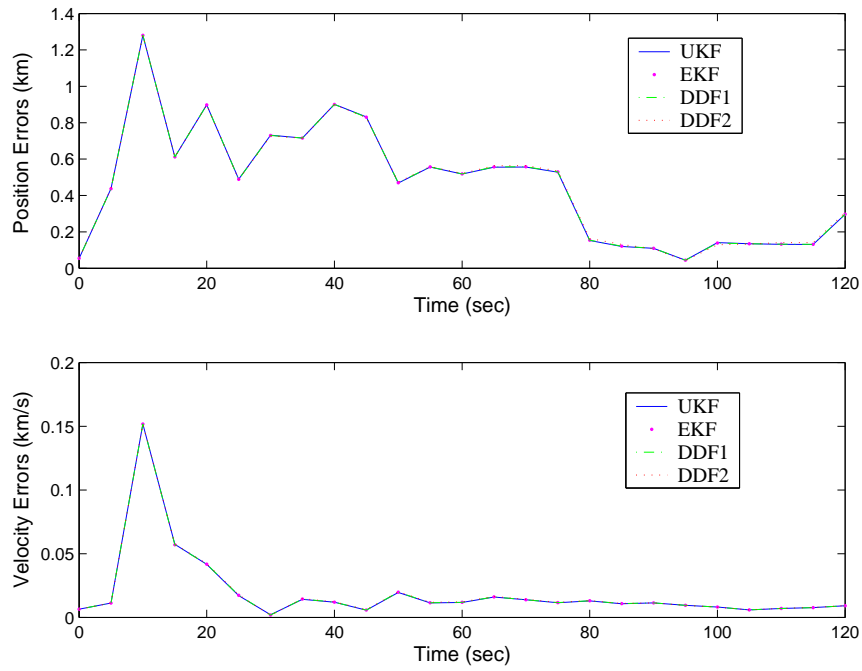


Fig. 8.2 Averaged Position and Velocity Errors with Small Initial Errors

measure taken in the state and covariance propagation part where the error \mathbf{e}_k is the difference between the true \mathbf{x}_k and predicted state $\hat{\mathbf{x}}_k^-$, and \mathbf{P}_k^{-1} is the inverse of the predicted state covariance \mathbf{P}_k^- . In the result the UKF shows the most consistent and accurate prediction performance among the filters. As expected the EKF and DDF1 have similar performance in the prediction accuracy and the DDF2 has a performance between the UKF and the EKF. It indicates that the neglected higher-order terms in the series expansion are not ignorable and could affect the prediction accuracy of the filters with a long time span. Thus, the estimation accuracy with long time separations between tracks will be highly subject to the prediction accuracy of the filters.

In the next simulation, the values of the state and process noise covariance matrices are increased to check the robustness to modeling errors and the sensitivity analysis of the filters to initial errors by multiplying the scale factors, $k_P = 10^1$, $k_Q = 10^2$ for

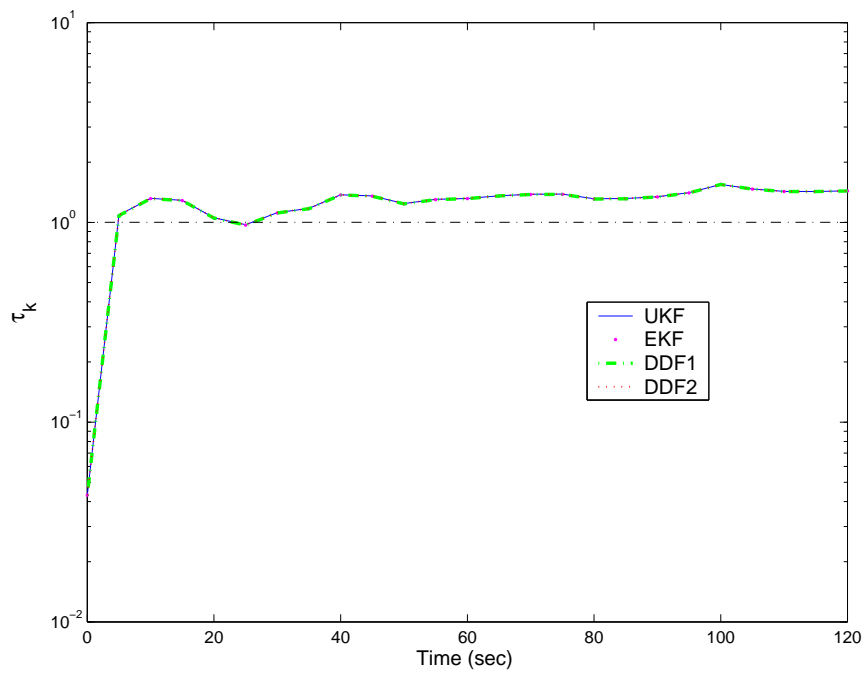


Fig. 8.3 Optimality Index for the State and Covariance Estimates

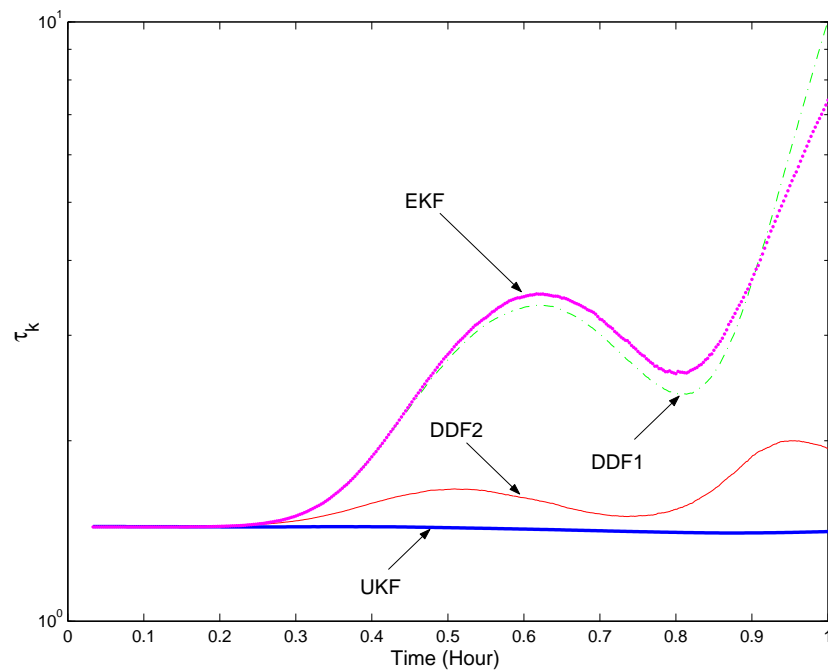


Fig. 8.4 Consistency Analysis for the State and Covariance Prediction

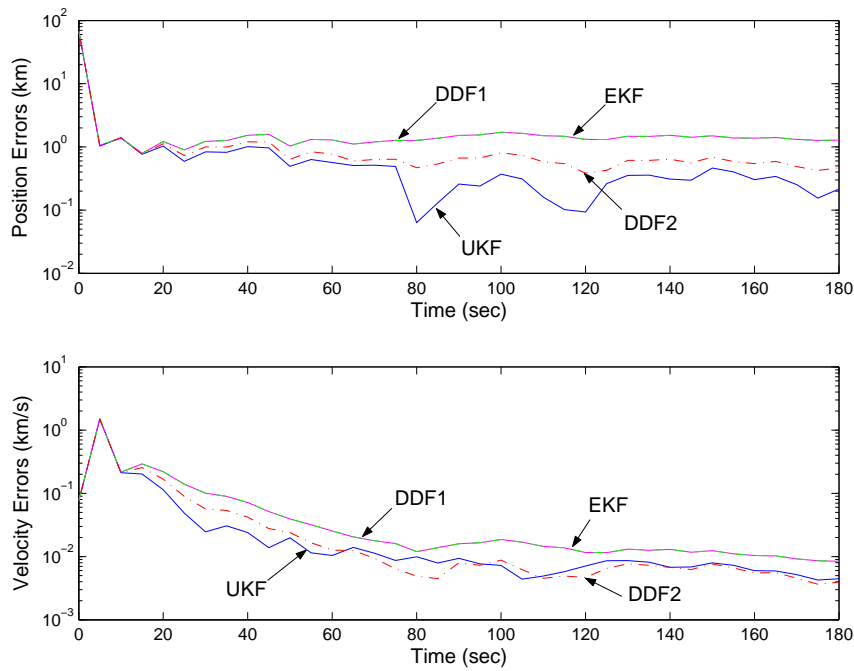


Fig. 8.5 Averaged Position and Velocity Errors for Large Initial Errors

$\mathcal{P}_0 = k_p \mathbf{P}_0$ and $\mathcal{Q} = k_Q \mathbf{Q}(t)$ and by setting large initial errors, $\hat{\mathbf{x}}(t_0) = 0.99 \times \mathbf{x}_{true}(t_0)$. The simulation results in Fig. 8.5 illustrate a possible realistic scenario in orbit estimation due to large initial errors. The position estimate error in the EKF and the DDF1 is reducing, but converges with some biased error. However, the UKF and the DDF2 converge continuously and fast with small bias error, which indicates also that they are performing in a near optimal fashion. The performance in the velocity estimation also shows that the UKF and the DDF2 provide a more accurate estimate than the first-order filters (the EKF and the DDF1). The UKF and the DDF2 exhibit similar performance in the velocity estimation accuracy, but the UKF has smaller RMS position errors than the DDF2.

Now, we consider the estimation with multiple tracks that have a large time separation between themselves. The orbit estimation is performed in the second track separated by 12 hours from the first track. First, the estimated states and

updated covariance matrix obtained from the first track estimation are propagated by using each propagation method until the next available track. For the covariance propagation the differential Lyapunov equation or the state transition matrix is used for the EKF, the scaled unscented transformation (SUT) is applied for the UKF, and the compound matrices containing divided differences are utilized in the DDF1 and the DDF2. Each propagation method gives a different level of the prediction accuracy. The first-order filters, the EKF and the DDF1, have almost the same approximation accuracy up to the first-order Taylor series expansion. The UKF and the DDF2 have the identical state propagation accuracy, but they have slightly different propagation accuracy for the covariance. Both the UKF and the DDF2 filters, however, result in approximations accurate up to the third order of the Taylor series expansion for a Gaussian distribution. Therefore, the inputs to the orbit determination in the second track are the state estimate and covariance matrix propagated from the end of the first track. As the separation time between tracks increases the prediction errors increase due to the neglected nonlinear terms and also the secular terms that arise from an error in the predicted semi-major axis, which affects the estimation in the second track. From the experience in the previous simulation examples, it is expected that superior performance results of the UKF and the DDF2 should be obtained over the first-order filters.

Fig. 8.6 depicts the estimation of the states in the second track with a 3-min observation length separated from the first track by 12 hours. In the results, the qualitative observation is made that the UKF and the DDF2 show far better performance in the state estimation over the EKF when the tracks are separated for multiple orbits with sparse measurement data. It can also be observed how quickly the UKF and the DDF2 converge to the true state of the system. The degradation of the EKF performance is related to the fact that the state and covariance prediction

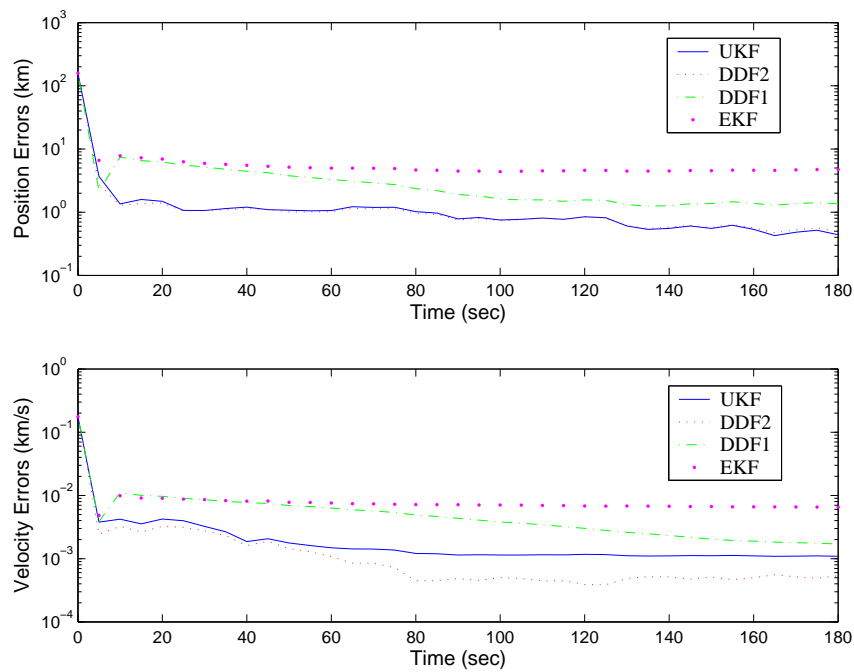


Fig. 8.6 Averaged Position and Velocity Errors After 12-Hour Propagation

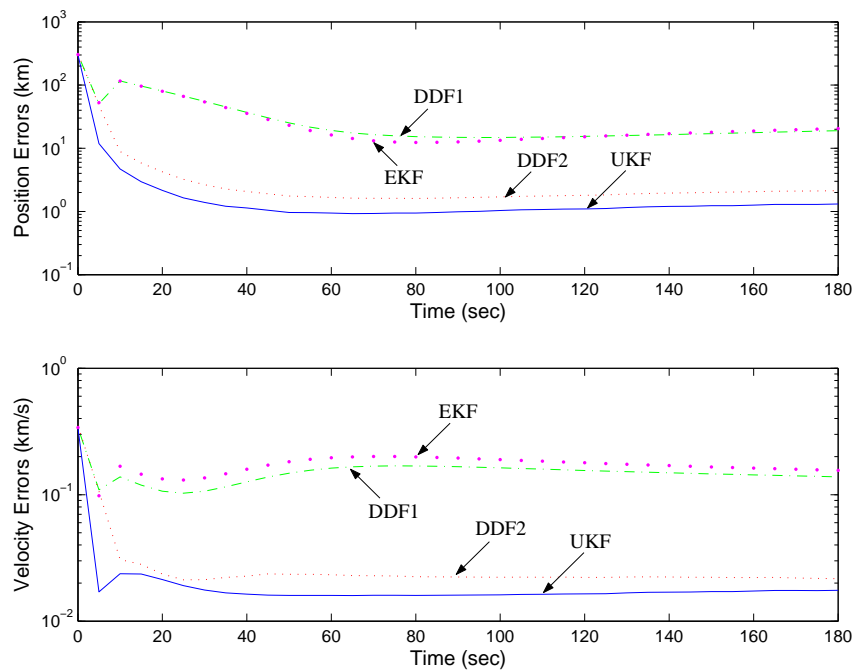


Fig. 8.7 Averaged Position and Velocity Errors After 24-Hour Propagation

executed for a long time interval leads to large prediction errors due to the effects of the neglected nonlinear terms, especially, for the first order filters.

In Fig. 8.7, the second track is separated by 24 hours from the first track, which results in the large initial state and covariance errors to the nonlinear filters. The state estimation errors of the EKF and the DDF1 do not converge well, and result in large bias errors by the end of the second track, whereas the UKF and the DDF2 still show quick convergence performance for both position and velocity estimates. Even more encouraging, the UKF shows the best performance among the filters. This agrees well with our expectations and indicates that the higher-order terms are necessary to adequately capture the large initial condition errors, and they play a role in the accurate estimation of the state and covariance in space catalog maintenance.

Conclusion and Summary

In this paper new nonlinear filtering algorithms, called Sigma Point Filters (SPFs) that include the unscented Kalman filter (UKF) and the Divided Difference Filters (DDF1, DDF2), are utilized in order to obtain accurate and efficient orbit estimation for space catalog maintenance. In addition to the filter consistency performance test with the $3\text{-}\sigma$ bound, an efficient method is introduced in order to measure the dynamic and measurement nonlinearities of the proposed filters along with the whiteness criteria for estimation optimality analysis. Simulation results indicate that the performances of the UKF and the DDF2 are similar, but superior to both the standard Extended Kalman filter and the first-order Divided Difference Filter in terms of the estimate accuracy and sensitivity to large initial errors. In particular, the robustness of the UKF to the initial covariance matrices makes it easy to tune the filter, and the SPFs provide the flexibility of implementation without the derivation of the Jacobian and/or Hessian matrix. The advantages of the proposed algorithms make it suitable

for the efficient real-time satellite orbit estimation for establishing the initial orbits of space objects so that they can be entered into the space object catalog.

8.2.4.2 Adaptive Nonlinear Filtering

In this section, the performance of the proposed adaptive nonlinear filters, the AEKF, AUKF, and ADDF is demonstrated through simulation examples along with estimation optimality analysis. The satellite under consideration has the same orbit parameters used in Table 8.1.

Note that estimating the drag coefficient with the position and velocity compensates for errors in the density. The *solve-for* state vector \mathbf{x} consists of the position, velocity, and drag coefficient, $\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, C_d]^T \in \mathbb{R}^{7 \times 1}$. The true initial values of the state variables were

$$\begin{aligned} x_0 &= 4011.571\text{km}, y_0 = 4702.649\text{km}, z_0 = 3238.358\text{km} \\ \dot{x}_0 &= -5.653\text{km/s}, \dot{y}_0 = 1.540\text{km/s}, \dot{z}_0 = 4.776\text{km/s} \end{aligned}$$

and the drag coefficient C_d was 2.0. For the nominal reference trajectory, the following initial estimates were used

$$\begin{aligned} \hat{x}_0 &= 4011.578\text{km}, \hat{y}_0 = 4702.657\text{km}, \hat{z}_0 = 3238.355\text{km} \\ \dot{\hat{x}}_0 &= -5.654\text{km/s}, \dot{\hat{y}}_0 = 1.537\text{km/s}, \dot{\hat{z}}_0 = 4.772\text{km/s} \end{aligned}$$

and the initial estimate of the drag coefficient was $\hat{C}_d = 3.0$. The initial covariance $\mathbf{P}_0 \in \mathbb{R}^{7 \times 7}$ used for the filters had diagonal elements

$$\mathbf{P}_0 = \text{diag}([10^2 \quad 10^2 \quad 10^2 \quad 5 \times 10^{-2} \quad 5 \times 10^{-2} \quad 5 \times 10^{-2} \quad 5 \times 10^{-1}])$$

and the process noise matrix $\mathbf{Q}(t)$ takes the values

$$\mathbf{Q}(t) = \text{diag}([0 \quad 0 \quad 0 \quad 10^{-16} \quad 10^{-16} \quad 10^{-16} \quad 5 \times 10^{-4}])$$

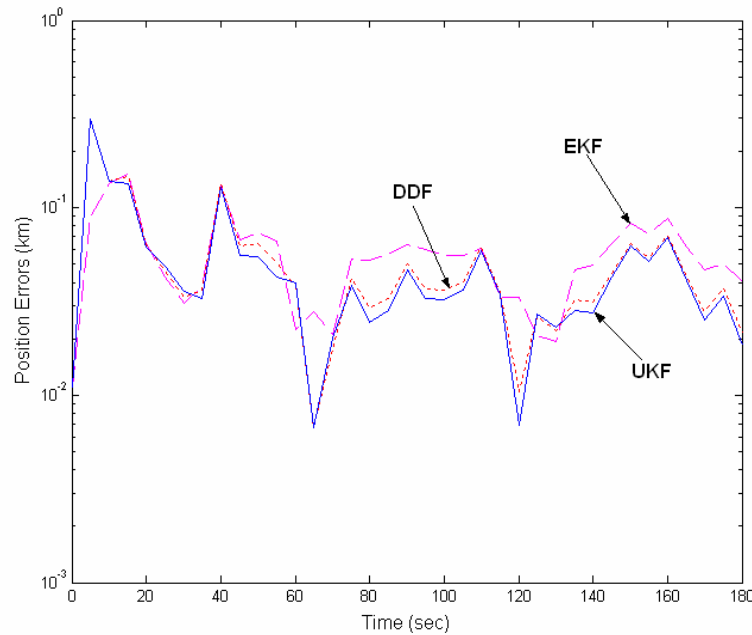


Fig. 8.8 Absolute Position Estimation Errors in First Track

Note that since the process noise covariance $\mathbf{Q}(t)$ comes from a continuous-time dynamic system model, the discrete-time form of the covariance \mathbf{Q}_k can be obtained by using Eq. (8.18).

For establishing accurate estimation conditions a few measurements were utilized to produce an initial orbit determination of the state of the satellite. The initial state estimation was executed by employing the Herrick-Gibbs algorithm,⁹⁸ and the output of the state estimate becomes the input to the recursive nonlinear filters. In the first track the performance of the three nonlinear filters (EKF, UKF, and DDF) are compared without integration of the adaptive process noise estimator.

The absolute magnitude values of the position, velocity and drag coefficient estimation errors for three nonlinear filters are shown in Figures 8.8, 8.9, and 8.10, respectively. As can be seen, the advantage of the SPFs over the EKF in the position

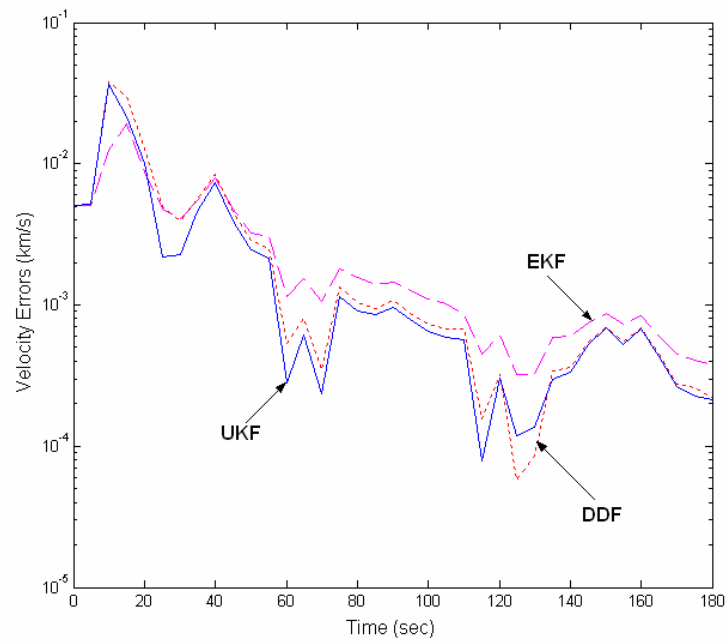


Fig. 8.9 Absolute Velocity Estimation Errors in First Track

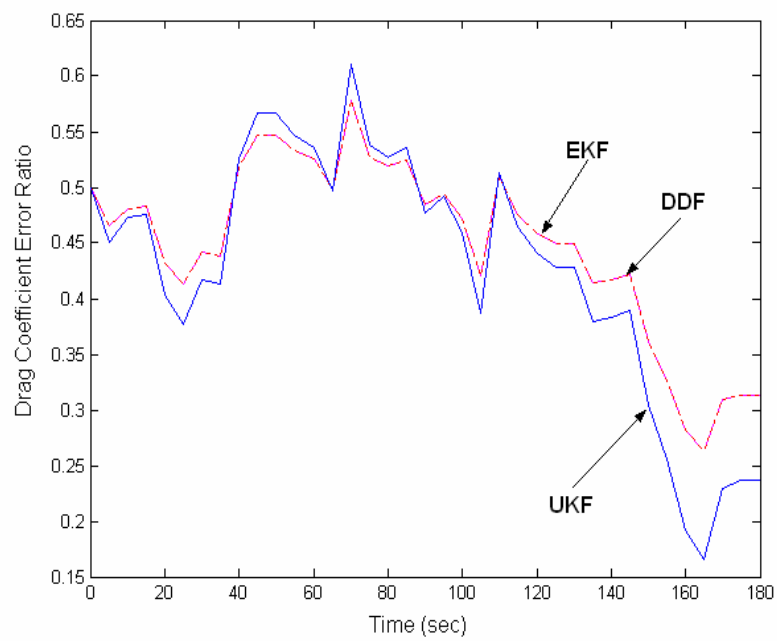


Fig. 8.10 Drag Coefficient Estimation Error Ratio in First Track

and velocity estimation errors is not obvious, but the SPFs are slightly better, which indicates that the effect of nonlinearity on the filters is not severe with the small initial state errors along with the small process noises over the short track length. In Fig. 8.9 the absolute values of the average of the drag coefficient estimate error for the proposed nonlinear filters are presented. All the filters produce converging estimates. The UKF demonstrates accurate performance in the drag parameter estimation, but still the estimate error is not fully converged. This is because the short track does not allow the filters to extract enough information to get the parameter estimate converged. Note that usually long arcs of several months are employed in order to estimate or calibrate the ballistic or drag coefficient in order to obtain a good estimate of the ballistic coefficient.

The results represented in the above can be verified by employing the optimality index τ_k . Fig. 8.11 shows the optimality measure taken in the state and parameter estimation. The results from the SPFs and EKF exhibit similar consistent performance with the value close to unity, which indicates that the neglected higher-order terms in the series expansion are not severe, and all the filters are producing accurate estimates with small biased errors. Fig. 8.12 shows the measurement innovation errors that lie inside the 2-sigma bound without any deviations. Even though each filter produces slightly different innovation errors, they all fall inside the boundary with close optimal performance. According to the RMS errors, the optimality index, and the innovation error criteria it is seen that the performance of the nonlinear filters is near-optimal when the process and measurement noise are correctly selected but the SPFs are slightly better than the conventional EKF.

Now, the orbit estimation is performed in the second track which is separated from the end of the first track by 24 hours. Due to the long time span between the tracks large uncertainties exist at the beginning of the second track. During the

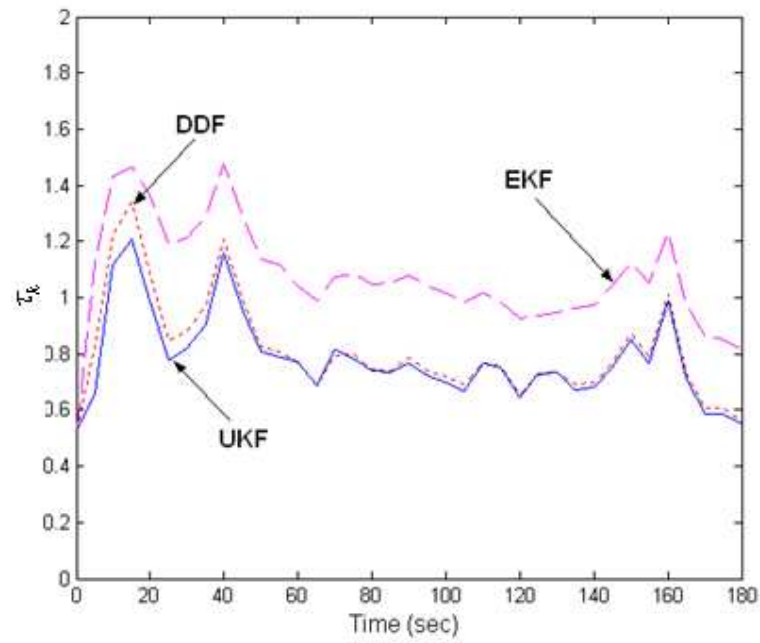


Fig. 8.11 Consistency Test with Optimality Index

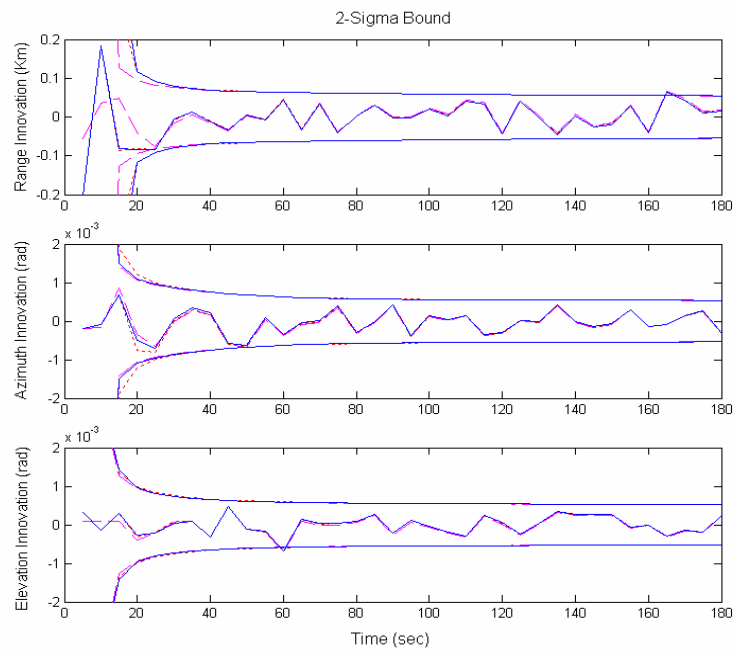


Fig. 8.12 Measurement Innovation Errors with 2-sigma Bound

prediction period between the first and second tracks, the discrete-time covariance propagation equation given in Eq. (7.84) was used in order to propagate the estimated covariance matrix. The estimated state was propagated by numerically integrating the filter dynamic model. Thus, the inputs to the orbit determination in the second track are the *a priori* state estimate and covariance matrix propagated from the end of the first track. The filter prediction equations used consist of the two-body motion, J_2 zonal perturbation and standard exponential drag perturbation, but no process noise was added to it.

As the separation time between tracks increases the prediction errors increase due to the neglected nonlinear terms in the prediction process and errors in the estimated state, which affects the estimation in the second track. The uncertainties or neglected modeling errors in the second track, however, can be compensated by utilizing the proposed adaptive nonlinear filtering techniques that adaptively estimate the process noise covariance matrix. Thus, the purpose of the orbit estimation in the second-track is to compare the performance of the proposed adaptive nonlinear filters, AEKF, AUKF and ADDF with the standard nonlinear filters.

The true initial values of the state variables for the second track estimation were

$$\begin{aligned} x_0 &= 5064.297\text{km}, y_0 = 4058.090\text{km}, z_0 = 2563.877\text{km} \\ \dot{x}_0 &= -4.769\text{km/s}, \dot{y}_0 = 2.647\text{km/s}, \dot{z}_0 = 5.236\text{km/s} \end{aligned}$$

and the drag coefficient was set $C_d = 2.0$. For the nominal reference trajectory, the initial estimates for the second track estimation were used

$$\begin{aligned} \hat{x}_0 &= 4916.498\text{km}, \hat{y}_0 = 4136.048\text{km}, \hat{z}_0 = 2721.477\text{km} \\ \dot{\hat{x}}_0 &= -4.946\text{km/s}, \dot{\hat{y}}_0 = 2.501\text{km/s}, \dot{\hat{z}}_0 = 5.142\text{km/s} \end{aligned}$$

and the initial estimate of the drag coefficient was $\hat{C}_d = 3.0$. The initial covariance

$\mathbf{P}_0 \in \mathbb{R}^{7 \times 7}$ was obtained (only diagonal terms were represented)

$$\mathbf{P}_0 = \text{diag}([6447.7081 \quad 1539.250 \quad 6727.063 \quad 0.000008 \quad 0.00610 \quad 0.00275 \quad 5.3125])$$

The process noise covariance matrix was taken based on the value in the first track, but the variance value of the drag coefficient was increased in order to consider uncertainty effects due to the propagation.

$$\mathbf{Q}(t) = \text{diag}([0 \quad 0 \quad 0 \quad 10^{-16} \quad 10^{-16} \quad 10^{-16} \quad 5 \times 10^{-3}])$$

The adaptive nonlinear filters used for the state and parameter estimation are based on the identification of the process noise $\mathbf{Q}(t)$. Since each adaptive filter produces a different value of the objective cost function J that is the sum of the innovation errors, the scales factors calibrated from the *Downhill Simplex*⁸⁸ optimization method are different. The weight factor or wind size γ obtained from the adaptive noise estimator was 4.5×10^5 for the AEKF, the values for the AUKF and ADDF were close with $\gamma = 1.65 \times 10^2$.

Figures 8.13 and 8.14 are the plots of the performance comparison of the adaptive filters and nonlinear filters with respect to the position and velocity estimation errors in the second track, respectively, which illustrate a possible realistic scenario in orbit determination. From the previous simulation results in the first track, it is expected that superior performance results of the UKF should be obtained over the EKF. However, the UKF results of the averaged magnitudes of the position and velocity estimation errors are very close to those of the EKF with the biased estimation errors. The degradation of the EKF and UKF performance is related to the fact that the state and covariance prediction executed for a long-time interval leads to large prediction errors due to the effects of the neglected nonlinear terms, and also the parameter uncertainties such as the drag coefficient. Thus, the filters start estimating the states

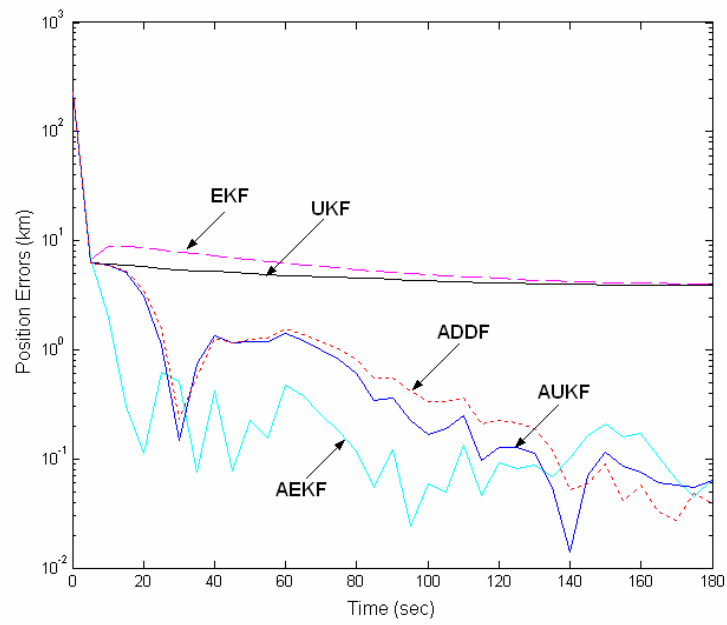


Fig. 8.13 Absolute Position Estimation Errors in Second Track

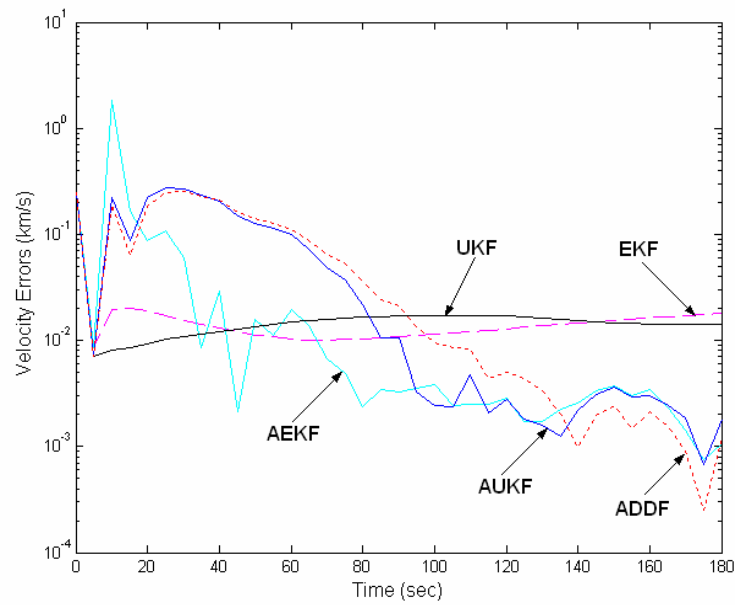


Fig. 8.14 Absolute Velocity Estimation Errors in Second Track

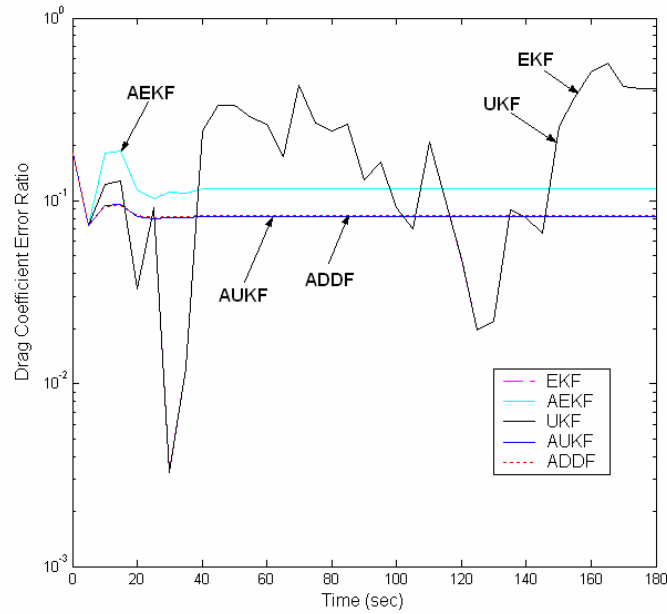


Fig. 8.15 Drag Coefficient Estimation Error Ratio in Second Track

with large initial state errors and large covariance matrices with unknown statistical information of the uncertainties or noise. Under the incorrect noise information, even the UKF or more higher-order nonlinear filters can't produce optimal estimates due to the violation of the optimality conditions. On the other hand, all the adaptive filters (AUKF, ADDF and AEKF) converge continuously and fast with small bias error, which indicates they are performing in a near optimal fashion. The performance in the velocity estimation also shows that the adaptive nonlinear filters provide a more accurate estimate than that of the standard nonlinear filters (the EKF and the UKF). This agrees well with our expectations and indicates that the correct noise information is necessary for the filters to perform optimally.

In Fig. 8.15 the absolute values of the drag coefficient error ratio with respect to the proposed adaptive filters are shown during the second track, where the error ratio is the ratio between the true and estimated drag coefficient. As expected the

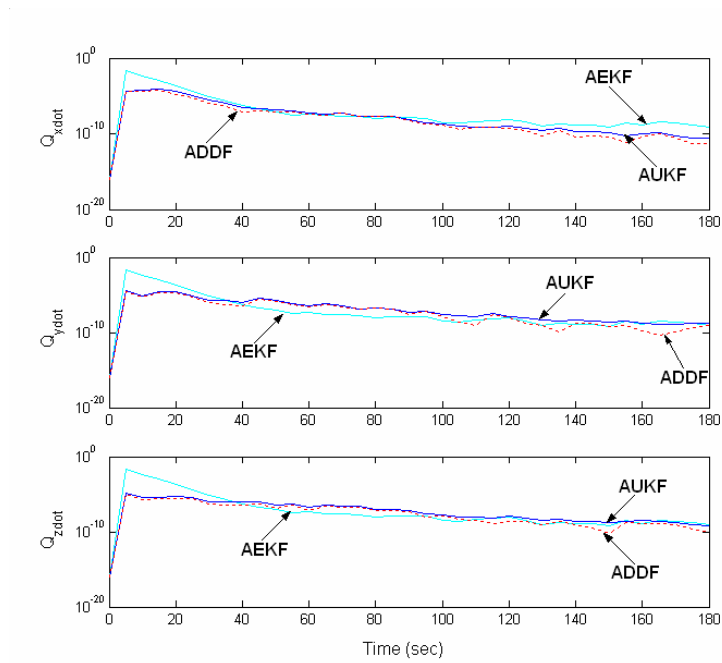


Fig. 8.16 Adaptive Covariance Estimation with Q Adaptation

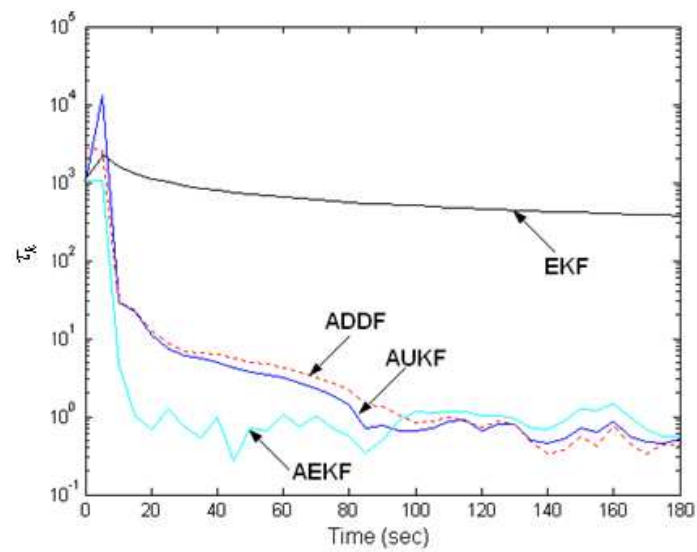


Fig. 8.17 Consistency Test with Optimality Index

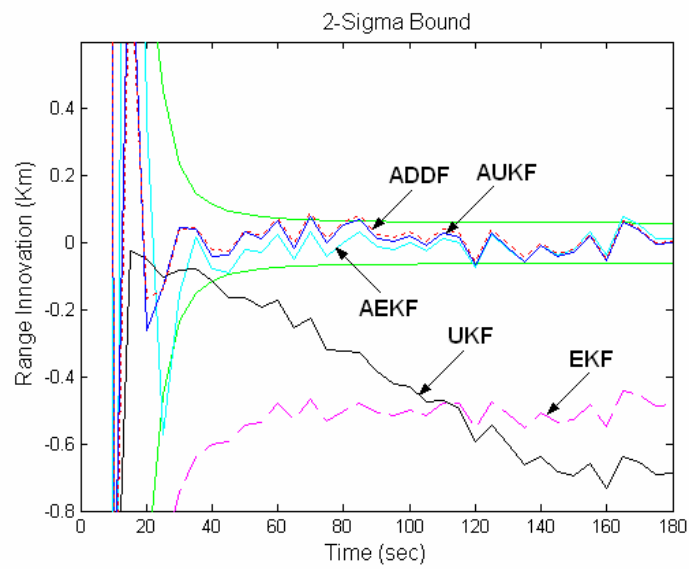


Fig. 8.18 Range Innovation Errors with 2-Sigma Bound

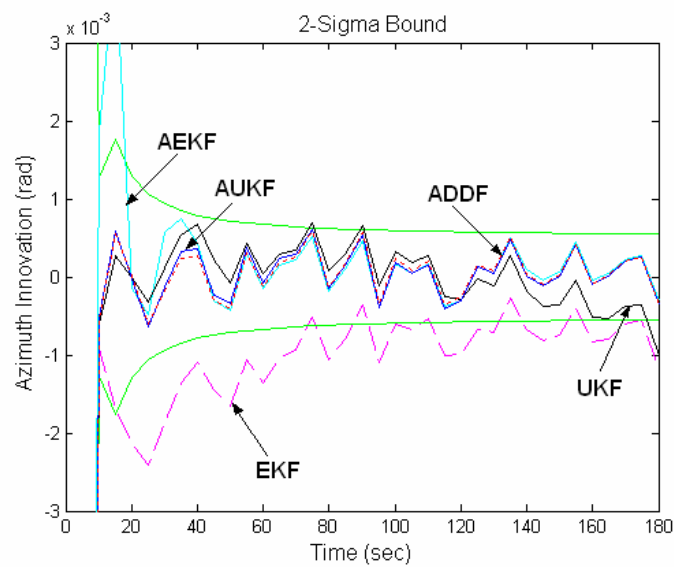


Fig. 8.19 Azimuth Innovation Errors with 2-Sigma Bound

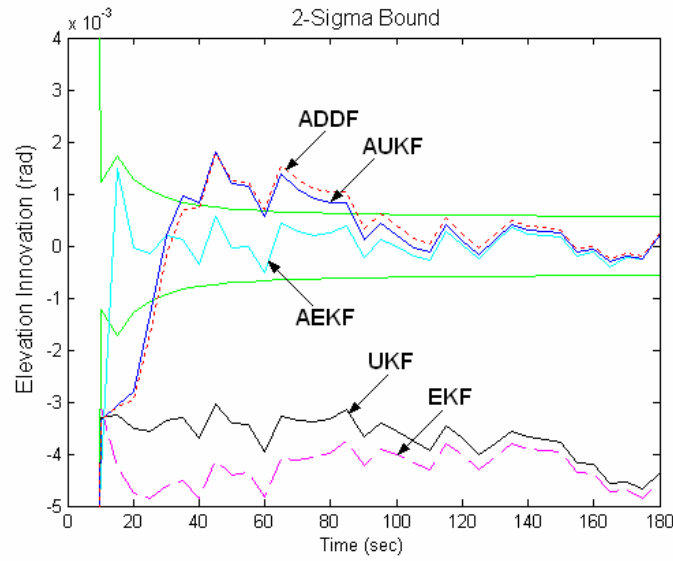


Fig. 8.20 Elevation Innovation Errors with 2-Sigma Bound

parameter estimation with the adaptive filters also successfully generated converged solutions with fast and accurate estimates. The EKF and UKF also converge, but with large bias errors. In the drag coefficient estimation, the AUKF and ADDF show better performance over the AEKF.

Fig. 8.16 illustrates the adaptation of the process noise variance generated from the adaptive nonlinear filters as a function of time. It is seen that while the manually-tuned covariance is constant with time, the estimated covariance has time-varying values by continuously estimating and adapting the noise statistics for an optimal performance. From the results it is seen that the increased process noise variances at the initial estimation make the prediction covariance and the Kalman gain larger, therefore the observations have much influence on the filters. In contrast, as the variances decrease with time, the Kalman gain become small, thus the observations are not incorporated into the filters. For optimal performance of the filters the process noise variance is required to increase from the initial variances.

Now, the results obtained above can be verified by evaluating the nonlinearity index, τ_k , in Eq. (8.12) and the whiteness of the innovation vector, $\bar{\rho}_l$, in Eq. (8.13), respectively. Fig. 8.17 shows the nonlinearity measure in the estimation of the satellite state and parameter from the proposed filters. The nonlinear index obtained by the adaptive nonlinear filters varied around unity, but the index from the EKF remained over the value with large deviations. This indicates that the ASPFs and AEKF exhibit similar optimal performance, and successfully compensate for the neglected higher-order terms in the series expansion and unknown uncertainties. Figures 8.18 ~ 8.20 depict the innovation errors with the 2-sigma bound. The innovation errors from the adaptive filters vary inside the sigma bound, but the innovations from the EKF and UKF are outside the bound. According to these results, we can also judge that the adaptive filters achieved the near-optimal performance.

According to the RMS errors, the nonlinearity index, and the innovation error criteria presented so far, it is seen that the performance of the adaptive nonlinear filters is optimal in the sense that they compensate for the neglected modeling errors, as well as the unknown uncertainties.

Conclusion and Summary

In this paper new adaptive nonlinear filtering algorithms called the adaptive unscented Kalman filter (AUKF) and the adaptive divided difference filter (ADDF) were derived by integrating the sigma point filters (UKF and DDF) with a new adaptive estimator formulated by a numerical optimization. The purpose of the proposed adaptive nonlinear filters was to not only compensate for the nonlinearity effects neglected from linearization, but also to take into account the system modeling errors by adaptively estimating the noise statistics and unknown parameters. The performance of the AUKF and the ADDF was superior to the standard nonlinear filters (EKF and

UKF) in terms of the fast convergence and estimation accuracy. The advantages of the proposed adaptive nonlinear filtering algorithms make these attractive alternatives to the standard nonlinear filters for efficient state and parameter estimation not only in satellite orbit determination but also in other application areas.

8.3 Autonomous GPS Navigation

In this section, the performance of the proposed nonlinear filters with integration of the dynamic model compensation (DMC) method is illustrated with application to autonomous navigation applications.

When the spacecraft has the potential for significant accelerations, it is usually necessary to measure and account for the changes in velocity, which leads to the integrated GPS/INS navigation system. However, in this formulation it is assumed that the Inertial Measurement Unit (IMU) including gyroscopes and accelerometers is not integrated with the GPS. This chapter focuses on how the GPS observations are utilized to provide a navigation solution. Therefore, the stand-alone GPS navigation system estimates the vehicle state and parameters. Literature for the integrated GPS/INS navigation system is found in Refs. 113, 115, and 123.

8.3.1 Generation of User True Trajectories

The simulated user satellite has the following orbital elements at epoch defined to be simulation time zero.

The simulated true orbit for the user satellite is generated by numerical integration of the acceleration function by means of a fourth-order Runge-Kutta method. The acceleration function consists of a gravitational component and a drag component. The gravitational potential is modeled as a sixth-order aspherical potential

Table 8.2 User Satellite Orbital Elements

Orbital Elements	Values
a	7678160.0 (m)
e	0.0
i	1.0996 (rad)
Ω	1.5708 (rad)
ω	0.0 (rad)
M	0.0 (rad)

function in Eq. (7.4), and the acceleration due to gravity is derived by taking the gradient of this potential function in Eq. (7.5), respectively.

8.3.2 Generating of GPS Satellite Trajectories

To simulate a GPS scenario a description of typical GPS satellite orbit trajectories is needed. The GPS satellite information is usually given by a GPS almanac such as the Yumma almanac, which provides orbital elements including semi-major axis, eccentricity, inclination, right ascension, argument of perigee, and mean anomaly. These parameters are used to provide an initial position and velocity in the ECI coordinate system. The orbital elements at simulation epoch for the GPS satellites are listed in in the following Table 8.3.

The description of the orbital elements is illustrated in Fig 8.21 and each specific

Table 8.3 GPS Constellation Elements

Satellites	a (m)	e	i (deg)	Ω_0 (deg)	f_0 (deg)
1	26559800	0.0	55.0	325.7	72.1
2	26559800	0.0	55.0	25.7	343.9
3	26559800	0.0	55.0	85.7	214.9
4	26559800	0.0	55.0	145.7	211.9
5	26559800	0.0	55.0	205.7	93.9
6	26559800	0.0	55.0	265.7	27.9

equations and values are given by

$$f(t) = f_0 + \sqrt{\frac{\mu}{a^3}} \times (t - t_0) \quad (8.20)$$

$$\Omega(t) = \Omega_0 - \omega_e(t - t_0) \quad (8.21)$$

$$i = 55^\circ \quad (8.22)$$

$$R = 26560 \text{ km} \quad (8.23)$$

The secular variation of Ω is due to the rotation of the Earth, w_e is the rate of rotation of the Earth and μ is the gravitational constant

$$\omega_e = 7.29212 \times 10^{-5} \text{ rad/s} \quad (8.24)$$

$$\mu = 398600.44 \text{ km}^3/\text{s}^2 \quad (8.25)$$

The GPS simulated true orbits are propagated by a two-body Keplerian procedure. More detailed descriptions are available in Ref. 120.

8.3.3 Filter Dynamic Model and Error Sources

The dynamic state solution requires a dynamic model for the propagation of the state vector and covariance matrix. The perturbing acceleration terms for the filter

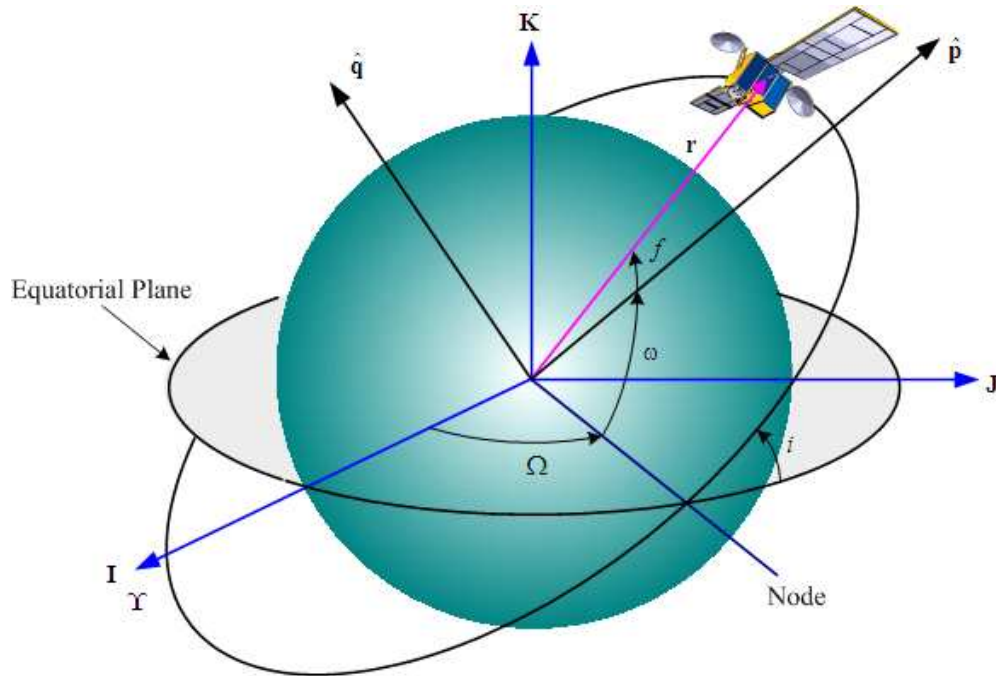


Fig. 8.21 Description of the Geometry of GPS Satellite Orbit

dynamic model consist of the gravitational acceleration term and the atmospheric drag. The gravitational potential for the filter dynamic model includes only the J_2 harmonic term

$$U_p = \frac{\mu}{r} \left[1 - \frac{J_2}{2} \left(\frac{r_e}{r} \right)^2 (3 \sin^2 \delta - 1) \right] \quad (8.26)$$

The gradient of this provides the components of the gravitational acceleration for the filter dynamic model.

The errors used for dynamic model parameters, GPS ephemeris data, and measurements are listed in Table 8.4 where the subscript “p” denotes the user perceived value and the values given in the table are the true simulation values.

The initial estimates of position, velocity, clock bias, and clock drift are calculated by means of the geometric state solution, and these initial estimates become the *a priori* values to start the dynamic state solution process. The assembling pseudorange

Table 8.4 Simulation Error Sources

Error Sources	Values
GPS Position	Mean = 0.0 m
GPS Velocity	Mean = 0.0 m
Pseudorange Measurement	Mean = 0.0 m
Pseudorange Rate Measurement	Mean = 0.0 m
User Clock Bias	0.12 (sec) (At epoch)
User Clock Bias	1.0×10^{-9} sec / sec

and range rate measurements for the measurement update process, the user satellite searches for visible GPS satellites. The availability of a particular GPS satellite at a given user location is investigated by checking the vertical elevation angle. The user satellite can access all visible GPS satellites which are above the 10 degree elevation angle.

8.3.4 Simulation Results and Performance Analysis

The time span for the filter dynamic simulation has a 1500 (sec) interval. A geometric solution at the initial epoch time $t_0 = 0$ provides the *a priori* state estimate to start the dynamic filtering process, and the interval for the GPS observation is updated at each 5 (sec).

Note that both the geometric solution and the least-squares point solution do not provide any state covariance data, thus the initial a priori covariance matrix \mathbf{P}_0 is assumed to be chosen with diagonal elements. The initial variances for the position and velocity components are given by

$$\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = 1.0 \times 10^5 (m)^2 \quad (8.27)$$

$$\sigma_{\dot{x}}^2 = \sigma_{\dot{y}}^2 = \sigma_{\dot{z}}^2 = 1.0 \times 10^3 (m/s)^2 \quad (8.28)$$

and the initial variances for the system parameters μ , J_2 , C_d , bc , \dot{bc} , and τ are 1.0×10^{20} , 1.0×10^2 , 1.0×10^{10} , 10, 10, 1.0×10^4 , respectively. The variances of the three DMC acceleration components have the same input value of σ_u^2 used for the process noise matrix for a particular run. Therefore, the *a priori* covariance matrix \mathbf{P}_0 has the following elements

$$\mathbf{P}_0 = \begin{bmatrix} 10^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{20} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{10} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_u^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_u^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_u^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^4 \end{bmatrix} \quad (8.29)$$

The standard deviation of the process noise for clock bias clock drift, and correlation time is set

$$\sigma_b = \sqrt{q_b} = 0.4472, \sigma_d = \sqrt{q_d} = 0.1414, \sigma_\tau = \sqrt{q_\tau} = 0.0 \quad (8.30)$$

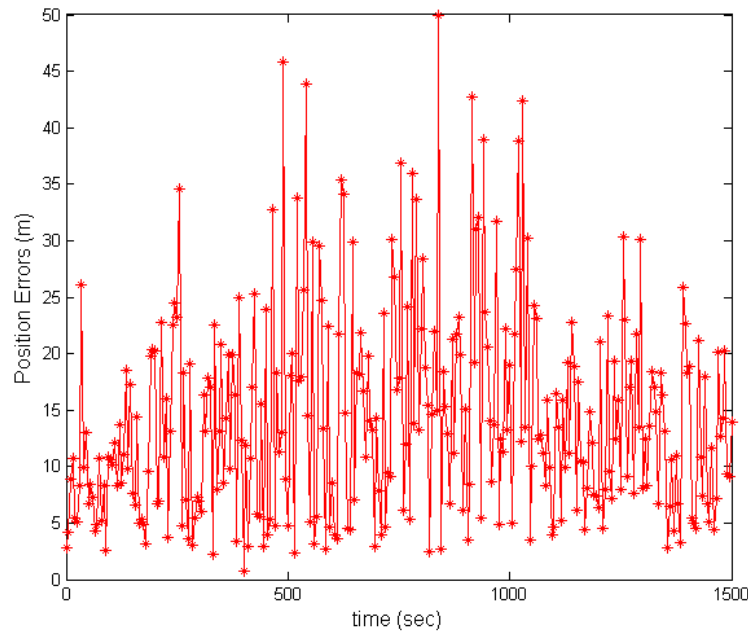


Fig. 8.22 Position Errors for Geometric Solution

In order to make the simulation simple, an optimal value of τ is decided by observing the time history of its value over the time span from the filtering process. Once a near optimum for τ has been determined, the value of σ_u can be decided. In this simulation work, the estimated values of the time correlation and associated standard deviation of the process noise in the DMC are set as

$$\tau = 4000, \sigma_u = 0.1 \quad (8.31)$$

Figs. 8.22, 8.23, and 8.24 show the position, velocity, and clock bias errors of the user satellite generated from the geometric solution as a function of time. It is seen that the position, velocity, and clock bias error histories have large variation since the geometric solution is dependent on the GPS satellite geometry. One of the disadvantage of the geometric solution lies in the fact that it does not propagate the state

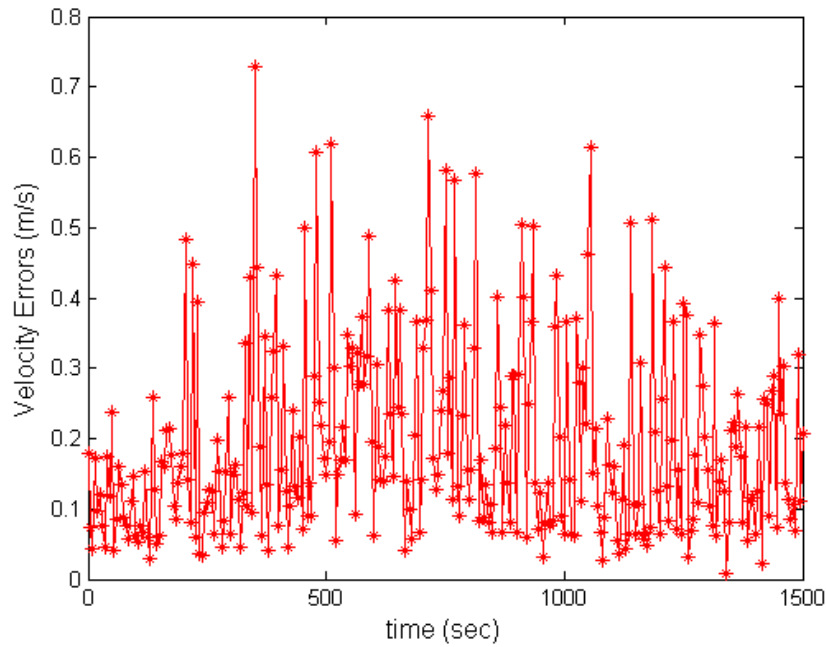


Fig. 8.23 Velocity Errors for Geometric Solution

Table 8.5 RMS Errors from Geometric Solution (Static Solution)

RMS	$\varepsilon_{\mathbf{r}}(m)$	$\varepsilon_{\mathbf{v}}(m/s)$	$\varepsilon_{bias}(sec)$
User RMS Errors	14.21	0.189	7.25e-09

information from one measurement to the next and each solution is computed independently. However, this independent characteristic provides a stable, non-divergent solution that the dynamic estimation solution may face.

The RMS values for the position, velocity, and the user clock bias errors from the geometric solution are summarized in Table 8.5.

In the following simulations the dynamic model compensation technique is applied. The dynamic model compensation estimation is also called adaptive filtering in that it compensates for the unknown perturbing acceleration errors in the system

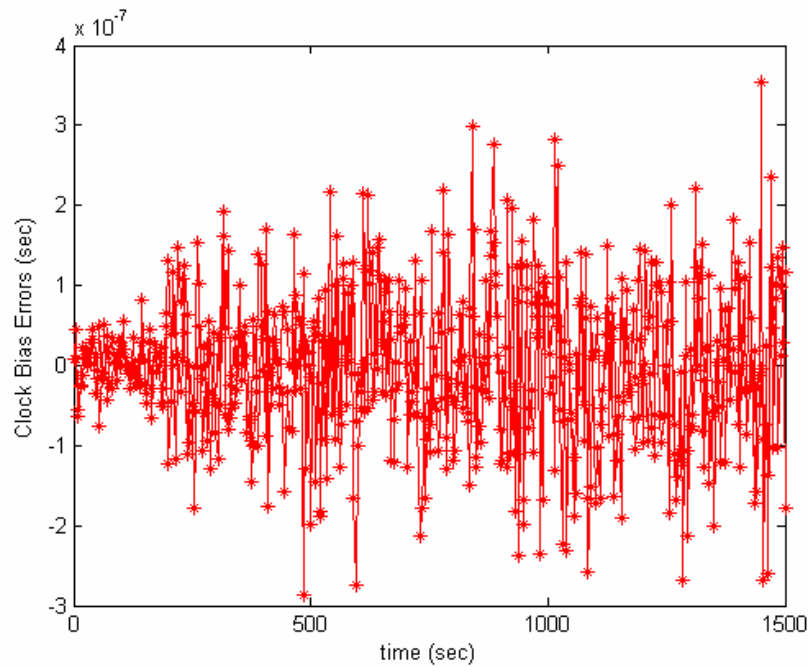


Fig. 8.24 Clock Bias Errors for Geometric Solution

model through the first-order Gauss Markov process.

Figs. 8.25 and 8.26 show the position, velocity, and clock bias errors of the user satellite generated from the dynamic model compensation estimation solution as a function of time. Figs. 8.27 and 8.28 exhibit the user clock bias and clock drift errors generated from the dynamic model compensation estimation solution as a function of time. In these results, the EKF algorithm is used to formulate the DMC estimation for the autonomous GPS navigation. As can be seen, the EKF based DMC adaptive estimation achieves considerably better estimation accuracy than the geometric solution. The dynamic filtering approach produces a small error in the user clock bias, but the clock error from the geometric solution is nearly unbiased. The consistent clock drift error was obtained shown in Fig. 8.28.

The RMS values for the position, velocity, and the user clock bias errors generated

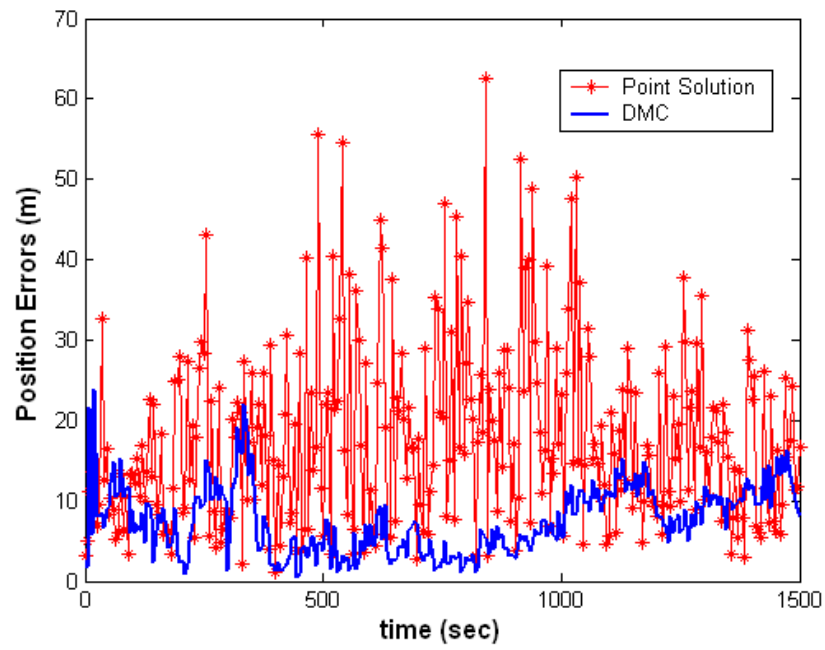


Fig. 8.25 Position Errors for DMC Solution

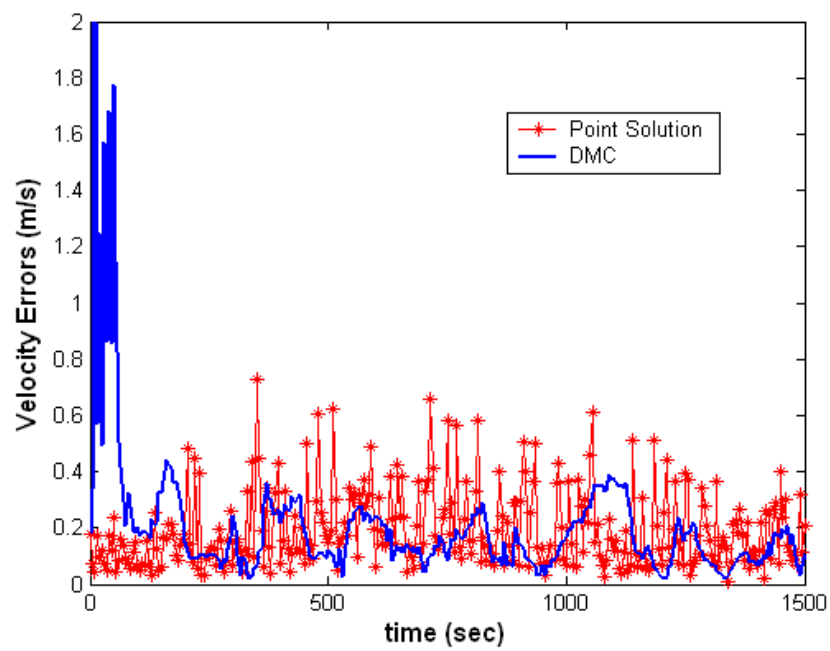


Fig. 8.26 Velocity Errors for DMC Solution

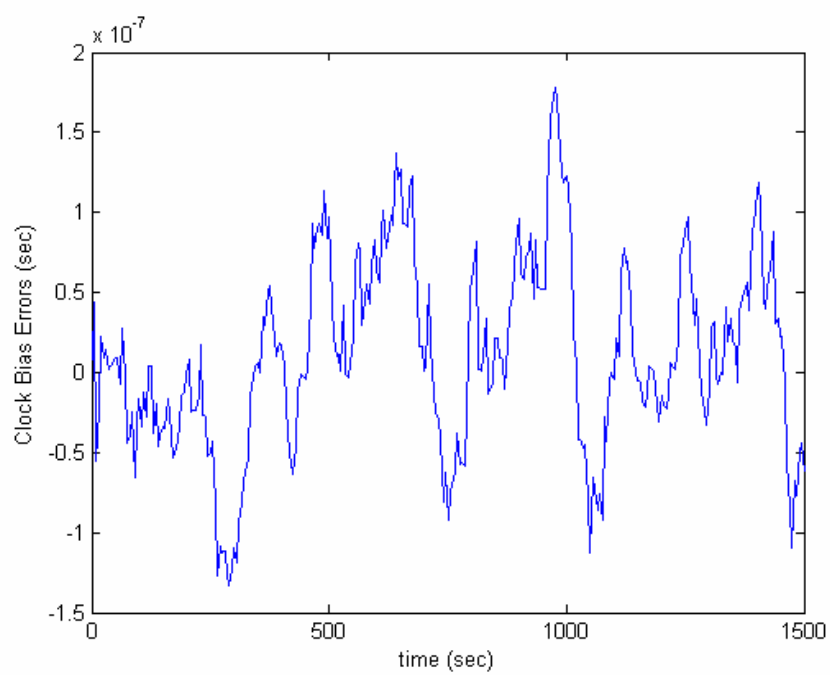


Fig. 8.27 Clock Bias Errors for DMC Solution

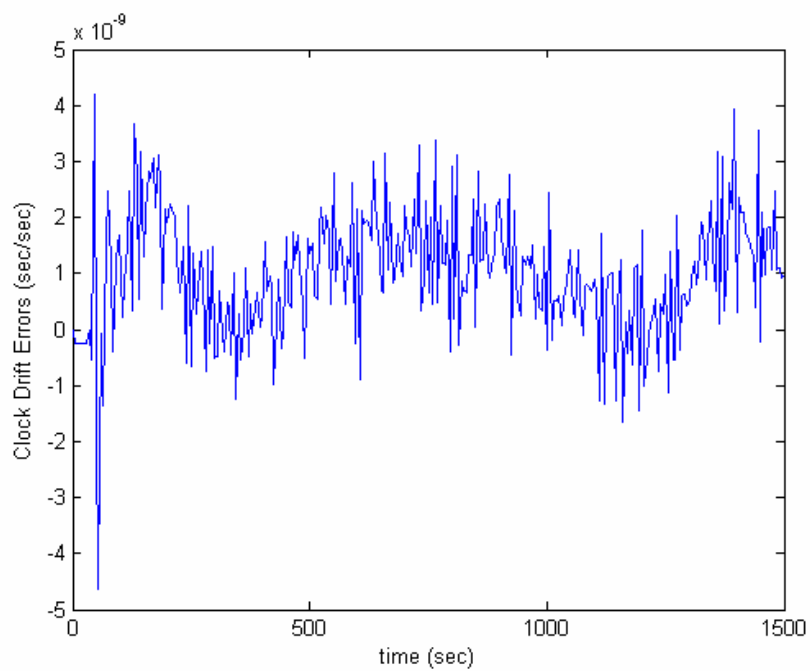


Fig. 8.28 Clock Drift Errors for DMC Solution

from the adaptive model compensation estimation are summarized in Table 8.6. It is

Table 8.6 RMS Errors from Dynamic Model Compensation Estimation

RMS	$\varepsilon_{\mathbf{r}}(m)$	$\varepsilon_{\mathbf{v}}(m/s)$	$\varepsilon_{bias}(sec)$
User RMS Errors	7.92	0.098	3.92e-09

seen that the RMS errors from the DMC estimation results in smaller errors compared with the geometric point estimation. The DMC navigation solution can substantially improve the definitive and predictive accuracy of GPS receiver point solution fixes, achieving accuracies on the order of 10 meters and 0.09 m/sec. However, the DMC method is dependent on the tuning parameters of the time correlation time τ and the variance of the process noise σ_u^2 in order to produce sub-optimal or near-optimal estimation results. The question that arises is how the tuning parameters can be optimally estimated or selected. One solution to the question is to utilize fuzzy logic or a genetic algorithm to optimize the tuning parameters. The adaptive model compensation using the genetic algorithm had been investigated in Ref. 97.

8.3.5 Conclusion

The geometric solution offers the simple and fast calculation and is not subject to dynamic modeling errors. However, the series of the geometric solution exhibits large variations. The performance of the geometric approach is dependent on measurement accuracy and viewing geometry of the GPS satellites. When an *a priori* state information is not available, this geometric solution can play a role of a batch filter to provide an initial state for a dynamic estimation filter. The integrated adaptive EKF dynamic algorithm provides better accurate estimation results. The DMC estimation play a role of an adaptive filtering in that it compensates unknown perturbing ac-

celeration errors in the system model. However, the adaptive capability of the DMC estimation is limited since it is subject to optimizing the tuning parameters, the time correlation and the variances of the process noise.

8.4 Autonomous Robot Navigation

Mobile robots are typically equipped with sensors such as GPS, IMU, video cameras, ultrasonic or infrared range, acoustic, and contact sensors.¹²⁴ The unknown state processes are robots' position and/or velocity. In this study, an autonomous robot shown in Fig. 8.29 attempts to determine its position using the measurements from a minimum set of sensors. One sensor measures the range or distance between the center of the robot and a particular point on a wall using ultrasound, and the other sensor measures the robot's speed. The robot also has proximity sensors to avoid collisions with objects and the wall. In general, for the precise local robot navigation, an additional set of sensors that measure its heading angle and angle rate is required. Without rate sensors, the performance of the standard nonlinear filters such as the EKF in the robot navigation could be degraded.¹²⁴

In this section, efficient cutting-edge nonlinear filtering algorithms for robot navigation with a minimum set of sensors are investigated. The purpose of this application is to test how the state-of-art filtering algorithms work with a minimum number of sensors. The nonlinear filters range from the sigma point filters (DDF, UKF), the generic SIR particle filter to the newly proposed sigma-point based particle filter,¹²⁵ and their performances are compared in terms of the RMS estimation error and computational load.

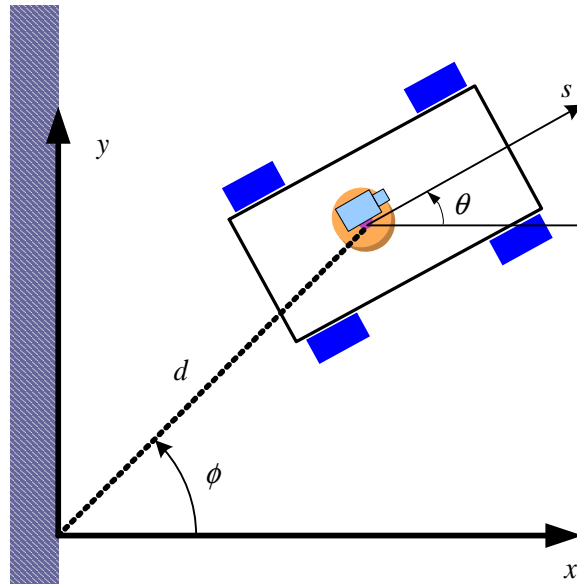


Fig. 8.29 Diagram for Autonomous Robot Navigation System

8.4.1 Robot Dynamic System

The robot dynamics are described in terms of the following scenario. There are several constraints on the robot motion; 1) the robot's speed can not be negative, 2) there is a maximum value for the robot's speed s_{max} , 3) there is no slippage of the tracks of the robot, and 4) if the proximity sensor indicates that the robot is about to hit the wall (its x position is about to become negative), then it stops and turns around (its speed is set to zero and its orientation is preserved).

Now the kinematic model for the robot motion is described along with incorporating these constraints. The components of the robot state \mathbf{x}_k are defined by

$$\mathbf{x}_k = \begin{bmatrix} x(k) \\ y(k) \\ s(k) \\ \theta(k) \end{bmatrix} \quad (8.32)$$

where $x(k)$ is the robot's position, $y(k)$ is the robot's position, $s(k)$ is the robot's speed, and $\theta(k)$ is the robot's heading angle, respectively.

To develop the discrete-time nonlinear dynamics equations, the following definitions are made

$$\begin{aligned} x'(k+1) &= x(k) + s(k)\Delta t \cos \theta(k) + w_x(k) \\ s'(k+1) &= s(k) + w_s(k) \end{aligned} \quad (8.33)$$

where Δt is the time between observations. With these definitions, the discrete-time dynamics model for robot's motion is described by

$$\begin{aligned} x(k+1) &= \max(0, x'(k+1)) \\ y(k+1) &= y(k) + s(k)\Delta t \sin \theta(k) + w_y(k) \\ s(k+1) &= \begin{cases} 0, & x'(k+1) \leq 0 \text{ or } s'(k+1) \leq 0 \\ s_{\max}, & s'(k+1) \geq s_{\max} \\ s'(k+1), & \text{otherwise} \end{cases} \\ \theta(k+1) &= \begin{cases} \theta(k) + \pi + w_\theta(k), & x'(k+1) \leq 0 \\ \theta(k) + w_\theta(k), & x'(k+1) \geq 0 \end{cases} \end{aligned} \quad (8.34)$$

where $\mathbf{w}_k = [w_x, w_y, w_s, w_\theta]^T$ is a zero-mean white Gaussian process noise vector whose covariance matrix is $E\{\mathbf{w}_k(k)\mathbf{w}_j^T(k)\} = \mathbf{Q}(k)\delta_{kj}$.

8.4.2 Sensor Measurements

One of the most important tasks of an autonomous robot is to acquire its environments, position and velocity, by taking measurements from various sensors such as GPS, heading-angle sensors, speed sensors and so on.¹²⁶ In this work, an autonomous mobile robot attempts to determine its position by using the measurements from two sensors. One sensor measures the distance between the robot and a particular point on a wall using ultrasound, and the other sensor measures the robot's speed. The

robot also has proximity sensors to avoid collisions with objects and the wall.

8.4.2.1 Range Measurement

A range sensor provides a distance measurement of the robot's relative to a known location, the origin (x_0, y_0) . If the planar location (x, y) is given, then the distance $d(k)$ of the robot from the origin is decided by

$$d(k) = \sqrt{(x(k) - x_0)^2 + (y(k) - y_0)^2} \quad (8.35)$$

The linear mapping \mathbf{H}_{range} of the range observation is

$$\mathbf{H}_{range} = \frac{\partial h_{range}}{\partial \mathbf{x}} = \frac{\partial d(k)}{\partial \mathbf{x}} = \left[\frac{x - x_0}{d}, \frac{y - y_0}{d}, 0, 0 \right]^T \quad (8.36)$$

8.4.2.2 Speed Measurement

Some sensors measure directly the relative motion between the robot and its environment. Since an object is moving relative to the robot's reference frame, sensors detect relative motion and its speed is estimated. Speed sensors are based on the Doppler effect which utilizes the change of an electromagnetic or sound wave known as the Doppler shift.¹²⁷ The robot's speed $s(k)$ is assumed to be measured by a speed sensor relative to the reference frame.

8.4.2.3 Measurement Equations

The autonomous mobile robot determines its position by using the measurements from the range and speed sensors, but heading angle sensors are not utilized. Then the measurement equation with an additive Gaussian noise vector is expressed by

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k = \begin{bmatrix} d(k) \\ s(k) \end{bmatrix} + \mathbf{v}_k \quad (8.37)$$

where \mathbf{v}_k has a covariance matrix \mathbf{R}_k .

8.4.3 Simulation and Performance Analysis

In this section, the performance of the proposed nonlinear filtering algorithms are demonstrated through a robot navigation problem. A robot attempts to determine its position by utilizing a minimum set of sensors, range and speed sensors.

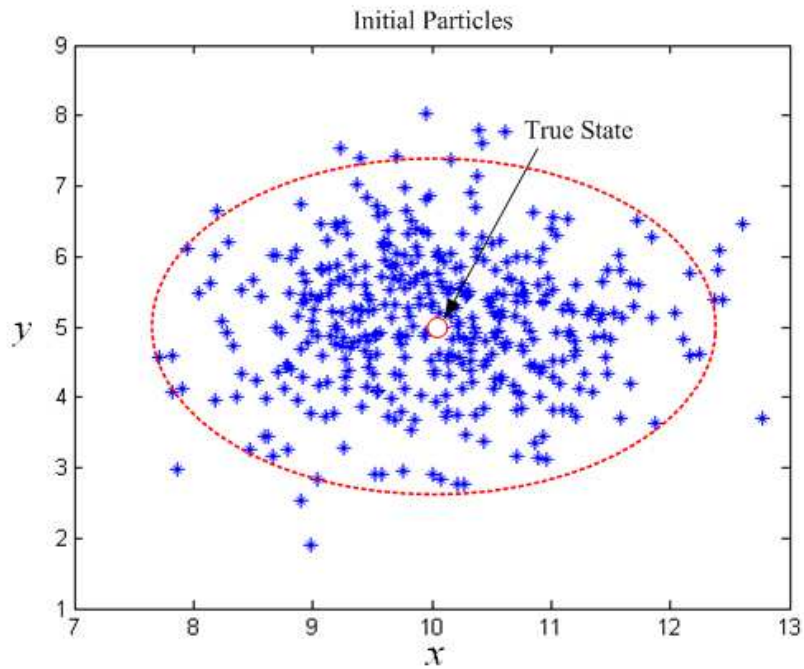


Fig. 8.30 Initial Position Particles at $k = 0$

The parameters values for the robot navigation simulation are given by

$$s_{\max} = 3 \text{ m/sec}, \quad dt = 1.0 \text{ sec} \quad (8.38)$$

An initial true state \mathbf{x}_0 for simulation study is set such that $\mathbf{x}_0 = [10, 5, 3, 0]^T$. The number of samples used is $N = 400$, and initial particles at time $k = 0$ shown in Fig. 8.30 are drawn from the true distribution of $p(\mathbf{x}_0)$. The initial covariance

$\mathbf{P}_0 \in \Re^{4 \times 4}$ used for the filters is assumed to be diagonal such that the position, speed, and heading-angle variances are given by

$$\mathbf{P}_0 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix} \quad (8.39)$$

The discrete-time form of the covariance \mathbf{Q}_k is assumed to be

$$\mathbf{Q}_k = \begin{bmatrix} 0.1^2 & 0 & 0 & 0 \\ 0 & 0.1^2 & 0 & 0 \\ 0 & 0 & 0.5^2 & 0 \\ 0 & 0 & 0 & \left(\frac{\pi}{18}\right)^2 \end{bmatrix} \quad (8.40)$$

where the units are meter (m) and radian (rad). The measurement noise covariance matrix \mathbf{R}_k is given by

$$\mathbf{R}_k = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.01 \end{bmatrix} \quad (8.41)$$

All of the following numerical simulations have an execution time of 30 seconds. The true paths of the planar robot is depicted in Fig. 8.31 for simulation study.

Fig. 8.32 is a plot of the estimated position errors of the robot as a function of time k . As can be seen, the UKF and DDF have degraded convergence performance, because the measurements of the heading angle θ is not updated. The particle filters, the generic PF and PF-MCMC, generated accurate, converged estimates of the planar motion without the heading angle information. Specially, the particle filter with the MCMC modification show the best performance in the state estimation results.

The estimated speed and heading-angle errors are depicted in Fig. 8.33 and Fig. 8.34. The particle filters leads to accurate estimation performance with small

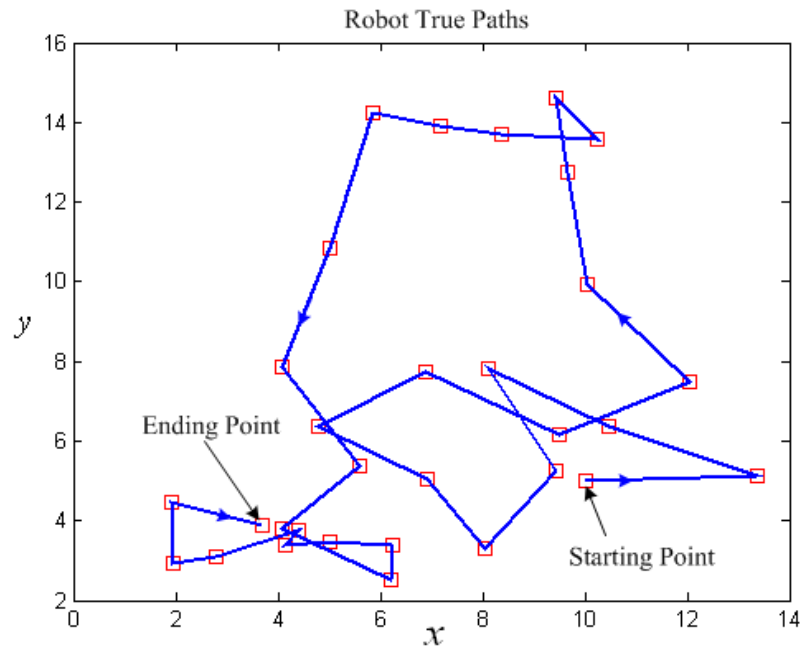


Fig. 8.31 True Paths of Planar Robot Motion

biased errors, whereas the UKF and DDF have degraded estimation results.

The initial particles are updated and resampled during the robot operation time, and the updated particles for the position states at time $k = 30$ is shown in Fig. 8.36 where the true position is located around the updated particles.

The evolution of the probability distributions of the estimated states in the planar motion are shown in Figs. 8.37 and 8.38. In the robot navigation without the heading angle information, the standard minimum mean square estimators, the UKF and DDF, can be misleading because they do not provide enough information about the shape of the distribution. Note that one of the sequential Monte-Carlo estimators is that they provide a complete description of the posterior distribution.

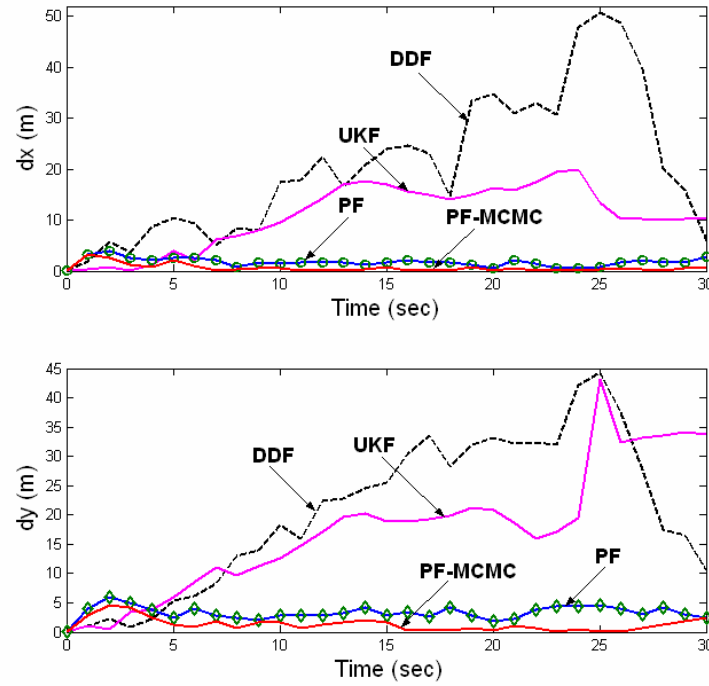


Fig. 8.32 Planar Position Estimation Errors

8.4.4 Conclusion

In this chapter, the performance of the deterministic sigma point filters and statistical particle filters are compared in terms of accuracy and robustness by using the autonomous robot navigation with limited sensor information. The simulation results showed that the sequential Monte-Carlo filters can be efficient alternatives to other nonlinear filtering algorithms when the measurement information is not enough due to limited sensor operation. Specially, the MCMC technique in the particle filtering yields improvements over the sigma point filters (UKF, DDF).

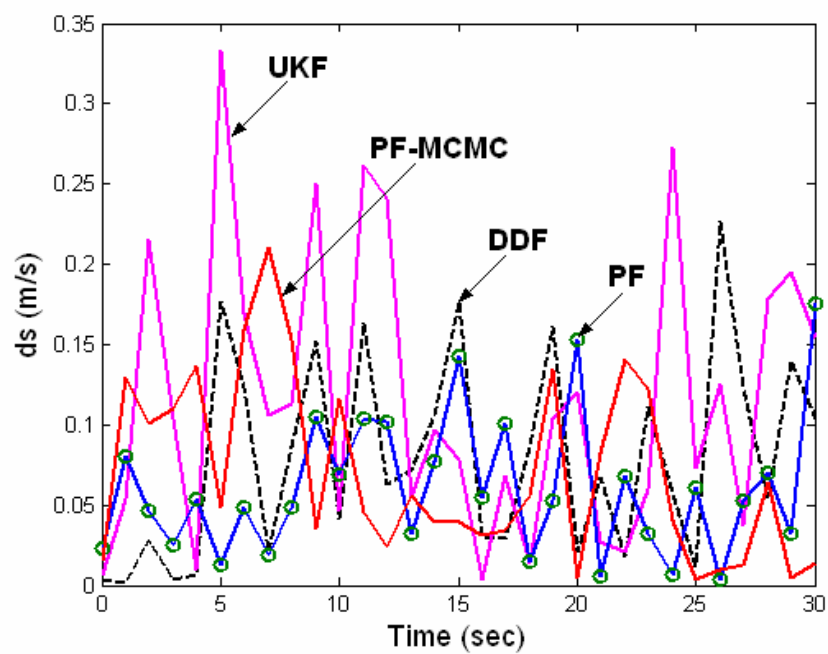


Fig. 8.33 Speed Estimation Errors

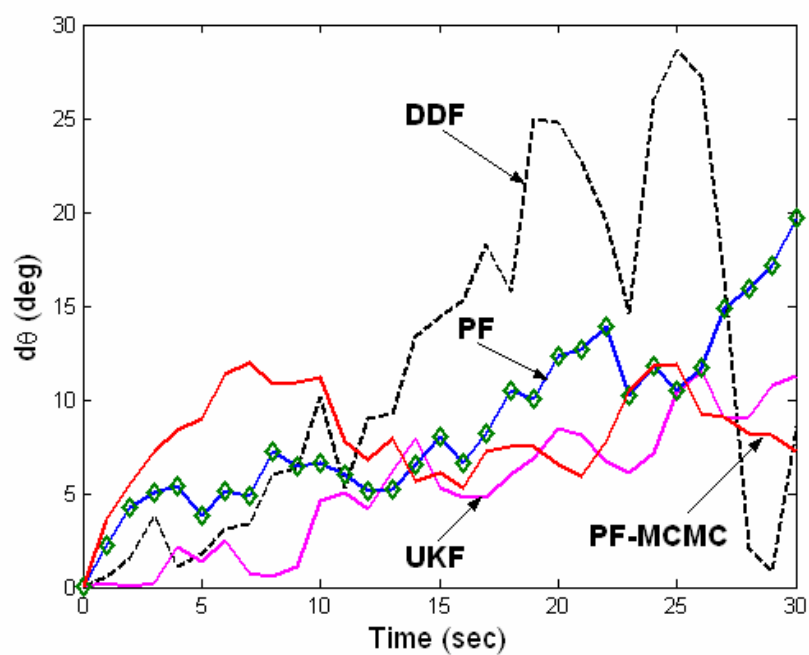


Fig. 8.34 Heading-Angle Estimation Errors

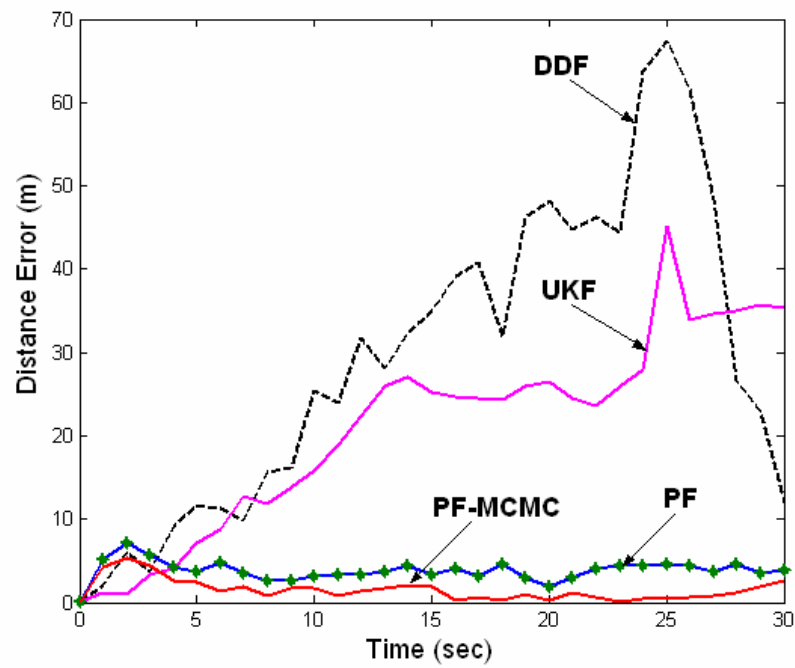


Fig. 8.35 Distance Estimation Errors

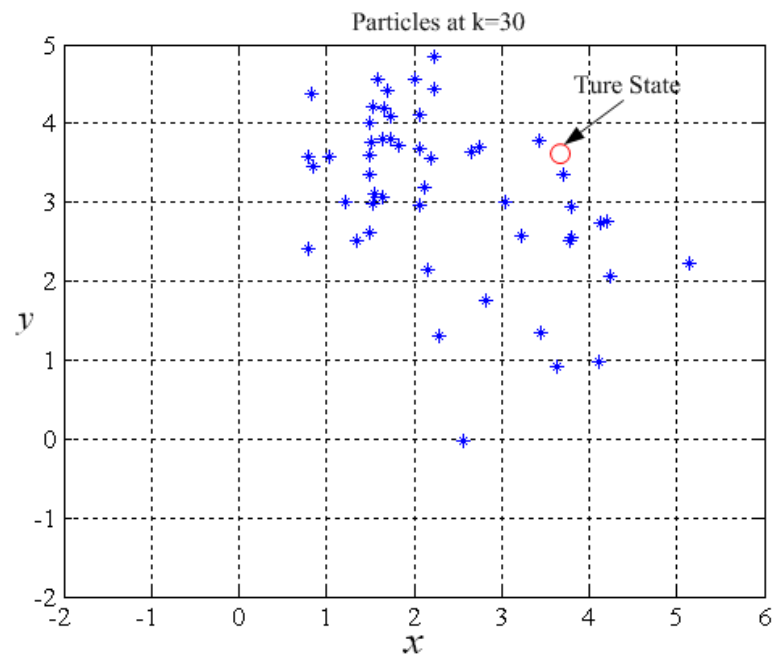


Fig. 8.36 Final Updated Position Particles

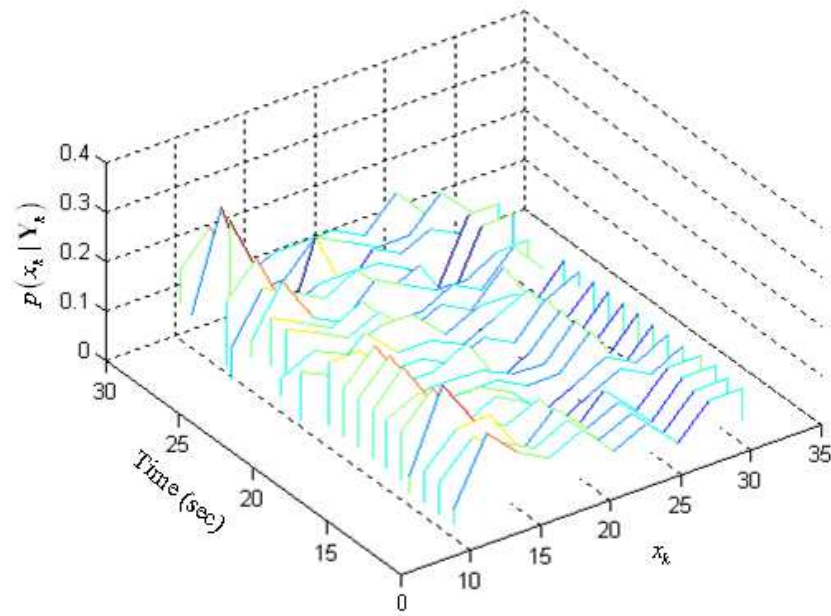


Fig. 8.37 Evolution of Probability Density Function for x State

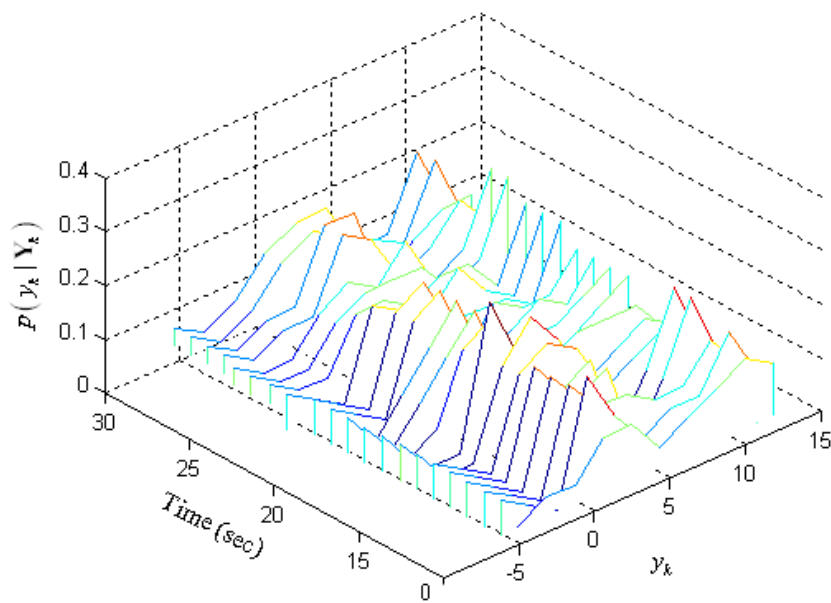


Fig. 8.38 Evolution of Probability Density Function for y State

CHAPTER IX

CONCLUSION AND SUMMARY

Over the past 20 years the extended Kalman filter has become a standard technique in the areas of all engineering fields that need estimation algorithms and related applications. However, recently, the novel and more accurate nonlinear filters have been proposed as more accurate alternatives to the extended Kalman filter within the framework of state and parameter estimation. Like most new algorithms, the new filtering methods were not widely known or understood and their application has been limited. In this work, the state-of-art nonlinear filtering algorithms from the sigma point filters to the particle filters are investigated. We have attempted to unify these differently motivated and derived efficient algorithms under the recursive Bayesian filtering framework.

First, analytical approximate nonlinear filtering algorithms called the sigma point filters (SPFs), that include the unscented Kalman filter (UKF), and the divided difference filter (DDF), were reviewed. The unscented Kalman filter, which belongs to a type of sampling-based filters, is based on the nonlinear transformation called the unscented transformation in which a set of sampled sigma points are used to parameterize the mean and covariance of a probability distribution efficiently. The divided difference filter, which falls into the sampling-based polynomial filters, adopts an alternative linearization method called a central difference approximation in which derivatives are replaced by functional evaluations, leading to an easy expansion of the nonlinear functions to higher-order terms. Secondly, a direct numerical nonlinear filter called the finite difference filter (FDF) was introduced where the state conditional probability density is calculated by applying fast numerical solvers to the Fokker-Planck equation (FPE) in continuous-discrete system models.

For general nonlinear and/or non-Gaussian filtering problems, the sequential Monte Carlo method is investigated. The sequential Monte Carlo can be loosely defined as a simulation-based method that use a Monte Carlo simulation scheme in order to solve on-line estimation and prediction problems. The flexible nature of the Monte Carlo simulations results in these methods often being more adaptive to some features of the complex systems. We investigate recent particle filtering algorithms in a unified way developed independently in various engineering fields.

We have also extended the theoretical understanding of the sigma point filter based techniques and developed new novel algorithms, the adaptive sigma point filters and the divided difference particle filter. The adaptive nonlinear filtering algorithms called the adaptive unscented Kalman filter and the adaptive divided difference filter were derived by integrating the sigma point filters with a new adaptive noise estimator formulated by a numerical optimization. The purpose of the proposed adaptive nonlinear filters and the divided difference particle filters was not only to compensate for the nonlinearity effects neglected from linearization, but also to take into account the system modeling errors by adaptively estimating the noise statistics and unknown parameters. In addition, the local linearized particle filtering algorithm with the combination of the divided difference filter was formulated in order to compensate for the sample degeneracy phenomenon.

For qualitative and quantitative performance analysis among the proposed nonlinear filters, systematic methods for measuring the nonlinearities and optimality of the proposed filters are introduced. For the simulation study, the proposed nonlinear optimal and sub-optimal filtering algorithms with applications to spacecraft orbit estimation and autonomous navigation are investigated.

In the orbit estimation scenario, the performance of the new adaptive nonlinear filters was superior to the standard nonlinear filter, such as the extended Kalman

filter, in terms of the fast convergence and estimation accuracy. In GPS navigation applications, the integrated adaptive dynamic model compensation algorithm provides better accurate estimation results than the geometric solution. The dynamic model compensation estimation plays a role of an adaptive filtering in that it compensates for unknown perturbing acceleration errors in the system model with a limited adaptive capability. In the robot navigation application, the versatility and improved performance of the particle filters over the conventional Gaussian filters, the UKF and DDF was demonstrated.

The advantages of the proposed Bayesian nonlinear filters as well as the adaptive nonlinear filters make these attractive alternatives to the standard extended Kalman filter for efficient state and parameter estimation, not only in satellite orbit determination and navigation, but also other applications.

REFERENCES

- ¹ Jazwinski, A. H., *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.
- ² Ho, Y. C., and Lee, R. C. K., "A Bayesian Approach to Problems in Stochastic Estimation and Control," *IEEE Transactions on Automatic Control*, Vol. 9, October 1964, pp. 333–339.
- ³ Anderson, B. D. O., and Moore, J. B., *Optimal Filtering*, Prentice Hall, Englewood Cliffs, NJ, 1979.
- ⁴ Sorenson, H. W., "On the Development of Practical Nonlinear Filters," *Information Sciences*, Vol. 7, 1974, pp. 253–270.
- ⁵ Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions on ASME, Journal of Basic Engineering*, Vol. 82, March 1960, pp. 34–45.
- ⁶ Tanizaki, H., *Nonlinear Filters: Estimation and Applications*, Springer, New York, 1996.
- ⁷ Gelb, A., (ed.), *Applied Optimal Estimation*, The MIT Press, Cambridge, MA, 1974.
- ⁸ Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., "A New Approach for Filtering Nonlinear Systems," *Proceedings of the American Control Conference*, American Automatic Control Council, Evanston, IL, 1995, pp. 1628–1632.

- ⁹ Julier, S. J., and Uhlmann, J. K., “A New Extension of the Kalman Filter to Nonlinear Systems,” *The 11th International Symposium on Aerospace Defense Sensing, Simulation and Controls*, Orlando, FL, 1997.
- ¹⁰ Schei, T. S., “A Finite-Difference Method for Linearization in Nonlinear Estimation Algorithms,” *Automatica*, Vol. 33, No. 11, November 1997, pp. 2053–2058.
- ¹¹ Ito, K. and Xiong, K., “Gaussian Filters for Nonlinear Filtering Problems,” *IEEE Transactions on Automatic Control*, Vol. 45, No. 5, May 2000, pp. 910–927.
- ¹² Nørgaard, M., Poulsen, N. K., and Ravn, O., “New Developments in State Estimation for Nonlinear Systems,” *Automatica*, Vol. 36, No. 11, November 2000, pp. 1627–1638.
- ¹³ Kastella, K., and Zatezalo, A., “Nonlinear Filtering for Low Elevation Targets in the Presence of Multipath Propagation,” *Proceedings of SPIE Conference, Signal and Data Processing of Small Targets*, Vol. 3373, 1998, pp. 452–459.
- ¹⁴ Kastella, K., “Finite Difference Methods for Nonlinear Filtering and Automatic Target Recognition,” in *Multitarget-Multisensor Tracking: Applications and Advances*, Bar-Shalom, Y., and Blair, W. D., Volume III, Artech House, Norwood, MA, 2000, pp.233–258.
- ¹⁵ Gordon, N., Salmond, D., and Smith, A. F. M., “Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation,” *IEE Proceedings-F*, Vol. 140, No. 2, 1993, pp. 107–113.
- ¹⁶ Chen, R., and Liu, J. S., “Mixture Kalman Filters,” *Journal of Royal Statistical Society*, Vol. 62, Part. 3, 2000, pp. 493–508.

- ¹⁷ Doucet, A., de Freitas, N., and Gordon, N. J., *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001.
- ¹⁸ Gordon, N., Salmond, D., and Smith, A. F. M., "Bayesian State Estimation for Tracking and Guidance using the Bootstrap Filter," *Journal of Guidance, Control and Dynamics*, Vol. 18, No. 6, 1995, pp. 1434–1443.
- ¹⁹ MacCormick, J., and Blake, A., "A Probabilistic Exclusion for Tracking Multiple Objects," *International Journal of Computer Vision*, Vol. 39, No. 1, 2000, pp. 57–71.
- ²⁰ Carpenter, J., Clifford, P., and Fearnhead, P., "Improved Particle Filter for Non-linear Problems," *IEE Proceedings on Radar, Sonar, and Navigation*, Vol. 146, No. 1, February 1999, pp.2–7.
- ²¹ Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T., "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, Vol. 50, Feb. 2002, pp. 174–188.
- ²² Doucet, A., Godsill S., and Andrieu, C., "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistical Computing*, Vol. 10, No. 3, 2000, pp. 197–208.
- ²³ Mehra, R. K., "On the Identification of Variances and Adaptive Kalman Filtering," *IEEE Transactions on Automatic Control*, Vol. 15, No. 2, April 1970, pp. 175–184.
- ²⁴ Sage, A. P., and Husa G. W., "Adaptive Filtering with Unknown Prior Statistics," *10th Joint Automatic Control Conference*, Boulder, CO, August 5-7, 1969.

- ²⁵ Myers, K. A., and Tapley, B. T., “Adaptive Sequential Estimation with Unknown Noise Statistics,” *IEEE Transactions on Automatic Control*, Vol. 21, No. 4, August 1976, pp. 520–523.
- ²⁶ Lee, D.-J., and Alfried, K. T., “Adaptive Sigma Point Filtering for State and Parameter Estimation,” *AIAA/AAS Astrodynamics Specialist Conference*, Providence, RI, August 16-19, 2004.
- ²⁷ Bar-Shalom, Y., Li, X. R., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, Inc, New York, 2001.
- ²⁸ Athans, M., Wishner, R. P., and Bertolini, A., “Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements,” *IEEE Transactions on Automatic Control*, Vol. 13, No. 3, October 1968, pp. 504–514.
- ²⁹ Schwartz, L., and Stear, E. B., “A Computational Comparison of Several Nonlinear Filters,” *IEEE Transactions on Automatic Control*, Vol. AC-13, Feb. 1968, pp. 83–86.
- ³⁰ Henriksen, R., “The Truncated Second-Order Nonlinear Filter Revisited,” *IEEE Transactions on Automatic Control*, Vol. AC-27, Feb. 1982, pp. 247–251.
- ³¹ Lee, D.-J., and Alfried, K. T., “Sigma Point Kalman Filters for Efficient Orbit Estimation,” *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, August 3-7, 2003.
- ³² Beneš, V. E., “New Exact Nonlinear Filters with Large Lie Algebras,” *Systems and Control Letters*, Vol. 5, No. 4, Feb. 1985, pp. 217–221.

- ³³ Lototsky, S., Milulevicius, R., and Rozovskii, B. E., “Nonlinear Filtering Revisited: A Spectral Approach,” *SIAM Journal on Control and Optimization*, Vol. 35, No. 2, March 1997, pp. 435–461.
- ³⁴ Brigo, D., Hanzon, B., and LeGland, F., “A Differential Geometric Approach to Nonlinear Filtering: The Projection Filter,” *IEEE Transactions on Automatic Control*, Vol. 43, No. 2, February 1998, pp. 247–252.
- ³⁵ Daum, F. E., “Exact Finite-Dimensional Nonlinear Filters,” *IEEE Transactions on Automatic Control*, Vol. 37, No. 7, July 1986, pp. 616–622.
- ³⁶ Schmidt, G. C., “Designing Nonlinear Filters based on Daum’s Theory,” *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, March–April 1993, pp. 371–376.
- ³⁷ Beard, R., Kenney, J., Gunther, J., Lawton, J., and Stirling, W., “Nonlinear Projection Filter Based on Galerkin Approximation,” *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, March–April 1999, pp. 258–266.
- ³⁸ Challa, S., and Bar-Shalom, Y., “Nonlinear Filtering Design Using Fokker-Planck-Kolmogorov Probability Density Evolutions,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 36, No. 1, January 2000, pp. 309–315.
- ³⁹ Alspach, D. L., and Sorenson, H. W., “Nonlinear Bayesian Estimation using Gaussian Sum Approximations,” *IEEE Transactions on Automatic Control*, Vol. 17, No. 4, August 1972, pp. 439–448.
- ⁴⁰ Alspach, D. L., “Gaussian Sum Approximations in Nonlinear Filtering and Control,” *Information Sciences*, Vol. 7, 1974, pp. 271–290.

- ⁴¹ Fearnhead, P., *Sequential Monte Carlo Methods in Filter Theory*, Ph.D. Dissertation, Merton College, University of Oxford, Oxford, UK, 1998.
- ⁴² Bucy, R. S., and Senne, K. D., “Realization of Optimum Discrete-Time Nonlinear Estimator,” *1st Symposium on Nonlinear Estimation*, San Diego, 1970, pp. 6–17.
- ⁴³ Bucy, R. S., and Youssef, H., “Nonlinear Filter Representation via Spline Function,” *5th Symposium on Nonlinear Estimation*, San Diego, 1974, pp. 51–60.
- ⁴⁴ Kitagawa, G., “Non-Gaussian State-Space Modeling of Non-Stationary Time Series (with Discussion),” *Journal of the American Statistical Association*, Vol. 82, 1987, pp. 1032–1063.
- ⁴⁵ Kramer, S. C., and Sorenson, H. W., “Recursive Bayesian Estimation using Piece-Wise Constant Approximations,” *Automatica*, Vol. 24, No. 6, 1988, pp. 789–801.
- ⁴⁶ Sorenson, H. W., and Alspach, D. L., “Recursive Bayesian Estimation Using Gaussian Sums,” *Automatica*, Vol. 7, 1971, pp. 465–479.
- ⁴⁷ Ristic, B., Arulampalam, S, and Gordon, N. J., *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Boston, MA, 2004.
- ⁴⁸ Crassidis, J. L., and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, Chapman & Hall/CRC, New York, 2004.
- ⁴⁹ Frost, P. A., and Kailath, T., “An Innovations Approach to Least-Squares Estimation-Part III: Nonlinear Estimation in White Gaussian Noise,” *IEEE Transactions on Automatic Control*, Vol. AC-16, No. 3, June 1971, pp. 217–226.

- ⁵⁰ Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., "A New Method for Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, March 2000, pp. 477–482.
- ⁵¹ Julier, S. J., "The Scaled Unscented Transformation," *Proceedings of the American Control Conference*, American Automatic Control Council, Evanston, IL, May 2002, pp. 1108–1114.
- ⁵² van der Merwe, R., and Wan, E. A., "Efficient Derivative-Free Kalman Filters for Online Learning," *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, April 2001.
- ⁵³ Julier, S. J., "A Skewed Approach for Filtering," *The 12th International Symposium on Aerospace Defense Sensing, Simulation and Controls*, Orlando, FL, 1998.
- ⁵⁴ Grewal, M. S., and Andrews, A. P., *Kalman Filtering: Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1993.
- ⁵⁵ Fung, C.-P., *New Numerical Algorithms for Nonlinear Filtering*, Ph.D. Dissertation, Department of Applied Mathematics, University of Southern California, Los Angeles, CA, December 1995.
- ⁵⁶ Greengard, L., "Fast Algorithms for Classical Physics," *Science*, Vol. 265, No. 5174, August 1994, pp. 909-914.
- ⁵⁷ Challa, S., Bar-Shalom, Y., and Krishnamurthy, V., "Nonlinear Filtering via Generalized Edgeworth Series and Gauss-Hermite Quadrature," *IEEE Transactions on Signal Processing*, Vol. 48, No. 6, June 2000, pp. 1816–1820.

- ⁵⁸ Strikwerda, J. C., *Finite Difference Method and Partial Differential Equations*, Chapman & Hall, New York, 1989.
- ⁵⁹ Liu, J. S., Chen, R., and Logvinenko, T., “A Theoretical Framework for Sequential Importance Sampling with Resampling,” in *Sequential Monte Carlo Methods in Practice*, edited by Doucet, A., de Freitas, N., and Gordon, N. J., Springer, New York, 2001, Chap 11.
- ⁶⁰ Hammersley, J. M., and Morton, K. W., “Poor Man’s Monte Carlo,” *Journal of the Royal Statistical Society B*, Vol. 16, 1954, pp. 23–38.
- ⁶¹ Rosenbluth, M. N., and Rosenbluth, A. W., “Monte Carlo Calculation of the Average Extension of Molecular Chains,” *Journal of Chemical Physics*, Vol. 23, 1955, pp. 356–359.
- ⁶² Handschin, J. E., “Monte Carlo Techniques for Prediction and Filtering of Non-Linear Stochastic Processes,” *Automatica*, Vol. 6, 1970, pp. 555–563.
- ⁶³ Liu, J. S., Chen, R., and Wong, W. H., “Rejection Control and Sequential Importance Sampling,” *Journal of the American Statistical Association*, Vol. 93, No. 443, Sep. 1998, pp. 1022–1031.
- ⁶⁴ Pitt, M. K., and Shephard, N., “Filtering via Simulation: Auxiliary Particle Filters,” *Journal of the American Statistical Association*, Vol. 94, No. 446, 1999, pp. 590–599.
- ⁶⁵ van der Merwe, R., Doucet, A., de Freitas, N., and Wan, E., “The Unscented Particle Filter,” *Technical Report*, CUED/F-INFENG/TR 380, Cambridge University Engineering Department, Cambridge, England, August 2000.

- ⁶⁶ Akashi, H., and Kumamoto, H., “Random Sampling Approach to State Estimation in Switching Environments,” *Automatica*, Vol. 13, 1977, pp. 429–434.
- ⁶⁷ Liu, J. S., and Chen, R., “Sequential Monte Carlo Methods for Dynamic Systems,” *Journal of the American Statistical Association*, Vol. 93, No. 443, Sep. 1998, pp. 1032–1044.
- ⁶⁸ Bergman, N., *Recursive Bayesian Estimation: Navigation and Tracking Applications*, Ph.D. Dissertation, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1999.
- ⁶⁹ Kitagawa, G., “Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models,” *Journal of Computational and Graphical Statistics*, Vol. 5, NO. 1, 1996, pp. 1–25.
- ⁷⁰ Musso, C., Oudjane, N., and LeGland, F., “Improving Regularised Particle Filters,” in *Sequential Monte Carlo Methods in Practice*, edited by Doucet, A., de Freitas, N, and Gordon, N., Springer, New York, 2001, Chap. 12.
- ⁷¹ de Freitas, J. F. G, Niranjana, M., and Gee, A. H., “Hierarchical Bayesian-Kalman Models for Regularisation and ARD in Sequential Learning,” *Technical Report CUED/FINFENG/TR 307*, Cambridge University Engineering Department, University of Cambridge, Cambridge, UK, 1998.
- ⁷² Carlin, B. P., and Polson, N. G., and Stoffer, D. S., “A Monte Carlo Approach to Nonnormal and Nonlinear State-Space Modeling,” *Journal of the American Statistical Association*, Vol. 87, No. 418, 1992, pp. 493–500.
- ⁷³ de Freitas, J. F. G, *Bayesian Methods for Neural Networks*, Ph.D. Dissertation, Cambridge University Engineering Department, University of Cambridge,

Cambridge, UK, 1999.

- ⁷⁴ Murphy, K., and Russell, S., “Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks,” in *Sequential Monte Carlo Methods in Practice*, edited by Doucet, A., de Freitas, N., and Gordon, N. J., Springer, New York, 2001, Chap. 24.
- ⁷⁵ Nordlund, P.-J., *Sequential Monte Carlo Filters and Integrated Navigation*, Thesis, Department of Electrical Engineering, Linköpings University, Linköpings, Sweden, 2002.
- ⁷⁶ Silverman, B. W., *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, Bristol, UK, 1986.
- ⁷⁷ MacEachern, S. N., Clyde, M., and Liu, J. S., “Sequential Importance Sampling for Nonparametric Bayes Models: The Next Generation,” *Canadian Journal of Statistics*, Vol. 27, 1999, pp. 251–267.
- ⁷⁸ Berzuini, C., Best, N. G., Gilks, W. R., and Larissz, C., “Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods,” *Journal of the American Statistics*, Vol. 92, 1997, pp. 1403–1412.
- ⁷⁹ Hastings, W. K., “Monte Carlo Simulation Methods using Markov Chain and their Applications,” *Biometrika*, Vol. 57, No. 1, 1970, pp. 97–109.
- ⁸⁰ Metropolis, N., Rosenblutt, N., Rosenblutt, A. W., Teller, M. N., and Teller, E., “Equations of State Calculations by Fast Computing Machines,” *The Journal of Chemical Physics*, Vol. 21, No. 6, 1953, pp. 1087–1092.
- ⁸¹ Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I., “An Introduction to MCMC for Machine Learning,” *Machine Learning*, Vol. 50, 2003, pp. 5–43.

- ⁸² Geman, S., and Geman, D., “Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, 1984, pp. 731–741.
- ⁸³ Doerschuk, P. C., “Cramer-Rao Bounds for Discrete-Time Nonlinear Filtering Problems,” *IEEE Transactions on Automatic Control*, Vol. 40, No. 8, August 1995, pp.1465-1469.
- ⁸⁴ Tichavsky, P., and Muravchik, C. H., “Posterior Cramer-Rao Bounds for Discrete-Time Nonlinear Filtering,” *IEEE Transactions on Signal Processing*, Vol. 46, No. 5, May 1998, pp.1386-1396.
- ⁸⁵ Denham, W. F., and Pines, S., “Sequential Estimation When Measurement Function Nonlinearity is Comparable to Measurement Error,” *AIAA Journal*, Vol. 4, No. 6, June 1966, pp. 1071–1076.
- ⁸⁶ Lee, D.-J., and Alfriend, K. T., “Precise Real-Time Orbit Estimation Using the Unscented Kalman Filter,” *AAS/AIAA Space Flight Mechanics Meeting*, Ponce, Puerto Rico, Feb. 9-13, 2003.
- ⁸⁷ Maybeck, P., *Stochastic Models, Estimation, and Control*, Vol. 2, Academic Press, New York, 1972.
- ⁸⁸ Powell, T. D., “Automated Tuning of an Extended Kalman Filter Using the Downhill Simplex Algorithm,” *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, AK, August 16-19, 1999, pp. 1029–1039.
- ⁸⁹ Busse, F. D., and How, J. P., “Demonstration of Adaptive Extended Kalman Filter for Low Earth Orbit Estimation Using DGPS,” *Institute of Navigation GPS Meeting*, September 2002.

- ⁹⁰ Jazwinski, A. H., "Adaptive Filtering," *Automatica*, Vol. 5, No. 4, 1969, pp. 475–485.
- ⁹¹ Blanchet, I., Frankignoul, C., and Cane, M., "A Comparison of Adaptive Kalman Filters for a Tropical Pacific Ocean Model," *Monthly Weather Review*, Vol. 125, 1997, pp. 40–58.
- ⁹² Press, W. H., Vetterling, W. H., Teukolsky, S. A., and Flannery B. P., *Numerical Recipes in C*, Cambridge University Press, New York, 1992, pp. 408–412.
- ⁹³ Ingram, D. S., *Orbit Determination in the Presence of Unmodeled Accelerations*, Ph.D. Dissertation, The University of Texas, Austin, TX, August 1970.
- ⁹⁴ Wu, S. C., Yunck, T. P., and Thornton, C. L., "Reduced Dynamic Technique for Precise Orbit Determination of Low Earth Satellite," *AIAA Paper 87-410*, *AAS/AIAA Astrodynamics Specialist Conference*, Kalispell, MT, August 10-13 1987, pp. 101-113.
- ⁹⁵ Myers, K. A., *Filtering Theory Methods and Applications to the Orbit Determination Problem for Near Earth Satellites*, Ph.D. Dissertation, The University of Texas, Austin, TX, January 1974.
- ⁹⁶ Yunck, T. P., Wu, S. C., Wu, J. T., and Thornton, C. L., "Precise Tracking of Remote Sensing Satellites with the Global Positioning System," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 28, No. 1, January 1990, pp. 108-116.
- ⁹⁷ Cruickshank, D. R., *Genetic Model Compensation: Theory and Applications*, Ph.D. Dissertation, Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO, December 1998.

- ⁹⁸ Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, McGraw-Hill, New York, 1997.
- ⁹⁹ Sorenson, H. W., "Least-Squares Estimation: From Gauss to Kalman," *IEEE Spectrum*, July 1970, pp. 63–68.
- ¹⁰⁰ Lee, D.-J. No, T. S., Choi, S.-W., Lee, S.-R., Kim, H.-J., and Alfriend, K. T., "Precise Onboard Ephemeris Propagation Method using CW Frame and Multiple Compressions," *Journal of Guidance, Control and Dynamics*, Vol. 26, No. 5, 2004, pp. 781–785.
- ¹⁰¹ Junkins, J. L., Akella, M. R., and Alfriend, K. T., "Non-Gaussian Error Propagation in Orbital Mechanics," *the Journal of the Astronautical Sciences*, Vol. 44, No. 4, 1996, pp. 541–563.
- ¹⁰² Alfriend, K. T., "A Dynamic Algorithm for the Processing of UCTs," *AAS/AIAA Astrodynamics Specialist Conference*, Sun Valley, ID, Aug. 4–7, 1997, pp. 123–131.
- ¹⁰³ Alfriend, K. T., Akella, M. R., Lee, D.-J., Wilkins, M. P., Frisbee J., and Foster J. L., "Probability of Collision Error Analysis," Paper No. 98-4279, *AIAA/AAS Astrodynamics Specialist Conference*, Boston, MA, Aug. 10–12, 1998.
- ¹⁰⁴ Vallado, D. A., "Accurate Orbit Determination from Short-Arc Dense Observational Data," *The Journal of the Astronautical Sciences*, Vol. 46, No. 2, April–June 1998, pp. 195–213.
- ¹⁰⁵ Bate, R. R., Mueller, D. D., and White, J. E., *Fundamentals of Astrodynamics*, Dover, New York, 1971.

- ¹⁰⁶ Shanklin Jr., R. E., Lee, T., Mallick, M. K., and Cappelari Jr., J. O., "Comparative Studies of Atmospheric Density Models Used for Earth Satellite Orbit Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 7, No. 2, March–April 1984, pp. 235–237.
- ¹⁰⁷ Dowd, D. L., and Tapley, B. D., "Density Models for the Upper Atmosphere," *Celestial Mechanics*, Vol. 20, 1979, pp. 271–295.
- ¹⁰⁸ Barker, W. N., "Space Station Debris Avoidance Study, Final Report," *KSPACE 97-47*, Kaman Sciences, Colorado Springs, CO, January 31, 1997.
- ¹⁰⁹ Rauch, H., "Optimum Estimation of Satellite Trajectories Including Random Fluctuations in Drag," *AIAA Journal*, Vol. 3, 1965, pp. 717–722.
- ¹¹⁰ Montenbruck, O., and Gill, E., *Satellite Orbits*, Springer, New York, 2000.
- ¹¹¹ Hough, M. E., "Improved Performance of Recursive Tracking Filters Using Batch Initialization and Process Noise Adaptation," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 5, September–October 1999, pp. 675–681.
- ¹¹² Tapley, B. D., Schutz, B. E., and Born, G. H., *Statistical Orbit Determination*, Elsevier Academic Press, Burlington, MA, 2004.
- ¹¹³ Grewal, M. S., Weill, L. R. and Andrews, A. P., *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley & Sons, New York, 2001.
- ¹¹⁴ Kaplan, E. D., *Understanding GPS: Principles and Applications*, Artech House, Boston, 1996.
- ¹¹⁵ Brown, R. G., and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, New York, 1997.

- ¹¹⁶ Parkinson, B. W., and Spiker Jr., J. J., *Global Positioning System: Theory and Applications*, Vols 1 and 2, American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 1996.
- ¹¹⁷ Bancroft, S., "An Algebraic Solution of the GPS Equations," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 21, No. 7, Jan. 1985, pp. 56-59.
- ¹¹⁸ Krause, L. O., "A Direct Solution to GPS Type Navigation Equations," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 23, No. 2, March 1987, pp. 225-232.
- ¹¹⁹ Abel, J. S., and Chaffee, J. W., "Existence and Uniqueness of GPS Solutions," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 27, No. 6, November 1991, pp. 952-956.
- ¹²⁰ Cruickshank, D. R., *Algorithms for Autonomous Satellite Navigation Using GPS Measurement*, M.S. Thesis, Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO, April 1994.
- ¹²¹ Mendel, J. M., "Extension of Friedland's Bias Filtering Technique to a Class of Nonlinear Systems, *IEEE Transactions on Automatic Control*," Vol. 21, No. 2, April 1976, pp. 296-298.
- ¹²² Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Taylor & Francis, London, England, 1975.
- ¹²³ Carvalho, H., Del Moral, P., Monin, A., and Salut, G., "Optimal Nonlinear Filtering in GPS/INS Integration," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 33, No. 3, July 1997, pp. 835-850.

- ¹²⁴ Siegwart, R., and Nourbakhsh, I. R., *Introduction to Autonomous Mobile Robots*, The MIT Press, Cambridge, MA, 2004.
- ¹²⁵ Kwok, C. T., *Robust Real-Time Perception for Mobile Robots*, Ph.D. Dissertation, Department of Computer Science and Engineering, University of Washington, Seattle, WA, December 2004.
- ¹²⁶ Leonard, J. E., and Durrant-Whyte, H. F., *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic Publishers, Norwood, MA, 1992.
- ¹²⁷ Borenstein, J., Everett, H.R., and Feng, L., “Where am I? Sensors and Methods for Mobile Robot Positioning,” *Technical Report*, University of Michigan, Ann Arbor, MI, March 1996.

APPENDIX A

PARTIAL DERIVATIVE

The filter acceleration model consists of a gravity term and a drag term. The system sensitivity matrix \mathbf{F} is obtained by taking the partial derivative of the above acceleration functions with respect to the state vector \mathbf{x} . The non-zero terms of \mathbf{F} are listed below in terms of $F_{i,j}$

$$F_{1,4} = F_{2,5} = F_{3,6} = F_{10,11} = 1 \quad (\text{A.1})$$

$$F_{4,1} = \frac{\partial \ddot{x}}{\partial x} = -\frac{\mu}{r^3} \left(1 - 3\frac{x^2}{r^2} + \frac{15}{2}J_2\frac{r_e^2 z^2}{r^4} \left(7\frac{x^2}{r^2} - 1 \right) + \frac{3}{2}J_2\frac{r_e^2}{r^2} \left(1 - 5\frac{x^2}{r^2} \right) \right) + \frac{1}{2}\frac{C_DA}{m}\rho(\dot{x} + \omega_e y) \left(\frac{\omega_e(\dot{y} - \omega_e x)}{v_{rel}} + \frac{v_{rel}x}{Hr} \right) \quad (\text{A.2})$$

$$F_{4,2} = \frac{\partial \ddot{x}}{\partial y} = \frac{3\mu xy}{r^5} \left(1 - \frac{35}{2}J_2\frac{r_e^2 z^2}{r^4} + \frac{5}{2}J_2\frac{r_e^2}{r^2} \right) - \frac{1}{2}\frac{C_DA}{m}\rho \left(v_{rel}\omega_e + (\dot{x} + \omega_e y) \left(\frac{\omega_e(\dot{x} + \omega_e y)}{v_{rel}} - \frac{v_{rel}y}{Hr} \right) \right) \quad (\text{A.3})$$

$$F_{4,3} = \frac{\partial \ddot{x}}{\partial z} = \frac{3\mu xz}{r^5} \left(1 - \frac{35}{2}J_2\frac{r_e^2 z^2}{r^4} + \frac{15}{2}J_2\frac{r_e^2}{r^2} \right) + \frac{1}{2}\frac{C_DA}{m}\rho \frac{v_{rel}z(\dot{x} + \omega_e y)}{Hr} \quad (\text{A.4})$$

$$F_{4,4} = \frac{\partial \ddot{x}}{\partial \dot{x}} = -\frac{1}{2}\frac{C_DA}{m}\rho \left(v_{rel} + \frac{(\dot{x} + \omega_e y)^2}{v_{rel}} \right) \quad (\text{A.5})$$

$$F_{4,5} = \frac{\partial \ddot{x}}{\partial \dot{y}} = -\frac{1}{2}\frac{C_DA}{m}\rho \left(\frac{(\dot{x} + \omega_e y)(\dot{y} - \omega_e x)}{v_{rel}} \right) \quad (\text{A.6})$$

$$F_{4,6} = \frac{\partial \ddot{x}}{\partial \dot{z}} = -\frac{1}{2}\frac{C_DA}{m}\rho \frac{\dot{z}(\dot{x} + \omega_e y)}{v_{rel}} \quad (\text{A.7})$$

$$F_{4,7} = \frac{\partial \ddot{x}}{\partial \mu} = -\frac{x}{r^3} \left(1 - \frac{3}{2}J_2\frac{r_e^2}{r^2} \left(5\frac{z^2}{r^2} - 1 \right) \right) \quad (\text{A.8})$$

$$F_{4,8} = \frac{\partial \ddot{x}}{\partial J_2} = \frac{3}{2}\mu x \frac{r_e^2}{r^5} \left(5\frac{z^2}{r^2} - 1 \right) \quad (\text{A.9})$$

$$F_{4,9} = \frac{\partial \ddot{x}}{\partial C_d} = -\frac{1}{2}\frac{A}{m}\rho v_{rel}(\dot{x} + \omega_e y) \quad (\text{A.10})$$

$$F_{5,1} = \frac{\partial \ddot{y}}{\partial x} = \frac{3\mu y x}{r^5} \left(1 - \frac{35}{2} J_2 \frac{r_e^2 z^2}{r^4} + \frac{5}{2} J_2 \frac{r_e^2}{r^2} \right) - \frac{1}{2} \frac{C_D A}{m} \rho \left(-v_{rel} \omega_e - (\dot{y} - \omega_e x) \left(\frac{\omega_e (\dot{y} - \omega_e x)}{v_{rel}} + \frac{v_{rel} x}{Hr} \right) \right) \quad (A.11)$$

$$F_{5,2} = \frac{\partial \ddot{y}}{\partial y} = -\frac{\mu}{r^5} \left(1 - \frac{3y^2}{r^2} + \frac{15}{2} J_2 \frac{r_e^2 z^2}{r^4} \left(7 \frac{y^2}{r^2} - 1 \right) + \frac{3}{2} J_2 \frac{r_e^2}{r^2} \left(1 - 5 \frac{y^2}{r^2} \right) \right) - \frac{1}{2} \frac{C_D A}{m} \rho \left((\dot{y} - \omega_e x) \left(\frac{\omega_e (\dot{x} + \omega_e y)}{v_{rel}} - \frac{v_{rel} y}{Hr} \right) \right) \quad (A.12)$$

$$F_{5,3} = \frac{\partial \ddot{y}}{\partial z} = \frac{3\mu y z}{r^5} \left(1 - \frac{35}{2} J_2 \frac{r_e^2 z^2}{r^4} + \frac{15}{2} J_2 \frac{r_e^2}{r^2} \right) + \frac{1}{2} \frac{C_D A}{m} \rho \frac{v_{rel} z (\dot{y} - \omega_e x)}{Hr} \quad (A.13)$$

$$F_{5,4} = \frac{\partial \ddot{y}}{\partial \dot{x}} = -\frac{1}{2} \frac{C_D A}{m} \rho \frac{(\dot{y} - \omega_e x) (\dot{x} + \omega_e y)}{v_{rel}} \quad (A.14)$$

$$F_{5,5} = \frac{\partial \ddot{y}}{\partial \dot{y}} = -\frac{1}{2} \frac{C_D A}{m} \rho \left(v_{rel} + \frac{(\dot{y} - \omega_e x)^2}{v_{rel}} \right) \quad (A.15)$$

$$F_{5,6} = \frac{\partial \ddot{y}}{\partial \dot{z}} = -\frac{1}{2} \frac{C_D A}{m} \rho (\dot{y} - \omega_e x) \frac{\dot{z}}{v_{rel}} \quad (A.16)$$

$$F_{5,7} = \frac{\partial \ddot{y}}{\partial \mu} = -\frac{y}{r^3} \left(1 - \frac{3}{2} J_2 \frac{r_e^2}{r^2} \left(5 \frac{z^2}{r^2} - 1 \right) \right) \quad (A.17)$$

$$F_{5,8} = \frac{\partial \ddot{y}}{\partial J_2} = \frac{3}{2} \mu y \frac{r_e^2}{r^5} \left(5 \frac{z^2}{r^2} - 1 \right) \quad (A.18)$$

$$F_{5,9} = \frac{\partial \ddot{y}}{\partial C_d} = -\frac{1}{2} \frac{A}{m} \rho v_{rel} (\dot{y} - \omega_e x) \quad (A.19)$$

$$F_{6,1} = \frac{\partial \ddot{z}}{\partial x} = \frac{3\mu zx}{r^5} \left(1 + \frac{15}{2} J_2 \frac{r_e^2}{r^2} - \frac{35}{2} J_2 \frac{r_e^2 z^2}{r^4} \right) + \frac{1}{2} \frac{C_D A}{m} \rho \dot{z} \left(\frac{\omega_e (\dot{y} - \omega_e x)}{v_{rel}} + \frac{v_{rel} x}{Hr} \right) \quad (\text{A.20})$$

$$F_{6,2} = \frac{\partial \ddot{z}}{\partial y} = \frac{3\mu zy}{r^5} \left(1 + \frac{15}{2} J_2 \frac{r_e^2}{r^2} - \frac{35}{2} J_2 \frac{r_e^2 z^2}{r^4} \right) - \frac{1}{2} \frac{C_D A}{m} \rho \dot{z} \left(\frac{\omega_e (\dot{x} + \omega_e y)}{v_{rel}} - \frac{v_{rel} y}{Hr} \right) \quad (\text{A.21})$$

$$F_{6,3} = \frac{\partial \ddot{z}}{\partial z} = -\frac{\mu}{r^3} \left(1 - \frac{3z^2}{r^2} + 3J_2 \frac{r_e^2}{r^2} \left(\frac{3}{2} - \frac{15z^2}{r^2} + \frac{35}{2} \frac{z^4}{r^4} \right) \right) + \frac{1}{2} \frac{C_D A}{m} \rho \frac{v_{rel} z \dot{z}}{Hr} \quad (\text{A.22})$$

$$F_{6,4} = \frac{\partial \ddot{z}}{\partial \dot{x}} = -\frac{1}{2} \frac{C_D A}{m} \rho \dot{z} \frac{(\dot{x} + \omega_e y)}{v_{rel}} \quad (\text{A.23})$$

$$F_{6,5} = \frac{\partial \ddot{z}}{\partial \dot{y}} = -\frac{1}{2} \frac{C_D A}{m} \rho \dot{z} \frac{(\dot{y} - \omega_e x)}{v_{rel}} \quad (\text{A.24})$$

$$F_{6,6} = \frac{\partial \ddot{z}}{\partial \dot{z}} = -\frac{1}{2} \frac{C_D A}{m} \rho \left(\frac{\dot{z}^2}{v_{rel}} + v_{rel} \right) \quad (\text{A.25})$$

$$F_{6,7} = \frac{\partial \ddot{z}}{\partial \mu} = -\frac{z}{r^3} \left(1 + \frac{3}{2} J_2 \frac{r_e^2}{r^2} \left(3 - 5 \frac{z^2}{r^2} \right) \right) \quad (\text{A.26})$$

$$F_{6,8} = \frac{\partial \ddot{z}}{\partial J_2} = -\frac{3}{2} \mu z \frac{r_e^2}{r^5} \left(3 - 5 \frac{z^2}{r^2} \right) \quad (\text{A.27})$$

$$F_{6,9} = \frac{\partial \ddot{z}}{\partial C_d} = -\frac{1}{2} \frac{A}{m} \rho v_{rel} \dot{z} \quad (\text{A.28})$$

Now, the partial derivatives \mathbf{H} of the measurement equation \mathbf{h} with respect to the state vector \mathbf{x} is represented with the observations made from a ground-based radar sensor system. First, *Range* is defined by

$$\rho = \sqrt{\rho_u^2 + \rho_e^2 + \rho_n^2} \quad (\text{A.29})$$

$$H_{1,1} = \frac{d\rho}{dx} = \frac{\partial \rho}{\partial \rho_u} \left(\frac{\partial \rho_u}{\partial x} \right) + \frac{\partial \rho}{\partial \rho_e} \left(\frac{\partial \rho_e}{\partial x} \right) + \frac{\partial \rho}{\partial \rho_n} \left(\frac{\partial \rho_n}{\partial x} \right) \quad (\text{A.30})$$

and, it becomes

$$H_{1,1} = \frac{1}{\rho} \left\{ \rho_u \left(\frac{\partial \rho_u}{\partial x} \right) + \rho_e \left(\frac{\partial \rho_e}{\partial x} \right) + \rho_n \left(\frac{\partial \rho_n}{\partial x} \right) \right\} \quad (\text{A.31})$$

$$= \frac{1}{\rho} \{ \rho_u (\cos \phi \cos \theta) + \rho_e (-\sin \theta) + \rho_n (-\sin \phi \cos \theta) \} \quad (\text{A.32})$$

In a similar way, the followings are obtained

$$H_{1,2} = \frac{d\rho}{dy} = \frac{1}{\rho} \left\{ \rho_u \left(\frac{\partial \rho_u}{\partial y} \right) + \rho_e \left(\frac{\partial \rho_e}{\partial y} \right) + \rho_n \left(\frac{\partial \rho_n}{\partial y} \right) \right\} \quad (\text{A.33})$$

$$= \frac{1}{\rho} \{ \rho_u (\cos \phi \sin \theta) + \rho_e (\cos \theta) + \rho_n (-\sin \phi \sin \theta) \} \quad (\text{A.34})$$

$$H_{1,3} = \frac{d\rho}{dz} = \frac{1}{\rho} \left\{ \rho_u \left(\frac{\partial \rho_u}{\partial z} \right) + \rho_e \left(\frac{\partial \rho_e}{\partial z} \right) + \rho_n \left(\frac{\partial \rho_n}{\partial z} \right) \right\} \quad (\text{A.35})$$

$$= \frac{1}{\rho} \{ \rho_u (\sin \phi) + \rho_n (\cos \phi) \} \quad (\text{A.36})$$

Azimuth is defined by

$$az = \tan^{-1}(\beta) = \tan^{-1} \left(\frac{\rho_e}{\rho_n} \right) \quad (\text{A.37})$$

$$H_{2,1} = \frac{\partial az}{\partial x} = \frac{\partial az}{\partial \beta} \frac{\partial \beta}{\partial x} \quad (\text{A.38})$$

$$H_{2,2} = \frac{\partial az}{\partial y} = \frac{\partial az}{\partial \beta} \frac{\partial \beta}{\partial y} \quad (\text{A.39})$$

$$H_{2,3} = \frac{\partial az}{\partial z} = \frac{\partial az}{\partial \beta} \frac{\partial \beta}{\partial z} \quad (\text{A.40})$$

where

$$\frac{\partial az}{\partial \beta} = \frac{1}{1 + \beta^2} \quad (\text{A.41})$$

and

$$\frac{\partial \beta}{\partial x} = \frac{1}{\rho_n} \left(\frac{\partial \rho_e}{\partial x} \right) - \frac{\rho_e}{\rho_n^2} \left(\frac{\partial \rho_n}{\partial x} \right) \quad (\text{A.42})$$

$$\frac{\partial \beta}{\partial y} = \frac{1}{\rho_n} \left(\frac{\partial \rho_e}{\partial y} \right) - \frac{\rho_e}{\rho_n^2} \left(\frac{\partial \rho_n}{\partial y} \right) \quad (\text{A.43})$$

$$\frac{\partial \beta}{\partial z} = \frac{1}{\rho_n} \left(\frac{\partial \rho_e}{\partial z} \right) - \frac{\rho_e}{\rho_n^2} \left(\frac{\partial \rho_n}{\partial z} \right) \quad (\text{A.44})$$

Then, it reduces to

$$H_{2,1} = \frac{1}{1 + \beta^2} \left(\frac{1}{\rho_n} \right) \left\{ (-\sin \theta) + \frac{\rho_e}{\rho_n} (\sin \phi \cos \theta) \right\} \quad (\text{A.45})$$

$$H_{2,2} = \frac{1}{1 + \beta^2} \left(\frac{1}{\rho_n} \right) \left\{ (\cos \theta) + \frac{\rho_e}{\rho_n} (\sin \phi \sin \theta) \right\} \quad (\text{A.46})$$

$$H_{2,3} = \frac{\partial az}{\partial z} = \frac{1}{1 + \beta^2} \left(\frac{1}{\rho_n} \right) \left\{ -\frac{\rho_e}{\rho_n} (\cos \phi) \right\} \quad (\text{A.47})$$

Elevation is defined by

$$el = \sin^{-1}(\gamma) = \tan^{-1}(\psi) \quad (\text{A.48})$$

where

$$\gamma = \frac{\rho_u}{\rho}, \quad \psi = \frac{\rho_u}{\sqrt{\rho_e^2 + \rho_n^2}} \quad (\text{A.49})$$

$$H_{3,1} = \frac{\partial el}{\partial x} = \frac{\partial el}{\partial \gamma} \frac{\partial \gamma}{\partial x} \quad (\text{A.50})$$

$$H_{3,2} = \frac{\partial el}{\partial y} = \frac{\partial el}{\partial \gamma} \frac{\partial \gamma}{\partial y} \quad (\text{A.51})$$

$$H_{3,3} = \frac{\partial el}{\partial z} = \frac{\partial el}{\partial \gamma} \frac{\partial \gamma}{\partial z} \quad (\text{A.52})$$

where

$$\frac{\partial el}{\partial \gamma} = \frac{1}{\sqrt{1 - \gamma^2}} \quad (\text{A.53})$$

$$\frac{\partial \gamma}{\partial x} = \frac{1}{\rho} \left(\frac{\partial \rho_u}{\partial x} \right) - \frac{\rho_u}{\rho^2} \left(\frac{\partial \rho}{\partial x} \right) \quad (\text{A.54})$$

$$\frac{\partial \gamma}{\partial y} = \frac{1}{\rho} \left(\frac{\partial \rho_u}{\partial y} \right) - \frac{\rho_u}{\rho^2} \left(\frac{\partial \rho}{\partial y} \right) \quad (\text{A.55})$$

$$\frac{\partial \gamma}{\partial z} = \frac{1}{\rho} \left(\frac{\partial \rho_u}{\partial z} \right) - \frac{\rho_u}{\rho^2} \left(\frac{\partial \rho}{\partial z} \right) \quad (\text{A.56})$$

and becomes

$$H_{3,1} = \frac{\partial el}{\partial x} = \frac{1}{\sqrt{1-\gamma^2}} \left(\frac{1}{\rho} \right) \left\{ (\cos \phi \cos \theta) - \frac{\rho_u}{\rho} \left(\frac{\partial \rho}{\partial x} \right) \right\} \quad (\text{A.57})$$

$$H_{3,2} = \frac{1}{\sqrt{1-\gamma^2}} \left(\frac{1}{\rho} \right) \left\{ (\cos \phi \sin \theta) - \frac{\rho_u}{\rho} \left(\frac{\partial \rho}{\partial y} \right) \right\} \quad (\text{A.58})$$

$$H_{3,3} = \frac{1}{\sqrt{1-\gamma^2}} \left(\frac{1}{\rho} \right) \left\{ (\sin \phi) - \frac{\rho_u}{\rho} \left(\frac{\partial \rho}{\partial z} \right) \right\} \quad (\text{A.59})$$

where $\partial \rho / \partial x$, $\partial \rho / \partial y$, and $\partial \rho / \partial z$ are obtained in the range partials.

APPENDIX B

REFERENCE COORDINATE SYSTEMS

Transformation between ECEF and Geodetic Coordinate Systems

The relation between ECEF Cartesian coordinates and ECEF geodetic coordinates is given by

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (N + h) \cos \phi_{gd} \cos \lambda \\ (N + h) \cos \phi_{gd} \sin \lambda \\ (N(1 - e^2) + h) \sin \phi_{gd} \end{pmatrix} \quad (\text{B.1})$$

where N is the radius of curvature of the ellipsoid and e is the eccentricity of the Earth, and they are given by respectively

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi_{gd}}} \quad (\text{B.2})$$

$$e^2 = 2f - f^2 \quad (\text{B.3})$$

The quantity a is the length of semi-major axis of the Earth, which is the mean equatorial radius of the Earth, and f is the flattening parameter given by

$$a = 6378.13649 \text{ km} \quad (\text{B.4})$$

$$f = \frac{(a - b)}{a} = \frac{1}{298.257222101} = 0.003352813178 \quad (\text{B.5})$$

where b is the semi-minor axis of the Earth, also called the polar axis

$$b = 6356.7516005 \text{ km} \quad (\text{B.6})$$

APPENDIX C

TIME SYSTEM

The purpose of this appendix illustrates the computation of the local sidereal time $\theta(t) = \theta_{LST}(t)$ given by

$$\theta_{LST}(t) = \theta_{GST}(t) + \lambda \quad (C.1)$$

where λ is the longitude of the observer location, and $\theta_{GST}(t)$ is the Greenwich sidereal time (GST)

$$\theta_{GST}(t) = \theta_{GST0} + \omega_{\oplus} UT_1 \quad (C.2)$$

where ω_{\oplus} is the Earth's mean angular velocity

$$\begin{aligned} \omega_{\oplus} &= 7.292115 \times 10^{-5} \pm 1.5 \times 10^{-12} \text{ (rad/s)} \\ &= 0.250684477337^{\circ} \text{ (deg / min)} \end{aligned} \quad (C.3)$$

and UT_1 is the universal time in solar seconds. θ_{GST0} is the Greenwich mean sidereal time at midnight, (0 *hour*, 0 *min*, 0 *sec*) in UT_1 . The calculation of θ_{GST0} at a desired time is obtained by using the epoch of J2000 as a basis (Astronomical Almanac, 1984, S15)

$$\begin{aligned} \theta_{GST0} &= 100.4606184^{\circ} + 36000.77005361 T_{UT_1} \\ &\quad + 0.00038793 T_{UT_1}^2 - 2.6 \times 10^{-8} T_{UT_1}^3 \text{ (deg)} \end{aligned} \quad (C.4)$$

$$\begin{aligned} \theta_{GST0} &= 1.753368560 + 628.3319706889 T_{UT_1} + 6.7707 \times 10^{-6} T_{UT_1}^2 \\ &\quad - 4.5 \times 10^{-10} T_{UT_1}^3 \text{ (rad)} \end{aligned} \quad (C.5)$$

where T_{UT_1} is the number of Julian centuries elapsed from the epoch J2000, and is calculated by

$$T_{UT_1} = \frac{JD_0 - 2451545.0}{36525} \quad (C.6)$$

JD_0 is the Julian day numbers for the date of interest for the epoch J2000 and is simply the integer part of the Julian date, (JD), i.e., the JD at 0 h 0 min 0 s of the day. It is computed by

$$JD_0 = 367 \times (year) - INT \left\{ \frac{7 \left\{ yr + INT \left(\frac{month+9}{12} \right) \right\}}{4} \right\} \\ + INT \left(\frac{275 \times month}{9} \right) + day + 1721013.5 \quad (C.7)$$

where the $year$ must be four digits and the INT denotes real truncation. If $\theta_{GST}(t_{ref}) = \theta_{GST0}$, then, the local sidereal time, θ_{LST} , can be written by

$$\theta_{LST}(t) = \theta_{GST}(t_{ref}) + \omega_{\oplus}(t - t_{ref}) + \lambda \quad (C.8)$$

Julian date (JD) is the continuous amount of time measured in days from the epoch January 1, 4713 B.C., 1200, and can be computed by

$$JD = 367 \times (year) - INT \left\{ \frac{7 \{ year + INT \left(\frac{month+9}{12} \right) \}}{4} \right\} + \\ INT \left(\frac{275 \times month}{9} \right) + day + 1721013.5 + \frac{\left\{ \frac{(sec/60)}{60} + hour \right\}}{24} \quad (C.9)$$

VITA

Deok-Jin Lee was born in Iksan, Chonbuk, South Korea on May 12, 1972, the son of Suk-Bun Yun and Bong-Hyun Lee. He graduated from Deokjin High School in February 1999. The following March he entered Chonbuk National University in Jeonju, Chonbuk, South Korea and he received a Bachelor of Science degree in Aerospace Engineering in February, 1996. In September of 1997 he entered graduate school of Texas A&M University, College Station, TX, and he received a Master of Science degree in Aerospace Engineering in August, 1999. In December of 2000 he was employed by Korea Science and Engineering Foundation (KOSEF) as an intern researcher. He joined the Texas A&M University for Aerospace Engineering Ph.D. program, and received his Ph.D. in May, 2005.

In May 2003 he was awarded one of the *International Education Study Grants* by Texas A&M University and in August 2003 he was awarded *The John V. Breakwell Student Award* by American Astronautical Society (AAS).

His permanent address is: Wang-Gung Myen, Kwang-Arm Ri, 255-12 Iksan, Chonbuk, South Korea.