A NEW NAVIGATION FILTER

BENLIN XU

May 1996

TECHNICAL REPORT
NO. 182

A NEW NAVIGATION FILTER

Benlin Xu

Department of Geodesy and Geomatics Engineering
University of New Brunswick
P.O. Box 4400
Fredericton, N.B.
Canada
E3B 5A3

May 1996

PREFACE

This technical report is a reproduction of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Geodesy and Geomatics Engineering, January 1996. The research was supervised by Dr. Petr Vaníček, and funding was provided by the University of New Brunswick.

As with any copyrighted material, permission to reprint or quote extensively from this report must be received from the author. The citation to this work should appear as follows:

Xu, Benlin (1996). *A New Navigation Filter*. Ph.D. dissertation, Department of Geodesy and Geomatics Engineering, Technical Report No. 182, University of New Brunswick, Fredericton, New Brunswick, Canada, 111 pp.

Abstract

This dissertation describes a new self-learning navigation filter associated with probability space and non-Newtonian dynamics. This new filter relies basically on the information contained in measurements on the vehicle: position fixes, velocities and their error statistics. The basic idea behind this new navigation filter is twofold: (1) A cluster of the observed position fixes contains true kinematic information about the vehicle in motion, (2) A motion model of the vehicle associated with the error statistics of the position fixes should be able to get, to a large extent, the information out of the measurements for use. We base the new filter on an analogy. We consider the statistical confidence region of every position fix as "source" tending to "attract" the undetermined trajectory to pass through this region. With these position fixes and their error statistics, a virtual potential field is constructed in which an imaginary mass particle moves. To make the new filter flexible and responsive to a changing navigation environment, we leave some parameters free and let the filter determine their values, using a sequence of observations and the criterion of least squares of the observation errors. We show that the trajectory of the imaginary particle can well represent the real track of the vehicle.

The new navigation filter has been tested with both simulated and real navigation data, as an estimator, predictor, smoother and blunder detector. Its ability to accept navigator's intervention has also been tested. Compared with the Kalman filter, the new filter requires the uncertainties of observations to be known only relatively (cofactor matrix) and is able to offer a better navigation when the vehicle is under dynamic maneuvers and the data rate is small, but with a slower processing speed.

Contents

List of Tables

List of Figures

Acknowledgements

I wish to thank my supervisor, Prof. P. Vaníček, for his guidance, constant support and encouragement throughout my study and this research. I have benefited from his vast wealth of knowledge and efficient scientific research manner. I am truly grateful to Prof. P. Vaníček and The University of New Brunswick (UNB) for the financial assistance in my study and research in the University.

Thanks also to the other two members of my supervisory committee, Prof. D. E. Wells and Prof. A. Kleusberg, for their suggestions and criticism on this research, and to the other faculty members and the staff in Department of Geodesy and Geomatics Engineering for their helps, and to Mr. D. W. Dodd for offering me real navigation data, as well as to Mr. R. Hanson for his help in correcting grammatical errors in the dissertation.

I would like to extend my thanks to the members of the Examining Board. They are Dr. P. Vaníček, Dr. D. E. Wells, Dr. B. Tupper (Dept. of Mathematics and Statistics, UNB), Dr. R. Mureika (Dept. of Mathematics and Statistics, UNB) and Dr. M. E. Cannon (the external examiner, Dept. of Geomatics Engineering, The University of Calgary). Their constructive recommendations have improved the final version of this dissertation.

Last, I thank my parents and my wife, Ling Zheng, for their understanding, encouragement and love. With my beloved son, Sheng, they enlighten my life.

Chapter 1

introduction

Navigation deals with the kinematics of vehicles [Anderson, 1966; Wells, 1976]. The three classical functions of time in kinematics are motion, velocity and acceleration, although higher order time derivatives may be significant in some cases. Thus the ten basic navigation parameters are the three components of each of these three vectors, plus a time tag. Usually we must be satisfied with a subset of these ten parameters. In this dissertation, we assume that when a vehicle is traveling, instantaneous position fixes and the corresponding error statistics can be obtained. Our purpose here is to estimate the present position of the vehicle and predict its position at the next time instant.

1.1 navigation systems and requirements for precise navigation filters

Existing navigation systems available to navigators include classical ground-based radio positioning such as Loran, Tacan, Omega; satellite positioning such as GPS and Transit; and self-contained dead-reckoning systems such as inertial positioning,

gyrocompass and ship's log (used at sea). The radio positioning and satellite positioning can provide accurate and bounded long-term position information, while the self-contained inertial positioning can provide only accurate short-term position information (due to uncompensated gyro drift, all inertial navigation systems exhibit an unbounded position error). Each system has its own advantages and disadvantages. Descriptions of these can be found in many articles; for example, Wells and Grant [1981], Enge and McCullough [1988], Dove and Miller [1989], Bachri [1990] and Napier [1990].

It seems that satellite navigation systems will become a utility with a much wider range of applications than classical ground-based radio navigation aids. User requirements for satellite navigation systems can generally be satisfied by the following properties of the system: (1) accuracy (of centimetre level), (2) availability (whenever and whereever in the world), (3) integrity (blunder detection being possible) and (4) continuity (positioning rate of 2 Hz) of function; but the use of ground-based navigation system like Loran-C may continue for a rather long time [Stich, et al., 1994].

Table 1.1 gives an overview of applications which require precise velocity and position in exploration geophysics and resource mapping [Schwarz, et al. 1989], and Table 1.2 gives the requirements for navigation in agriculture [Abidin, 1993].

Table 1.3 gives the accuracies of some classical ground-based navigation services [Ackroyd and Lorimer, 1990] and we see that they are not able to meet the above requirements. Differential GPS can offer a positioning accuracy of 5 metres. The differential GPS with carrier-smoothed code can reach an accuracy level of 1 metre. With ambiguity-resolved carrier phase it can give a positioning accuracy at the centimetre level. Table 1.4 shows data processing accuracy of PNAV software [Ashtech, 1993], precise differential GPS navigation and surveying software, a typical commercial software. With the development of science and technology, we can expect more accurate positioning in future. PNAV is able to resolve integer ambiguity on-the-fly

Table 1.1: RMS accuracies required for various precise trajectory applications

TASK	RMS accuracy required for position and velocity	
	position (m)	velocity (cm/s)
gravity (sea)	20	< 10
gravity gradiometry (air)	20	< 10
relative geoid	1	10
3D seismic (land, sea)	1 - 3	50
aeromagnetics	1	30
resource mapping		
1:50 000	2	100
1:20 000	0.5	25
1:10 000	0.1	5

Table 1.2: Requirements for navigation in agriculture

kind of navigation	examples of tha tasks	required accuracy
rough navigation of the vehicle	soil sample acquisition detection of tram lines	± 1 m
navigation of tractor and harvest machine	mineral fertilizer spreading liquid manure spreading solid manure spreading application of pesticides soil cultivation	± 10 cm
tractor implement guidance	drilling hoeing plowing	± 1 cm

Table 1.3: Accuracies of some classical ground-based navigation services

navigation service	accuracy
Omega	2-3 nautical miles (95% probability level)
Loran-C	100-500 metres dependent on geometry and range
	(95% level)
Decca	50-200 metres in good-to-fair geometry

Table 1.4: PNAV data processing accuracy, PDOP < 4.0

processing mode	accuracy (m)
unsmoothed C/A-code pseudo-range	3-5
smoothed C/A-code pseudo-range	1-3
smoothed P-code pseudo-range	0.2-1
C/A-code pseudo-range + carrier phase, float integer ambiguity	1-3 in first 2-10 minutes, 0.1-1.0 thereafter. Best results (0.05-0.3 overall) can be achieved by forward and backward processing
pseudo-ranges + carrier phase, float integer ambiguity	0.5-2 in first 2-4 minutes, 0.1-0.5 thereafter. Best results (0.05-0.3 overall) can be achieved by forward and backward processing
pseudo-ranges and carrier phase, fixed integer ambiguity	0.01-0.1 when the ambiguities are fixed

(carrying out initialization in kinematic mode [Hofmann-Wellenhof, et al., 1992]). But in order to do this in a short period, the following requirements have to be met:

(1) The dual frequency observables must be available.

(2) At least 5 satellite are used.

(3) PDOP < 5.0.

(4) The baseline separation, the distance between base station and rover receiver, must not be over 10 kilometers.

Even so, the time to resolve integer ambiguity varies from 20 seconds to several minutes, depending on the number of satellites, satellite geometry, receiver noise level and data collection interval. Elimination of major errors and biases in the observations is required for a reliable ambiguity resolution. A chosen procedure for the elimination can affect the time and reliability of the ambiguity resolution. The update rate of the measurements is also important. For example, if an automobile is tracked, one should have a two- to five-second update rate; an aircraft requires at least a one-second update rate [Ashtech, 1993]. It is possible to use the single frequency observables for OTF, but it may take 50 minutes to resolve the ambiguities [Deloach, et al., 1995]. All the above requirements or limitations, in fact, often make the OTF solution impossible.

From the above, we can see that many existing navigation services, especially those of classical kinds, are not able to meet the more stringent requirements for precise position and velocity determination. Even with most advanced positioning equipment, a centimetre level accuracy can not always be achieved. It seems that the requirement of tractor implement guidance for drilling, hoeing and plowing in agriculture can not be met by any navigation service. To get a solution in which the random error effect has been reduced, one still needs navigation filter.

In many circumstances, the detection of blunders or the test of the reliability of the coming data, are required. For example, as we have pointed out, the chosen procedure

for the elimination of the major errors and biases in observations can affect the reliability of the ambiguity resolution. That means even with most advanced equipment there is a need to test the reliability of the solutions. No doubt, a navigation filter with blunder detection function is still useful. We think, there will be more activities in future which will require a more precise position and velocity determination.

1.2 existing navigation filters and their limitations

Navigation problems are commonly solved by combining two different kinds of information; observations of the motion of the vehicle, and a model of the associated "process" derived from some basic physical laws and represented by a differential equation. Here, the associated "process" may be either the motion of the vehicle, or the error of the determinated motion, described by a state vector [Minkler and Minkler, 1993]. The existing navigation filters can be classified into two kinds, according to whether their models are of random type or not. One is represented by the Kalman filter in which the "process" is modeled as a random process. The other is represented by a polynomial filter in which the "process" model is a deterministic model.

Kalman filter is the method most often used nowadays. It is based on some assumptions (see *Chapter* 2) and if all the assumptions are met it can offer optimal estimation and prediction. It has been very successfully applied in the space industry. But in marine navigation, the application has not been so successful. The difficulty of its application in marine navigation is related to its "process" model given a priori by the navigator. Realistically, in many environments, the a priori "process" model does not represent reality, especially in a marine environment. Dove and Miller [1989] wrote: " The lack of recently published papers describing the use of Kalman filters in marine navigation does suggest that perhaps there has not been hoped for progress in this area.".

In a polynomial filter (see *Chapter* 2), the associated "process" is modeled by a polynomial. The purpose of the filter is to fit the polynomial based on a sequence of observations made on the vehicle. The navigator has to select the degree of the polynomial in advance. Then biases or systematic errors will arise when the true "process" is not adequately approximated by the polynomial model, but the random errors will be large when the degree is large [Morrison, 1969].

1.3 a considered alternative and the motivation

The difficulty of the application of the existing filters in marine navigation is usually not related to measurements since precise observations of positions and velocity of the vehicle can be obtained. In light of this, we try to develop an alternative filter which relies solely on information from measurements of the vehicle: position fixes, velocities and their error statistics (covariance matrice) and allows the navigator to intervene. Here, we have assumed that the measurement errors are Gaussian distributed, so the covariance matrice can then fully describe the uncertainties of the measurements. A cluster of observed position fixes contains kinematic information about the trajectory of the vehicle. We wish to extract this kinematic information and use it in the navigation.

Real force fields, which affect the trajectories of real particles, are routinely represented as gradients of corresponding potential fields. We base our filter on an analogy where we consider position fixes as "force sources" tending to "attract" the undetermined trajectory to pass through these fixes. When the vehicle is traveling, it is assumed that instantaneous position fixes, and the corresponding error statistics, are obtained from one positioning system or another. Based on these position fixes and their error statistics, a virtual potential field is constructed in which an imaginary mass particle is forced to move. The potential field caused by a single position fix (we call it "position potential" field) should reflect the accuracy of the position fix.

The more accurate the position, the stronger it should attract the particle. Also the force generated by the position fix should monotonically decrease with the closeness of the particle to the source (the position fix). It is obvious that we are talking about a non-Newtonian potential field.

The impetus for our research was in Inzinga and Vaníček's [1985] attempt to create a navigation filter based on the same motivation. In this attempt, the chosen position potential function was related to the error function for the Gaussian probability function. In this way, the force produced by the position potential field on the particle was directly and simply linked with the probability associated with the position fix. But the equation of the motion of the particle based on the position potential function was very difficult to treat analytically.

1.4 methodology and scope of the research

To realize the above idea, we must first select a proper potential function for an individual position fix, i.e., the position potential function. Using this function we can establish a real-time position potential field for position fixes. To be flexible and to reflect the changing navigation environment, the potential function is endowed with some parameters the values of which are allowed to change. Then we can set up an equation (of the motion of the imaginary mass particle) which has to be solved to get the equation for the state of the motion. We call this equation of the state of the motion briefly a "motion model".

But without the parameters and some initial conditions, it is not possible to generate a unique trajectory. Accordingly we require some further information to narrow down our trajectory choice and to select the trajectory along which we believe the motion is taking place. We want the filter to determine (learn) the parameters and the initial conditions from the observations. The parameters and initial conditions in the motion model are related to the observations by the so called "observation model".

After the observation model has been constructed, a criterion is set up so that the selection procedure becomes best in some clearly defined sense. The criterion and the observation model basically form a filtering technique, the learning component of the new navigation filter.

This is a Self Learning (the parameters and initial conditions determined by the filter itself) filter that uses Probability space (the error statistics of observed position fixes used) and Non-Newtonian Dynamics (non-Newtonian potential field). For simplicity, we call it the SLPND filter or simply SLPND.

The SLPND is designed to perform functions of an estimator, predictor, smoother and blunder detector while easily accepting navigator's interventions. Here, word 'estimator' means SLPND used to estimate the state of the vehicle at present; word 'predictor' means SLPND used to predict the state at the next time instant; and word 'smoother' means SLPND used to get the smoothed state of the vehicle, for detail, see *Section* 4.3. We shall confine ourselves to two dimensional navigation problems, but this does not mean that the basic idea can not work with three dimensional problems.

1.5 outline of the dissertation

This dissertation is composed of eight chapters and three appendices. The contents of the chapters are as follows.

Chapter 2 briefly describes the existing navigation filters. This chapter lays a basis for comparisons between the existing navigation filters and the SLPND.

Chapter 3 presents in detail the basic ideas behind the SLPND. In this chapter a "position potential function" is selected and a real-time position potential field is built. Then the equation of the motion of a free particle in the field is formulated and solved. At the end of this chapter, the "motion model" is derived.

Chapter 4 introduces the SLPND's computational aspects and functions. The motion model and the observations are combined to form a filter. The least squares

of the observation error is brought in as a criterion for the selection of the unknown parameters. Both nonlinear and linear least squares methods are briefly discussed. All the necessary formulas for computation are derived or summoned. The SLPND as an estimator, predictor, smoother and blunder detector, is then discussed.

Chapter 5 presents the SLPND's navigation results with simulated vehicle tracks. Some SLPND's features are pointed out.

Chapter 6 presents some SLPND's navigation with two real vehicle tracks. Again, some SLPND features are pointed out.

Chapter 7 shows comparisons between the SLPND and the existing navigation filters. First, the comparisons are made conceptually. Then numerical comparisons of navigation results, obtained independently from the SLPND and Kalman filtering technique are made.

Chapter 8 summarizes the research, makes conclusions and recommends some points for further study.

Appendix I contains the procedure for the solution of the equation of the motion of the free particle using Taylor series.

Appendix II describes the coordinate transformation for diagonalization of a covariance matrix.

Appendix III contains some derived mathematics formulae needed in the computations.

1.6 research contributions of this dissertation

The main contributions of this research can be summarized as follows:

1. The creation of a new navigation filter, with a strategy and concepts quite different from any existing data processing methods.

2. The discovery of the analytic solution for the equation of motion of the particle and the derivations of all the necessary mathematical formulae for the new filter.

3. The realization of the new filter in computer software.
4. The evaluation of the new filter using both simulated and real data.
5. Comparisons between the new filter and the existing navigation filters.

Chapter 2

existing navigation filters

In this chapter, we briefly describe two basic types of existing navigation filters; one is the polynomial filter and the other is the Kalman filter. Because nowadays Kalman filter is the most often used navigation filter, we will give it more attention.

2.1 polynomial filter

We assume that a single aspect of the "process" of a vehicle (one of the components of motion, or one of the component of the error vector of the motion if a nominal trajectory is available) can be adequately approximated by the following polynomial,

$$s_k = \sum_{i=0}^{m} \beta_i \phi_i(t_k), \tag{2.1}$$

where,

s is the aspect of the vehicle at the time instant t_k,

$\phi_i(t_k)$ is the i-th term of the selected polynomial that is a function such as, for example, a particular algebraic function, the Legendre function.

β_i is the i-th unknown coefficient of the polynomial.

We also assume that the motion of the vehicle is observed. This gives rise to a sequence of observations and we retain the most recent $j+1$ of them in a "push-down

table", calling them

$$\begin{aligned} l_k, &\quad t_k \\ l_{k-1}, &\quad t_{k-1} \\ &\vdots \\ l_{k-j}, &\quad t_{k-j}. \end{aligned}$$

The purpose of the polynomial filter is to fit the "process" model *eqn.*(2.1) to these $2 \times (j+1)$ numbers in the sense of a well defined criterion, such as the least squares for optimization. An observation model is then established to realize this. The observation model can be written as

$$\mathbf{l} + \mathbf{v} = \mathbf{\Phi}\beta, \tag{2.2}$$

where

$$\mathbf{l} = \begin{pmatrix} l_{k-j} \\ l_{k-j+1} \\ \vdots \\ l_k \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{pmatrix},$$

$\mathbf{v}$ is the error vector associated with $\mathbf{l}$ and is assumed to be Gaussian, and $\mathbf{\Phi}$ is a Vandermode's matrix,

$$\mathbf{\Phi} = \begin{pmatrix} \phi_1(t_{k-j}) & \phi_2(t_{k-j}) & \cdots & \phi_m(t_{k-j}) \\ \phi_1(t_{k-j+1}) & \phi_2(t_{k-j+1}) & \cdots & \phi_m(t_{k-j+1}) \\ \vdots & & & \\ \phi_1(t_k) & \phi_2(t_k) & \cdots & \phi_m(t_k) \end{pmatrix}. \tag{2.3}$$

The least squares solutions for *eqn.*(2.2) can be found in Vaníček and Krakiwsky [1986]. Thereafter the resulting polynomial is taken to be an estimate of the aspect of the "process" which is associated with this sequence of observations. The prediction

of the aspect at the next time instant can be obtained by replacing t_k in *eqn.*(2.1) with t_{k+1}.

If the "process" has more than one independent aspect which are observed, as many polynomial filters as the observed aspects are needed [Morrison, 1969]. All these filters are completely unconnected, and operate entirely independently.

The major advantage of the polynomial filter is that it does not require much knowledge about the true "process". If the polynomial is of an adequate degree, it will automatically seek out the signal in the presence of the observation errors and give us its reasonable estimate. The major disadvantage of the polynomial filter is that users have to select the degree of the polynomial in advance [Morrison, 1969]. In fact, in many navigation environments we have no idea about the proper degree of the polynomial. If the true "process" is not adequately approximated by the polynomial model, the biases or systematic errors would arise, while increasing the degree of the polynomial the variances of quantifying the random errors would increase [Morrison, 1969].

2.2 Kalman filter

The standard Kalman filter can be described as follows. Consider a discrete-time state-space system described by the linear state and output equations [Minkler, 1993]

$$\mathbf{S}_k = \mathbf{\Phi}_{k-1}\mathbf{S}_{k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1} \tag{2.4}$$

$$\mathbf{l}_k = \mathbf{H}_k\mathbf{S}_k + \mathbf{v}_k \qquad k = 1, 2, \ldots, \tag{2.5}$$

where

$\mathbf{S}_k$ is an n-dimensional (system) state vector at the k-th time instant,

$\mathbf{w}_{k-1}$ is a discrete-time p-dimensional random noise from the system "process",

$\mathbf{l}_k$ is an m-dimensional vector representing the measurement of the system at k-th time instant,

$\mathbf{v}_k$ is a discrete-time m-dimensional random noise from the system measurement,

$\mathbf{\Phi}_{k-1}$ is an $n \times n$ nonsingular state transition matrix at the $(k-1)$-st time instant,

$\mathbf{G}_{k-1}$ is an $n \times p$ system associated matrix at the $(k-1)$-st time instant,

$\mathbf{H}_k$ is an $m \times n$ observation matrix at the k-th time instant.

Further, suppose that the system "process" and measurement noise and the system's initial state satisfy the following three assumptions:

1. The "process" and measurement noise random processes, $\{\mathbf{w}_i\}$ and $\{\mathbf{v}_i\}$, are uncorrelated zero-mean white-noise random processes with known auto-covariance functions and $E(\mathbf{v}_i \mathbf{v}_i{}^T)$ is nonsingular for all $i \geq 1$. Then, the auto and cross covariance functions of $\{\mathbf{w}_i\}$ and $\{\mathbf{v}_i\}$ can be expressed as

$$E(\mathbf{w}_i \mathbf{w}_j^T) = \begin{cases} \mathbf{Q}_i & i = j \\ \mathbf{0} & otherwise \end{cases} \tag{2.6}$$

$$E(\mathbf{v}_i \mathbf{v}_j^T) = \begin{cases} \mathbf{R}_i & i = j \\ \mathbf{0} & otherwise \end{cases} \tag{2.7}$$

$$E(\mathbf{w}_i \mathbf{v}_j^T) = \mathbf{0} \qquad for\ all\ i,\ j, \tag{2.8}$$

where $\mathbf{Q}_i$ is known symmetric positive semi-definite matrices, and $\mathbf{R}_i$ is nonsingular.

2. The initial system state, $\mathbf{S}_0$, is a random vector that is uncorrelated with the "process" or the measurement.

3. The initial system state, $\mathbf{S}_0$, has a known mean, $E(\mathbf{S}_0)$, and a known covariance matrix, $\mathbf{C}_0$.

Then, in the k-th iteration, the time update gives an optimal linear minimum variance prediction of the state vector, $\hat{\mathbf{S}}_k^-$, based on $\mathbf{l}_1, \ldots, \mathbf{l}_{k-1}$, and its error covariance matrix $\mathbf{C}_k^-$ [Minkler and Minkler, 1993]:

time update:

$$\hat{\mathbf{S}}_k^- = \Phi_{k-1} \mathbf{S}_{k-1} \tag{2.9}$$

$$\mathbf{C}_k^- = \Phi_{k-1} \mathbf{C}_{k-1} \Phi_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T. \tag{2.10}$$

And the measurement update provides an optimal linear minimum variance estimate of the state vector, $\hat{\mathbf{S}}_k$, based on $\mathbf{l}_1, \ldots, \mathbf{l}_k$, and its error covariance matrix $\mathbf{C}_k$ [Minkler and Minkler, 1993]:

measurement update:

$$\hat{\mathbf{S}}_k = \hat{\mathbf{S}}_k^- + \mathbf{K}_k(\mathbf{l}_k - \mathbf{H}_k\hat{\mathbf{S}}_k^-) \tag{2.11}$$

$$\mathbf{C}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{C}_k^- \tag{2.12}$$

$$\mathbf{K}_k = \mathbf{C}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{C}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}, \tag{2.13}$$

where $\mathbf{I}$ is an $n \times n$ identity matrix. So that under the three assumptions, and according to the given system model described by *eqns.*(2.4) (2.5), *eqns*(2.9) – (2.13) together with the initial conditions provide a sequential and recursive algorithm for the determination of an optimal linear minimum variance estimate for the system state, $\mathbf{S}_k$, based on the measurements $\mathbf{l}_1, \ldots, \mathbf{l}_k$. This sequential and recursive algorithm is called the discrete-time Kalman filtering algorithm. The derivations of *eqns.*(2.9) to (2.13) can be found in many books, e.g., Gelb [1974], Teunissen and Salzmann [1988], and Minkler [1993].

There are two more useful alternative forms and many useful extensions of the standard discrete-time Kalman filter. The two alternative forms are the inverse covariance form (or information matrix form) and the square root filtering technique [Minkler, 1993]. The former is particularly well suited to problems where the measurement dimension is large or where there exists no a priori knowledge of the initial system state. The latter can be used to provide improved numerical stability.

As to the extensions, they have been made in many directions. An adaptive Kalman filter extends the standard Kalman filter to deal with problems where some of the state-space system model parameters are unknown [Salzmann, 1988; Minkler, 1993]. A generalized Kalman filter deals with the problems where system models have known additive and deterministic input signals and the problems where the "process" and measurement noise random processes have non-zero means [Minkler, 1993].

A general Kalman filter extends the generalized Kalman filter for the cases where the "process" and measurement noise are correlated at the same time-point and provides the most general form of the discrete-time Kalman filter [Minkler, 1993]. The standard discrete-time Kalman filter has also been extended for system models with colored "process" and measurement noise [Gelb, 1974; Gao et al, 1992; Minkler, 1993]. Furthermore, the standard discrete-time Kalman filter has been extended for nonlinear systems [Gelb, 1974; Salzmann, 1988; Minkler, 1993]. If the nonlinear system is linearized about a nominal trajectory, we get a linearized Kalman filter. If the nonlinear system is linearized about Kalman filter's estimated trajectory, we get the extended Kalman filter.

2.3 navigation systems and Kalman filtering technique

According to Brown and Hwang [1992], navigation problems seem to form a natural setting for Kalman filtering for the following reasons:

1. The dynamics are usually linear or may be linearized with reasonable accuracy.

2. There is often a redundancy of measurement information from a variety of navigational sources.

3. The navigation problem is basically an on-line problem. Measurements must be processed essentially in real time, and thus efficient processing is a necessity.

4. There is frequently a need to squeeze the best possible performance out of the system, and thus the data processing must be optimal or at least nearly so.

Generally, in practical applications, the state vector of a navigation system involves either the whole state variables or error state variables. The whole state variables actually represent the values of position, velocity, attitude, etc. The error state variables represent the errors in the whole state variables, e.g., position error,

velocity error, etc. The Kalman filter with whole state vector or with error state vector has its own advantages and disadvantages [Minkler, 1993].

There are plenty of examples of using Kalman filter algorithm in navigation systems. For an example in an inertial navigation system see Wong [1982], for precise navigation using DGPS see Schwarz et al. [1989], Hwang and Brown [1990], Cannon [1990], Gao et al. [1992], Lu and Lachapelle [1992] and Liu et al. [1992], for blunder detection see Dodd [1992].

To get a reliable, consistent and accurate navigation, integrating two or more of the available navigation services has been the trend in navigation. Applying Kalman filtering algorithm to an integrated navigation system, a single comprehensive state vector must be specified which contains all possible navigation system parameters, together with the transition model interconnecting them, and an appropriate set of associated weights and correlations. One advantage of the Kalman filter over other approaches is that it can be partitioned, and various observation updates from different subsystems applied at different epochs, i.e., the Kalman filter can be asynchronous. A number of integrated navigation systems have been developed, for examples, see Grant [1976], Falkenberg and Smith [1981], Wells and Grant [1981], Wong and Schwarz [1983] and Schwarz and Cannon [1988].

2.4 limitation of Kalman filtering in marine navigation

With a Kalman filter, the state vector is modeled as a random process. Choosing an appropriate random process model is always an important consideration. Unfortunately, in many real environments, it is almost impossible to know the random process correctly and an application of Kalman filter in these environments may be risky.

As we have seen the Kalman theory requires that both system and measurement noises are white with zero means. A successful application of the Kalman filter is strongly dependent on whether the system dynamics is perfectly modeled in the mean sense. In practice, disturbances such as wind and current may have random variations superimposed on changing mean values (trend). This suggests that it is very difficult, if not impossible, to model the system dynamics of a navigation system perfectly because the environment always changes, including the various maneuvers of the vehicle. As far as measurement noise is concerned the system will require knowledge of the random errors, which are assumed Gaussian.

Wells and Grant [1981] pointed out that Kalman filter becomes difficult if not impossible to construct when

(a) unpredictable errors from environmental sources are present (for example, the effect of wind and current on dead-reckoning and the effect of both changing overland phaselag and shifts from groundwave to skywave propagation on Loran-C;

(b) the suite of navigation systems to be integrated varies from cruise to cruise, or even during a single cruise;

(c) the rate at which error estimates must be updated varies among the error parameters (for example cesium clock drift estimates can be safely updated a few times per day; estimates of ocean current biases on dead-reckoning should be updated a few times an hour).

It would seem that although the Kalman filtering algorithm is used extensively in aerospace navigation, where sudden environmental changes do not occur, it is not easily adaptable to the marine navigation where sudden environmental changes occur from time to time.

Chapter 3

basic idea behind the new filter

When a vehicle is traveling, it is assumed that instantaneous position fixes and the corresponding error statistics can be obtained. As we have mentioned in *Chapter* 1, the cluster of the observed position fixes contains information on the vehicle motion. We desire to extract this information and use it. We consider the statistical confidence regions of position fixes as an expression of a "force field" that tends to "attract" the undetermined trajectory to pass through these regions. Based on the position fixes and error statistics, we construct an imaginary potential field where a free particle will be forced to move. The aim is that the movement of the particle should describe the true trajectory of the vehicle as well as possible. In this chapter, first a position potential function is selected then a real-time position potential field is produced and finally the equation of the motion of the free particle is established and resolved.

3.1 position potential field

A proper "position potential function" U should produce such a "position potential field" that its intensity reflects the reliability of the position fix, and the produced force on a particle should monotonically decrease with the closeness of the particle to the position fix. We select the position potential function to have a positive parameter

G which plays a similar role to the gravitation constant (a scale factor); the magnitude of G should be a subject to choice. The field should also reflect anisotropic accuracy of the position fixes: the lower the uncertainty in a certain direction is, the stronger the attraction for the particle in that direction will be. Furthermore, to let the free particle get away from the past position fix, we assume the effect (potential) of the attracting past position fix to dissipate exponentially with time.

Various accuracy measures have been used to characterize uncertainty in navigation. Most often used accuracy measures are based upon the assumptions of normal distribution of random errors. Reasons for its popularity range from its simplicity to it being the building block for the central limit theorem. It seems reasonable that random errors may be equally positive as negative and that these errors should become less frequent as their size increases. It should be noted, however, that other types of distributions describing the stochastic behavior of errors originating from navigation measurements have been used. For example, Kuebler and Sommers [1982] indicated that the accuracy of position fixes obtained from most existing navigational systems (Omega, NAVSTAR, LORAN-C, Decca) is better described by the Weibull probability distribution.

In this dissertation, we assume that the error vector of the observed position is a Gaussian random vector. We will confine ourselves in two dimension navigation from now on, however, the expansion of the filter in three dimension problem is possible. The probability density function for a two dimensional Gaussian random vector is

$$\phi_{\mathbf{r}^0} = \frac{1}{\mathbf{K}} exp[-\frac{1}{2}(\mathbf{r}-\mathbf{r}^0)^T\mathbf{C}^{-1}(\mathbf{r}-\mathbf{r}^0)], \tag{3.1}$$

where

$$\mathbf{K} = 2\pi(det(\mathbf{C}))^{1/2}.$$

In the above, $\mathbf{r}^0$ is the observed position and $\mathbf{C}$ is its covariance matrix. The loci of constant density function values are described by equations of the form

$$(\mathbf{r}-\mathbf{r}^0)^T\mathbf{C}^{-1}(\mathbf{r}-\mathbf{r}^0) = h, \tag{3.2}$$

where h is a constant. This is an ellipse (also called an error ellipse) and h determines the size of the ellipse. This error ellipse completely describes the accuracy of the position fix. We select *eqn.*(3.2) as the basis of the position potential function U:

$$U_i(t) = G(\mathbf{r} - \mathbf{r}_i^0)^T \mathbf{C}_i^{-1}(\mathbf{r} - \mathbf{r}_i^0)e^{-\alpha(t-t_i)}, \quad t \geq t_i, \tag{3.3}$$

where

$$\mathbf{r} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathbf{r}_i^0 = \begin{pmatrix} x_i^0 \\ y_i^0 \end{pmatrix} \tag{3.4}$$

are the position vectors of the free particle and the i-th position fix respectively, $\mathbf{C}_i$ is the covariance matrix of the i-th position fix, reflecting the uncertainty of the position fix the G, as we have mentioned, is the positive parameter which acts as a scale factor, α is a positive dissipating parameter (both G and α are to be determined), t represents time and t_i is the time when the i-th position fix $\mathbf{r}_i^0$ was taken. Here the term $e^{-\alpha(t-t_i)}$ acts as a decaying factor with respect to the age of the position fix.

The real-time position potential field U at t produced by n position fixes is simply the superposition of the n individual position potential fields:

$$\begin{aligned} U(t) &= \sum_{i=1}^{n} U_i(t) \\ &= Ge^{-\alpha t} \sum_{i=1}^{n} (\mathbf{r} - \mathbf{r}_i^0)^T \mathbf{C}_i^{-1}(\mathbf{r} - \mathbf{r}_i^0)e^{\alpha t_i}, \quad t \geq t_n. \end{aligned} \tag{3.5}$$

Now, a Newtonian potential is a potential which is associated with an inverse square law of force, and therefore varies with distance in the same manner as a gravitational potential [McGraw -Hill, 1978]. So the potential field defined by *eqn.*(3.5) is clearly non-Newtonian. Here, we would like to note that in order to make the SLPND keep pace with the progress of the vehicle as soon as a new position fix appears, the real-time position potential field is updated to include the effect of the new fix.

3.2 equation of the motion of the particle

The partial derivative of the real-time position potential function (*eqn.*(3.5)) with respect to $\mathbf{r}$ gives the acceleration of the particle, i.e.,

$$\begin{aligned} \ddot{\mathbf{r}}(t) &= -\frac{\partial U(t)}{\partial \mathbf{r}} \\ &= -e^{-\alpha t} G(\mathbf{A}\mathbf{r} - \mathbf{B}), \quad t \geq t_n, \end{aligned} \tag{3.6}$$

where

$$\mathbf{A} = 2\sum_{i=1}^{n} e^{\alpha t_i} \mathbf{C}_i^{-1}, \tag{3.7}$$

$$\mathbf{B} = 2\sum_{i=1}^{n} e^{\alpha t_i} \mathbf{C}_i^{-1} \mathbf{r}_i^0, \tag{3.8}$$

which are not functions of time. This is the equation of the motion of the particle in the real-time position potential field after the appearance of n position fixes, i.e., in time $t \geq t_n$.

To avoid possible numerical trouble in computation, i.e., underflow with $e^{-\alpha t}$ and overflow with $e^{\alpha t_n}$, when t_n becomes very large, we move the origin of time t to t_n. Now the equation of the motion of the particle can be written as follows:

$$\ddot{\mathbf{r}}(t) = -e^{-\alpha t} G(\mathbf{A}\mathbf{r} - \mathbf{B}), \quad t \geq 0, \tag{3.9}$$

$$\mathbf{A} = 2\sum_{i=1}^{n} e^{\alpha(t_i - t_n)} \mathbf{C}_i^{-1}, \tag{3.10}$$

$$\mathbf{B} = 2\sum_{i=1}^{n} e^{\alpha(t_i - t_n)} \mathbf{C}_i^{-1} \mathbf{r}_i^0. \tag{3.11}$$

Equations (3.9), (3.10) and (3.11) are valid only from the time instant when the $n-th$ position fix appears to the instant when the $(n+1)-st$ position fix is produced. When the new $((n+1)-st)$ position fix is produced, they will be reformulated and the state of the particle at instant $(n+1)-st$ will be the initial condition for the new equation of the motion governing the particle from t_{n+1} to t_{n+2}. Because every position fix appears suddenly, the acceleration of the particle is not continuous,

however, due to the initial condition, the velocity and the trajectory of the particle are smooth.

Before we begin to solve *eqn.*(3.9), let us discuss the parameter α further. If α is small, the particle will be subject to a resultant force to attract it to the group of past position fixes; in an extreme situation where $\alpha = 0$, i.e., with no decaying ($e^{-\alpha t} = 1$), the particle will tend to oscillate. If α is larger, position potential field will be decaying very fast; when α is very large, all the past position fixes will have no influence on the particle, only the most recent position will give the particle a "kick" at the instance when it appears. Then a question may arise: can α be allowed to go to infinity? The answer is no, because this would correspond to a particle with zero inertial mass and produce an unrealistic motion.

3.3 solutions for the equation of the motion

In some special cases, the coordinates of position fixes in the x and y directions are uncorrelated, or can be regarded as uncorrelated; in most cases, however, they are correlated. For the first kind of cases, covariance matrices of the position fixes are diagonal. In this section, a solution for *eqn.*(3.9) is derived first for the case of a diagonal covariance matrix. Although $\mathbf{A}$ in *eqn.*(3.10) and $\mathbf{B}$ in *eqn.*(3.11) are function of all fixes for $t_i = t_1, t_2, \ldots, t_n$, the numerical results from all the simulations or applications we did have revealed that we have to use a large value of α to get good results (We shall show the numerical validation of this in *Chapter* 5.), then all the effect of the past fixes for $t_i = t_1, t_2, \ldots, t_{n-1}$ can be neglected and only the position fix for t_n remains in $\mathbf{A}$ and $\mathbf{B}$. The significance of this is that the solution for the second kind of cases, where the covariance matrices are fully populated, can be achieved by first making a coordinate transformation resulting in a diagonal covariance matrix and then using the solution for the first kind of cases.

3.3.1 a solution for the cases of diagonal covariance matrices

When the covariance matrices of the position fixes are diagonal, and their inverses are denoted by

$$\mathbf{C}_i^{-1} = \begin{pmatrix} p_{x_i} & 0 \\ 0 & p_{y_i} \end{pmatrix}, \tag{3.12}$$

eqn.(3.9) can be split into two uncoupled ordinary differential equations:

$$\ddot{x}(t) = -G(A_x x - B_x)e^{-\alpha t}, \quad t \geq 0 \tag{3.13}$$

and

$$\ddot{y}(t) = -G(A_y y - B_y)e^{-\alpha t}, \quad t \geq 0, \tag{3.14}$$

where

$$\begin{aligned} A_x &= 2\sum_{i=1}^{n} e^{\alpha(t_i - t_n)} p_{x_i}, \\ B_x &= 2\sum_{i=1}^{n} e^{\alpha(t_i - t_n)} p_{x_i} x_i^0, \\ A_y &= 2\sum_{i=1}^{n} e^{\alpha(t_i - t_n)} p_{y_i}, \\ B_y &= 2\sum_{i=1}^{n} e^{\alpha(t_i - t_n)} p_{y_i} y_i^0. \end{aligned}$$

We first derive the solution for *eqn.*(3.13), then we get the solution for *eqn.*(3.14) simply by substituting y for x.

As the first step, let's find the solution for the homogeneous form of *eqn.*(3.13), i.e.,

$$\ddot{x} + GA_x e^{-\alpha t} x = 0. \tag{3.15}$$

Let $x(t) = u(s)$ and $s = \frac{2}{\alpha} e^{-\frac{\alpha t}{2}} \sqrt{GA_x}$, then,

$$\begin{aligned} \dot{x} &= \frac{ds}{dt} u' = -e^{-\frac{\alpha t}{2}} \sqrt{GA_x} u' \\ \ddot{x} &= (\frac{ds}{dt})^2 u'' + \frac{d^2 s}{dt^2} u' = e^{-\alpha t} GA_x u'' + \frac{\alpha}{2} e^{-\frac{\alpha t}{2}} \sqrt{GA_x} u', \end{aligned}$$

where u' and u'' represent du/ds and d^2u/ds^2 respectively, and $eqn.(3.15)$ becomes

$$s^2u'' + su' + s^2u = 0. \tag{3.16}$$

This is one of the special forms of Bessel equation of order 0 [Rektorys, 1969]. Its solution is a cylindrical function (all kinds of Bessel functions and their linear combinations). The general solution reads:

$$u = c_1 J_0(s) + c_2 N_0(s), \tag{3.17}$$

where J_0 and N_0 are Bessel functions of order 0 and of the first and second (Neumann) kinds respectively, c_1 and c_2 are constants that depend on the initial conditions. By substituting back for u and s, we get the solution of $eqn.(3.15)$ as

$$x(t) = c_1 J_0(\frac{2}{\alpha} e^{-\frac{\alpha t}{2}} \sqrt{GA_x}) + c_2 N_0(\frac{2}{\alpha} e^{-\frac{\alpha t}{2}} \sqrt{GA_x}). \tag{3.18}$$

The second step is to find a particular solution for $eqn.(3.13)$. For $\ddot{x} = 0$, we easily find a particular solution: $x = B_x/A_x$. The general solution for the linear ordinary differential equation can be obtained by adding the general solution of its homogeneous form to one of its particular solution. Thus adding $eqn.(3.18)$ to the above particular solution we get the general solution for $eqn.(3.13)$:

$$\boxed{x(t) = B_x/A_x + c_1 J_0(\frac{2}{\alpha} e^{-\frac{\alpha t}{2}} \sqrt{GA_x}) + c_2 N_0(\frac{2}{\alpha} e^{-\frac{\alpha t}{2}} \sqrt{GA_x}).} \tag{3.19}$$

By substituting y for x, we obtain the solution for $eqn.(3.14)$ with little extra effort:

$$\boxed{y(t) = B_y/A_y + c_1' J_0(\frac{2}{\alpha} e^{-\frac{\alpha t}{2}} \sqrt{GA_y}) + c_2' N_0(\frac{2}{\alpha} e^{-\frac{\alpha t}{2}} \sqrt{GA_y}),} \tag{3.20}$$

where c_1' and c_2', like c_1 and c_2, are undetermined constants.

Taking the time derivative of $eqns.(3.19)$ and (3.20) respectively and noting that

$$\begin{aligned} \frac{dJ_0(z)}{dz} &= -J_1(z) \\ \frac{dN_0(z)}{dz} &= -N_1(z), \end{aligned}$$

where, J_1 and N_1 are respectively the first and second (Neumann) kind of Bessel functions of order 1, we get the following solutions for velocities:

$$\dot{x}(t) = \sqrt{GA_x}e^{-\frac{\alpha t}{2}}(c_1 J_1(\frac{2}{\alpha}e^{-\frac{\alpha t}{2}}\sqrt{GA_x}) + c_2 N_1(\frac{2}{\alpha}e^{-\frac{\alpha t}{2}}\sqrt{GA_x})) \tag{3.21}$$

$$\dot{y}(t) = \sqrt{GA_y}e^{-\frac{\alpha t}{2}}(c_1' J_1(\frac{2}{\alpha}e^{-\frac{\alpha t}{2}}\sqrt{GA_y}) + c_2' N_1(\frac{2}{\alpha}e^{-\frac{\alpha t}{2}}\sqrt{GA_y})). \tag{3.22}$$

Equations (3.19), (3.20), (3.21) and (3.22) are the *basic formulae* in our research. With known parameters G and α and known initial conditions, i.e., $(x_n,\ y_n)^T$ and $(\dot{x}_n,\ \dot{y}_n)^T$, the four constants c_1, c_2, c_1' and c_2' can be determined by solving a system of equations (3.19) to (3.22). Noting that

$$J_1(z)N_0(z) - J_0(z)N_1(z) = \frac{2}{\pi z},$$

we have

$$c_1 = -\frac{\pi}{\alpha}[(x_n - B_x/A_x)\sqrt{GA_x}N_1(\frac{2}{\alpha}\sqrt{GA_x}) - \dot{x}_n N_0(\frac{2}{\alpha}\sqrt{GA_x})] \tag{3.23}$$

$$c_2 = \frac{\pi}{\alpha}[(x_n - B_x/A_x)\sqrt{GA_x}J_1(\frac{2}{\alpha}\sqrt{GA_x}) - \dot{x}_n J_0(\frac{2}{\alpha}\sqrt{GA_x})] \tag{3.24}$$

$$c_1' = -\frac{\pi}{\alpha}[(y_n - B_y/A_y)\sqrt{GA_y}N_1(\frac{2}{\alpha}\sqrt{GA_y}) - \dot{y}_n N_0(\frac{2}{\alpha}\sqrt{GA_y})] \tag{3.25}$$

$$c_2' = \frac{\pi}{\alpha}[(y_n - B_y/A_y)\sqrt{GA_y}J_1(\frac{2}{\alpha}\sqrt{GA_y}) - \dot{y}_n J_0(\frac{2}{\alpha}\sqrt{GA_y})]. \tag{3.26}$$

All these four constants are valid only within the interval $[t_n, t_{n+1}]$. Then, the position and velocity at the next time instant t_{n+1} can be determined by *eqns.*(3.19) to (3.22) with $t = t_{n+1} - t_n$.

Now let us discuss the physical units of the quantities involved in the above formulae. First, α should have a unit of s^{-1}. According to *eqn.*(3.3), U must be in m^2s^{-2} (unit for a potential function, see Vaníček and Krakiwsky [1986]) and G should also be in m^2s^{-2}. Then, $\mathbf{A}$ is in m^{-2}, $\mathbf{B}$ in m^{-1}; and the argument of Bessel functions are unitless. According to *eqns.*(3.19) to (3.22), constants c_1, c_2, c_1' and c_2' (here, the word "constants" should be understood as quantities remained unchanged from t_n to t_{n+1}) are in m.

In the following discussion, we only target at the x component of the derived solutions and for simplicity, select a large α to neglect all the effect of the past fixes for $t_i = t_1, t_2, \ldots, t_{n-1}$ in **A** and **B**. Then, *eqns.*(3.19), (3.21), (3.23) and (3.24) become:

$\forall\ 0 \leq t \leq t_{n+1} - t_n$:

$$\begin{aligned} x(t) &= x_n^0 + c_1 J_0(T) + c_2 N_0(T) && (3.27)\\ \dot{x}(t) &= \sqrt{2Gp_{x_n}} e^{-\frac{\alpha t}{2}} (c_1 J_1(T) + c_2 N_1(T)) && (3.28)\\ c_1 &= -\frac{\pi}{\alpha}[\Delta x_n \sqrt{2Gp_{x_n}} N_1(T_0) - \dot{x}_n N_0(T_0)] && (3.29)\\ c_2 &= \frac{\pi}{\alpha}[\Delta x_n \sqrt{2Gp_{x_n}} J_1(T_0) - \dot{x}_n J_0(T_0)], && (3.30) \end{aligned}$$

where

$$\begin{aligned} T(t) &= \frac{2}{\alpha} e^{-\frac{\alpha t}{2}} \sqrt{2Gp_{x_n}} = e^{-\frac{\alpha t}{2}} T_0 && (3.31)\\ T_0 &= \frac{2}{\alpha}\sqrt{2Gp_{x_n}} && (3.32) \end{aligned}$$

and $\Delta x_n = x_n - x_n^0$ is the difference between the (initial) position of the particle and the corresponding observed position at t_n in the x direction. Note that $t = 0$ corresponds to the time instant t_n (We have moved the origin of time to t_n.). By substituting for c_1 and c_2 from *eqns.*(3.29) and (3.30) in *eqns.*(3.27) and (3.28), we get,

$\forall\ 0 \leq t \leq t_{n+1} - t_n$:

$$\begin{aligned} x(t) &= x_n^0 + W_1(\alpha, G, p_{x_n}, t)\Delta x_n + W_2(\alpha, G, p_{x_n}, t)\dot{x}_n && (3.33)\\ \dot{x}(t) &= \dot{x}_n + W_1'(\alpha, G, p_{x_n}, t)\Delta x_n + W_2'(\alpha, G, p_{x_n}, t)\dot{x}_n, && (3.34) \end{aligned}$$

where

$$\begin{aligned} W_1(\alpha, G, p_{x_n}, t) &= \frac{T_0\pi}{2}[J_1(T_0)N_0(T) - N_1(T_0)J_0(T)] && (3.35)\\ W_2(\alpha, G, p_{x_n}, t) &= \frac{\pi}{\alpha}[N_0(T_0)J_0(T) - J_0(T_0)N_0(T)] && (3.36)\\ W_1'(\alpha, G, p_{x_n}, t) &= \frac{\pi\alpha}{4} T T_0 [J_1(T_0)N_1(T) - N_1(T_0)J_1(T)] && (3.37)\\ W_2'(\alpha, G, p_{x_n}, t) &= \frac{T\pi}{2}[N_0(T_0)J_1(T) - J_0(T_0)N_1(T)] - 1. && (3.38) \end{aligned}$$

We may regard W_1, W_2, W_1' and W_2' as special weighting functions. They are the function of parameters α and G, the uncertainty of the observed position at t_n and t (time interval from t_n). When $t = 0$, we have $T = T_0$, $W_1 = 1$ and $W_2 = W_1' = W_2' = 0$, then $x = x_n$ and $\dot{x} = \dot{x}_n$. When $t > 0$, Δx_n and $\dot{x}_n$ are multipied by the weighting functions W_1 and W_2 to give an increment to x_n^0, and multiplied by W_1' and W_2' to give an increment to $\dot{x}_n$.

3.3.2 a solution for the cases of fully populated covariance matrices

When the covariance matrices are fully populated, it is very difficult, if not impossible, to find an analytical solution for *eqn.*(3.9). In this case we would be forced to use Taylor series for a numerical solution, see **Appendix I**. An early version of the SLPND used this approach but it was too slow to be a practical navigation filter. Fortunately, from the simulation we did, we found that the SLPND filter always selects the value of parameter α so large as to make the past fixes to have little influence on the particle (for more details see *Chapter* 5). This sounds physically reasonable since with a small α, the particle would always be attracted by the past position fixes, and its motion will always tend to decelerate. This means we can take only the terms relative to the most recent fix in the coefficient matrices A and B into account, because for larger values $\alpha(t_n - t_i)$, the values $e^{\alpha(t_i - t_n)}$ will be negligiblely small, for instance, for $\alpha = 10$ and $t_n - t_i = 2$ seconds, it equals to 2.0612×10^{-09}. Then *eqns.*(3.10) and (3.11) simplify to:

$$\mathbf{A} = 2\sum_{i=1}^{n} e^{\alpha(t_i - t_n)}\mathbf{C}_i^{-1} = 2e^{\alpha(t_n - t_n)}\mathbf{C}_n^{-1} = 2\mathbf{C}_n^{-1}, \tag{3.39}$$

$$\mathbf{B} = 2\sum_{i=1}^{n} e^{\alpha(t_i - t_n)}\mathbf{C}_i^{-1}\mathbf{r}_i^0 = 2e^{\alpha(t_n - t_n)}\mathbf{C}_n^{-1}\mathbf{r}_n^0 = 2\mathbf{C}_n^{-1}\mathbf{r}_n^0. \tag{3.40}$$

The significance of this simplification is that an analytical solution for *eqn.*(3.9) with fully populated covariance matrices can be always obtained by coordinate transformation. We can first shift the origin of the coordinate system to $\mathbf{r}_n^0$ and then rotate the coordinate system to diagonalize $\mathbf{C}_n$. After this, we can use the formulae derived in the previous section. The procedure for the diagonalization can be found in **Appendix II**.

3.4 motion model

Having derived the solutions for the equation of the motion of the particle (3.9), we can say that $\mathbf{S}_{n+1}$, the state of the particle at t_{n+1}, is a function of its state at t_n (as the initial condition) and the parameters G and α. Let us write

$$\mathbf{S}_{n+1} = \mathbf{f}_n(\mathbf{\Theta}, \mathbf{S}_n), \tag{3.41}$$

where

$$\mathbf{S}_i = \begin{pmatrix} \mathbf{r}_i \\ \dot{\mathbf{r}}_i \end{pmatrix}, \qquad \mathbf{\Theta} = \begin{pmatrix} G \\ \alpha \end{pmatrix}, \tag{3.42}$$

with the understanding that $\mathbf{f}_n$ is a function reflecting the previously observed position fixes and the initial conditions at t_1. This is the motion model of the particle. Provided that the parameters G, α and the initial state (conditions) of the particle are known, the state of the particle can be transformed to the next time instant of interest.

To close this chapter, we would like to make the following two points:

1. The above motion model is updated any time a new position fix appears. In order for the model to be flexible with respect to a changing navigation environment, at least one of the two parameters should not be given a priori. We should let the filter determine its most convenient value. In this way the motion model will be able to learn from the real progress of the vehicle.

2. On the other hand, because the motion model is constructed so that it is based only on the observed position fixes and their covariance matrices, the information obtained from the model has a certain lag behind the real progress of the vehicle. The magnitude of the lag depends on how severe the forced maneuvering is: the sharper the maneuvering, the larger the lag. Now let us consider two situations:

(i) Vehicle velocity at the time of the fix is not known. Then, due to the lag, the particle may initially displays a tendency towards overshooting. As soon as a new position fix is produced, as we have described earlier in this chapter, the particle will be attracted by the new position fix with a force with magnitude directly proportional to the distance between the new position fix and the particle and inversely proportional to the uncertainty of the new position fix.

(ii) Vehicle velocity at the time of the fix is known and communicated to the filter. If velocity observation is available, the lag will be greatly reduced. In fact, when the vehicle is maneuvering under own force, making a sharp turn for example, the filter should be informed of the change, even when the change is known only with a large uncertainty. We will discuss this situation which amounts to the navigator's intervention later.

Chapter 4

computational aspects and functions of the new filter

The derived motion model (*eqn.*(3.41)) can not be used by itself. With the parameters G and α being unknown and the initial state of the motion assumed not explicitly available, there are infinitely many trajectories that satisfy *eqn.*(3.41). It is not known which trajectory represent the best the real one. The SLPND thus requires further information to enable it to narrow down the choices and to produce the best trajectory. We let the SLPND use the past observations and select the trajectory which best fits these observations in the least squares sense, i.e., to "learn" from past experience. In this chapter we form a basic framework for the SLPND to learn: a fixed memory least squares filter. The word filter is intended to convey the idea of the signal (true motion) being allowed to pass, with the noise (the observation errors) being suppressed.

A navigation filter usually has at least two components: one is an estimator and the other is a predictor. An estimator accepts information (observations) in the presence of error and estimates the values of certain parameters. Based on the output of the estimator a predictor puts out predictions of the state vector valid for the time at which the next observation vector becomes available. In addition to the

estimator and predictor, we may also be interested in the smoothed state vector of the trajectory. Smoothing of a time series always implies using not only the past information but also the future information. In this chapter the use of the SLPND as a partial smoother is discussed. The word "partial" means only limited number of the observations are used to do the smoothing (the same data are used to estimate the parameter). In many applications of the navigation filter, blunder detections may be important, we will show that the SLPND can also serve as a blunder detector.

We will also describe the technique used to get the least squares solutions of the observation model. We use nonlinear least squares method to get the estimated parameter vector $\hat{\mathbf{\Theta}}$ and then the estimated and the partially smoothed positions of the particle. Because of the difficulty in the estimation of the associated covariance matrices when using a nonlinear method, after accurate values of the parameter vector $\mathbf{\Theta}$ have been obtained by the nonlinear least squares method, we linearize the observation model with respect to the parameter vector $\mathbf{\Theta}$ and observation vector $\mathbf{l}$ and use the linear least squares method to get the covariance matrices.

4.1 observation model

Assume that the positions and velocities of the vehicle are measured in the presence of additive errors at some discrete times t_i. Observations are read into a computer where they are stored in, what is called "a push-down table". This is a memory storage area in which the most recent observation is entered at the top, while all of the preceding observations are moved down to make room for the newest observation. In this way observations in the table are stored in the proper time sequence. The push-down table is usually of fixed length k, so each observation eventually reaches the bottom of the table. Upon the arrival of new data, the bottom-most entry is then simply discarded or forgotten.

Let the observation and error vectors at time t_i be,

$$\mathbf{l}_i = \begin{pmatrix} \mathbf{l}_{\mathbf{r}_i} \\ \mathbf{l}_{\dot{\mathbf{r}}_i} \end{pmatrix}, \ \mathbf{v}_i = \begin{pmatrix} \mathbf{v}_{\mathbf{r}_i} \\ \mathbf{v}_{\dot{\mathbf{r}}_i} \end{pmatrix}, \quad i = n-k+1, n-k+2, \ldots, n, \tag{4.1}$$

where subsripts $\mathbf{r}_i$ and $\dot{\mathbf{r}}_i$ are used for position and velocity respectively. We assume that $n-th$ position fix is the present fix and there exists the best $\mathbf{\Theta}$ for the trajectory between the $(n-k)-th$ and $n-th$ fixes. We denote the estimate of the best value as $\hat{\mathbf{\Theta}}_n^k$. The subscript n indicates the present fix and the superscript k indicates the number of the most recent position used in the estimation. Using *eqn.*(3.41) we can write:

$$\begin{aligned} \mathbf{l}_i + \mathbf{v}_i &= \mathbf{S}_i \\ &= \mathbf{f}_i(\mathbf{\Theta}, \mathbf{S}_{i-1}) \\ &= \mathbf{f}_i(\mathbf{\Theta}, \mathbf{l}_{i-1} + \mathbf{v}_{i-1}), \quad i = n-k+1, n-k+2, \ldots, n. \end{aligned} \tag{4.2}$$

We now assemble the above equations and obtain:

$$\begin{pmatrix} \mathbf{l}_n \\ \mathbf{l}_{n-1} \\ \vdots \\ \mathbf{l}_{n-k+2} \\ \mathbf{l}_{n-k+1} \end{pmatrix} + \begin{pmatrix} \mathbf{v}_n \\ \mathbf{v}_{n-1} \\ \vdots \\ \mathbf{v}_{n-k+2} \\ \mathbf{v}_{n-k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_n(\mathbf{\Theta}, \mathbf{l}_{n-1} + \mathbf{v}_{n-1}) \\ \mathbf{f}_{n-1}(\mathbf{\Theta}, \mathbf{l}_{n-2} + \mathbf{v}_{n-2}) \\ \vdots \\ \mathbf{f}_{n-k+2}(\mathbf{\Theta}, \mathbf{l}_{n-k+1} + \mathbf{v}_{n-k+1}) \\ \mathbf{f}_{n-k+1}(\mathbf{\Theta}, \mathbf{S}_{n-k}) \end{pmatrix}, \tag{4.3}$$

where $\mathbf{l}_{n-i}$ is the vector of observations at time t_{n-i} and $\mathbf{v}_{n-i}$ is the vector of errors in $\mathbf{l}_{n-i}$. We denote the total observation vector as $\mathbf{l}$ and its error vector as $\mathbf{v}$:

$$\mathbf{l} = \begin{pmatrix} \mathbf{l}_n \\ \mathbf{l}_{n-1} \\ \vdots \\ \mathbf{l}_{n-k+2} \\ \mathbf{l}_{n-k+1} \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} \mathbf{v}_n \\ \mathbf{v}_{n-1} \\ \vdots \\ \mathbf{v}_{n-k+2} \\ \mathbf{v}_{n-k+1} \end{pmatrix}. \tag{4.4}$$

The total observation vector $\mathbf{l}$ will form the input to our filter. We assume that the random variables forming $\mathbf{v}$ have all expected mean equal to 0, and we denote the covariance matrix of $\mathbf{v}$ by $\mathbf{C}$.

Then eqn.(4.3) can be written more neatly as

$$\mathbf{l} + \mathbf{v} = \mathbf{f}(\mathbf{\Theta}, \mathbf{l} + \mathbf{v}), \tag{4.5}$$

where

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}_n \\ \mathbf{f}_{n-1} \\ \vdots \\ \mathbf{f}_{n-k+2} \\ \mathbf{f}_{n-k+1} \end{pmatrix}. \tag{4.6}$$

Equation (4.5) shows that the entire set of observations, made over the time span $[t_{n-k+1}, t_n]$ can be related to the parameter vector $\mathbf{\Theta}$ and the initial state $\mathbf{S}_{n-k}$ of the motion at t_{n-k}. The most important task of this chapter is to estimate the best value of $\mathbf{\Theta}$ within $[t_{n-k}, t_n]$, i.e., $\hat{\mathbf{\Theta}}_n^k$ by using *eqn.*(4.5). It is obvious that $\hat{\mathbf{\Theta}}_n^k$ is function of the observations from t_{n-k} to t_n and the initial condition of the motion at t_{n-k}, i.e.,

$$\hat{\mathbf{\Theta}}_n^k = \Psi(\mathbf{l}, \mathbf{l}_{n-k}, \mathbf{S}_{n-k}). \tag{4.7}$$

Equation (4.5) is called the total observation equation and is used over and over again at any time a new position fix is produced.

Let us discuss the filtering options now. By solving *eqn.*(4.5) we can achieve $\hat{\mathbf{\Theta}}_n^k$. Based on $\hat{\mathbf{\Theta}}_n^k$ and *eqn.*(4.3) we can obtain a trajectory of the particle between t_{n-k} and t_n. The position of the particle at t_n, i.e., $\hat{\mathbf{l}}_n$ is regarded as the estimated present state of the vehicle and the position of the particle at $t_i, n-k < i < n$ can be regarded as the partially smoothed positions of the vehicle. We suppose that $\hat{\mathbf{\Theta}}_n^k$ will also be valid within $[t_n, t_{n+1}]$, then based on the estimated state $\hat{\mathbf{l}}_n$ and the motion model $\mathbf{f}_{n+1}$, which is newly derived on the arrival of the $n-th$ position fix, a prediction of

the state of the vehicle can be made. Based on the prediction, once the new $((n+1)st)$ position fix comes, a blunder detection can be made, for more detail see *Section* 4.3.3.

4.2 determination of unknown parameters

It is time for the parameter estimation. First we should select k. It is hard to determine a generally applicable k, but usually, if the vehicle is always under maneuvering it may be more proper to use a smaller k. In our study once k is selected it will not change during the commission. As we have mentioned, *eqn.*(4.5) is used over and over again at any time a new position fix is produced, so is the following procedure for the estimation of the unknown parameter vector.

4.2.1 criterion for parameter determination

Equation (4.5) is underdetermined and has an infinite number of solutions for $\mathbf{\Theta}$ and $\mathbf{v}$ [Vaníček and Krakiwsky, 1986]. A criterion is needed so that the selection is best in some clearly defined sense. As usual, we use the least squares criterion. This means for $\mathbf{\Theta} = (G\ \alpha)^T$

$$\begin{aligned} &\min \qquad \Phi(\mathbf{\Theta}) \Rightarrow \hat{\mathbf{\Theta}}_n^k, \\ &a_1 \leq G \leq b_1 \\ &a_2 \leq \alpha \leq b_2 \end{aligned} \tag{4.8}$$

where $\Phi(\mathbf{\Theta}) = \mathbf{v}^T\mathbf{C}^{-1}\mathbf{v}$ is called the objective function, and a_1, a_2, b_1, b_2 are all positive; they define the bounds of the parameter vector. Note that we may regard the initial state of the motion $\mathbf{S}_{n-k}$ as pseudo-observations; then we will enlarge the vector $\mathbf{v}$ and matrix $\mathbf{C}$ to include the corresponding pseudo-observations. The outstanding properties of this criterion are that the solution is comparatively easy from the mathematical point of view and lends itself to an easy statistical interpretation. For further detail, the reader is referred to Vaníček and Krakiwsky [1986].

The problem defined by (4.8) can be solved by either iterative linear least squares method or nonlinear least squares method. For the first method, one can find it in Vaníček and Krakiwsky [1986], the second in Pierre [1969], Bard [1974], Reklaitis, et al., [1983], Torn and Zilinskas [1989] and Teunissen [1990]. Both kinds of methods are iterative in nature. The linear method gives an explicit Ψ in *eqn.*(4.7), while the nonlinear method can not.

4.2.2 linear least squares method

If a good initial guess of the parameter vector $\mathbf{\Theta}$ is available, the linear least squares method will be the most efficient method for both the parameter and covariance matrix estimations. In this case *eqn.*(4.5) is linearized with respect to the initial guess $\mathbf{\Theta}^{(0)}$ and becomes

$$\mathbf{D}\delta + \mathbf{H}\mathbf{v} + \mathbf{w} = \mathbf{0}, \tag{4.9}$$

where δ is the correction vector to $\mathbf{\Theta}^{(0)}$ (small for a good initial estimate $\mathbf{\Theta}^{(0)}$), the misclosure vector $\mathbf{w}$ equals to $\mathbf{l} - \mathbf{f}(\mathbf{\Theta}^{(0)}, \mathbf{l})$ and the design matrices $\mathbf{D}$ and $\mathbf{H}$ given by

$$\mathbf{D} = -\left.\begin{pmatrix} \frac{\partial \mathbf{f}_n}{\partial \mathbf{\Theta}} \\ \frac{\partial \mathbf{f}_{n-1}}{\partial \mathbf{\Theta}} \\ \vdots \\ \frac{\partial \mathbf{f}_{n-k}}{\partial \mathbf{\Theta}} \end{pmatrix}\right|_{\mathbf{\Theta}=\mathbf{\Theta}^{(0)}} \qquad \mathbf{H} = \mathbf{I} + \left.\begin{pmatrix} \frac{\partial \mathbf{f}_n}{\partial \mathbf{l}} \\ \frac{\partial \mathbf{f}_{n-1}}{\partial \mathbf{l}} \\ \vdots \\ \frac{\partial \mathbf{f}_{n-k}}{\partial \mathbf{l}} \end{pmatrix}\right|_{\mathbf{\Theta}=\mathbf{\Theta}^{(0)}}, \tag{4.10}$$

where $\mathbf{I}$ is an identity matrix. From *eqn.*(4.2), we have

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{\Theta}} = \frac{\partial \mathbf{f}_i(\mathbf{\Theta}, \mathbf{S_{i-1}})}{\partial \mathbf{\Theta}} + \frac{\partial \mathbf{f}_i(\mathbf{\Theta}, \mathbf{S_{i-1}})}{\partial \mathbf{S}_{i-1}} \frac{\partial \mathbf{S}_{i-1}}{\partial \mathbf{\Theta}}. \tag{4.11}$$

All the partial derivatives needed by the above formula have been derived and can be found in **Appendix III**.

According to Vaníček and Krakiwsky [1986], the least squares solution of *eqn.*(4.9) is as follows,

$$\begin{aligned}
\hat{\mathbf{\Theta}}_n^k &= \mathbf{\Theta}^{(0)} + \hat{\delta} \\
&= \mathbf{\Theta}^{(0)} - \mathbf{N}^{-1}\mathbf{D}^T\mathbf{M}\mathbf{w}, && (4.12)\\
\mathbf{C}_{\hat{\mathbf{\Theta}}} &= \mathbf{N}^{-1} && (4.13)\\
\mathbf{N} &= \mathbf{D}^T\mathbf{M}\mathbf{D} && (4.14)\\
\mathbf{M} &= (\mathbf{H}\mathbf{C}\mathbf{H}^T)^{-1}. && (4.15)
\end{aligned}$$

The other equations of interest are:

$$\begin{aligned}
\hat{\mathbf{l}} &= \mathbf{l} - \mathbf{C}\mathbf{H}^T\mathbf{M}(\mathbf{D}\hat{\delta} + \mathbf{w}) && (4.16)\\
\mathbf{C}_{\hat{\mathbf{l}}} &= \mathbf{C} - \mathbf{C}\mathbf{H}^T\mathbf{M}(\mathbf{I} - \mathbf{D}\mathbf{N}^{-1}\mathbf{D}^T\mathbf{M})\mathbf{H}\mathbf{C} && (4.17)\\
\mathbf{C}_{\hat{\mathbf{\Theta}}_n^k\hat{\mathbf{l}}} &= -\mathbf{N}^{-1}\mathbf{D}^T\mathbf{M}\mathbf{H}\mathbf{C}, && (4.18)
\end{aligned}$$

where $\hat{\mathbf{\Theta}}_n^k$ is the least-squares estimate for $\mathbf{\Theta}$ in $[t_{n-k}, t_n]$, $\mathbf{C}_{\hat{\mathbf{\Theta}}_n^k}$ is an approximation of the covariance matrix of $\hat{\mathbf{\Theta}}_n^k$, $\hat{\mathbf{l}}$ is the least-squares estimate of the states of the motion, $\mathbf{C}_{\hat{\mathbf{l}}}$ is the least-squares estimate of the covariance matrix of $\hat{\mathbf{l}}$ and $\mathbf{C}_{\hat{\mathbf{\Theta}}_n^k\hat{\mathbf{l}}}$ is the estimated cross covariance matrix between $\hat{\mathbf{\Theta}}_n^k$ and $\hat{\mathbf{l}}$. The $\hat{\mathbf{\Theta}}_n^k$ with $\mathbf{C}_{\hat{\mathbf{\Theta}}_n^k}$, $\hat{\mathbf{S}}_n = \hat{\mathbf{l}}_n$ with $\mathbf{C}_{\hat{\mathbf{l}}_n}$ and $\mathbf{C}_{\hat{\mathbf{\Theta}}_n^k\hat{\mathbf{l}}_n}$ are used to predict the next state of the motion and its covariance matrix.

If the scale (variance factor) of the covariance matrix of the observations, σ_0^2 is not known, i.e., if only the relative size of the elements of $\mathbf{C}$ is known, we need to estimate this variance factor and then multiply *eqns.*(4.13),(4.17) and (4.18) by the estimated variance factor $\hat{\sigma}_0^2$ to get $\hat{\mathbf{C}}_{\hat{\mathbf{\Theta}}_n^k}$, $\hat{\mathbf{C}}_{\hat{\mathbf{l}}}$ and $\hat{\mathbf{C}}_{\hat{\mathbf{\Theta}}_n^k\hat{\mathbf{l}}}$, for more details, see Vaníček and Krakiwsky [1986].

We once tried to use iterative linear least squares method to solve *eqn.*(4.5). Unfortunately at a certain iteration, parameter G might become negative, and this made computer stop processing because of numerical error with $\sqrt{G}$. The author

made no effort to get rid of this trouble, instead, used nonlinear optimization method to continue the research.

4.2.3 a brief review of unconstrained nonlinear optimization methods

Let us now talk about unconstrained nonlinear optimization methods. The word "unconstrained" means that we do not pay attention to the inequality in eqn.(4.8). We start with a given initial guess $\mathbf{\Theta}_1$, and proceed to generate a sequence of points $\mathbf{\Theta}_2, \mathbf{\Theta}_3, \ldots$ which we hope converge to the point $\hat{\mathbf{\Theta}}_n^k$ at which $\Phi(\hat{\mathbf{\Theta}}_n^k)$ is the minimum. $\mathbf{\Theta}_i$ is the i-th iterate obtained by $(i-1)st$ iteration (computation). The vector

$$\sigma_i = \mathbf{\Theta}_{i+1} - \mathbf{\Theta}_i \tag{4.19}$$

is called the i-th step. We wish that each step brings us closer to the minimum. We consider the i-th step to have "improved" the situation (by bringing us closer to the minimum), if

$$\Phi_{i+1} < \Phi_i, \tag{4.20}$$

where

$$\Phi_j = \Phi(\mathbf{\Theta}_j) \qquad (j = 1, 2, \ldots).$$

We call the i-th step acceptable if $eqn.(4.20)$ holds. An iterative method is acceptable if all the steps it produces are acceptable.

Most nonlinear methods adhere to the following scheme:

1. Determine a vector $\mathbf{d}_i$ in the direction of the proposed i-th step.
2. Determine a scalar ρ_i such that the step

$$\sigma_i = \rho_i \mathbf{d}_i \tag{4.21}$$

is acceptable. That is, we take

$$\boldsymbol{\Theta}_{i+1} = \boldsymbol{\Theta}_i + \rho_i \mathbf{d}_i \tag{4.22}$$

and require that ρ_i be chosen so that $eqn.(4.20)$ holds.

3. Test whether the termination criterion is met. If not, increase i by one and return to step 1. If yes, accept $\boldsymbol{\Theta}_{i+1}$ as the value of $\hat{\boldsymbol{\Theta}}_n^k$.

The manner in which $\mathbf{d}_i$ and ρ_i are determined at each iteration specifies a particular method. Usually ρ_i is selected so as to minimize $\Phi(\boldsymbol{\Theta})$ in the $\mathbf{d}_i$ direction. So the nonlinear methods adhered to the above scheme have two features: iterative and descent.

The iterative techniques fall roughly into two classes: direct search methods and gradient methods. Direct search methods are those which do not require the explicit evaluation of any partial derivatives of the objective function $\Phi(\boldsymbol{\Theta})$. Instead they rely solely on values of the objective function, plus the information obtained from the earlier iterations. Gradient methods on the other hand, are those which select the direction vector $\mathbf{d}_i$ using values of partial derivatives of the objective function with respect to the parameters, as well as values of the objective function itself, together with the information obtained from the earlier iterations. The required derivatives, which for some methods are of order higher than the first, can be obtained either analytically or numerically using some finite difference scheme.

We will use an iterative descent gradient method for our solutions. Their iterative formula has a general form [Teunissen, 1990]

$$\boldsymbol{\Theta}_{i+1} = \boldsymbol{\Theta}_i - \rho_i \mathbf{Q}(\boldsymbol{\Theta}_i) \nabla_{\boldsymbol{\Theta}} \Phi(\boldsymbol{\Theta}_i), \tag{4.23}$$

where $\mathbf{Q}(\boldsymbol{\Theta}_i)$ is a positive-definite matrix and

$$\nabla_{\boldsymbol{\Theta}} = \frac{\partial}{\partial \boldsymbol{\Theta}}.$$

The convergence of the descent methods is guaranteed if

$$\forall i: \quad \| \mathbf{I} - \rho_i \mathbf{Q}(\boldsymbol{\Theta_i}) \nabla^{\mathbf{2}}_{\boldsymbol{\Theta}} \boldsymbol{\Phi}(\boldsymbol{\Theta_i}) \| < \mathbf{1} \tag{4.24}$$

where

$$\nabla^2_{\mathbf{\Theta}} = \frac{\partial^2}{\partial^2 \mathbf{\Theta}}$$

and if the initial guess is "sufficiently close" to the solution $\hat{\mathbf{\Theta}}$ [Teunissen, 1990].

Different choices for ρ_i and $\mathbf{Q}(\mathbf{\Theta}_i)$ correspond to different descent algorithms. Cauchy's method is also called the method of steepest descent. The major advantage of this method is its surefootedness [Reklaitis, et al., 1983]. The method typically produces good reduction in the objective function from points far from the minimum, using only the first-order partial derivatives of the objective function with respect to the parameters. But it becomes less efficient as minimum is approached, so the convergence rate of this method is very slow.

Newton's method has a quadratic rate of convergence [Teunissen, 1990]. But Newton's step will often be large when the initial guess is far from the minimum, and there is the real possibility of divergence [Reklaitis, et al., 1983]. So Newton's method can be unreliable for nonquadratic functions. Its other major disadvantage is that it requires the calculation of the second order derivatives of the objective function with respect to the parameters.

Marquardt's method is designed to overcome the first disadvantage of Newton's method. The major advantages of this method are its simplicity, the excellent convergence rate near the minimum and the absence of a line search used to determine ρ_i. It has proven very reliable in practice [Bard, 1974].

Gauss-Newton method takes the advantage of the "sum of squares" structure of the objective function and can be interpreted as a 'regularized' version of the Newton method [Teunissen, 1990]. Within this method the linear least squares approach (cf. *Section* 4.2.2) is repeatly used.

Conjugate Gradient methods tend to exhibit the positive characteristics of the Cauchy and Newton methods, requiring only first-order derivatives. That is, the methods are reliable far away from the minimum and accelerate as the sequence of

iterates approaches the minimum [Reklaitis, et al., 1983].

Variable metric methods are designed to more directly mimic the positive characteristics of Newton's method requiring only first-order derivatives. The specific scheme proposed by Davidon for computing the second derivatives of the objective function has been modified slightly by Fletcher and Powell, and has been widely used in this kind of methods, gaining a reputation of being the most efficient general unconstrained optimization method available [Bard, 1974]. The major disadvantage associated with methods in this class is the need to store a metric matrix [Reklaitis, et al., 1983].

4.2.4 optimization constrained by inequality

For constrained nonlinear problems, there is a family of methods which are essentially adaptations of the unconstrained methods mentioned above [Bard, 1974; Reklaitis, et al., 1983]. With this family of methods, an inequality constraint can be realized in many ways, and we have, for example, the penalty function method, projection method and the method of transformation of variables. In our research, we use the method of transformation of variables. For the inequality constraint in *eqn.*(4.8), we set

$$\mathbf{\Theta} = \begin{pmatrix} G \\ \alpha \end{pmatrix} = \begin{pmatrix} (b_1 - a_1)\sin^2 \Lambda_1 + a_1 \\ (b_2 - a_2)\sin^2 \Lambda_2 + a_2 \end{pmatrix} \tag{4.25}$$

and minimize $\Phi(\mathbf{\Theta})$ with respect to $\mathbf{\Lambda} = (\Lambda_1\ \Lambda_2)^T$; with $\mathbf{\Lambda}$ varying from $-\infty$ to ∞, $\mathbf{\Theta}$ remains within the prescribed bounds.

4.2.5 global optimization

Sometimes the objective function (*eqn.*(4.8)) has more than one minimum within the specified bounds. Figure 4.1 shows an example we got in one of the simulations, in the Figure "pvv" represents the value of the objective function. Then the derived

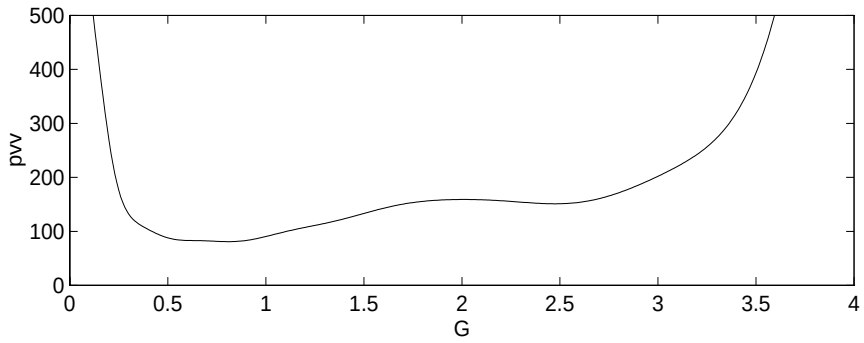

Figure 4.1: Value of objective function versus G

minimum may depend on the location of the initial guess of the parameter vector. If we are not certain the minimum we get is the global minimum, we need to try more than one initial guess. In our research a widely used "uniform covering" method [Torn and Zilinskas, 1989] is used for the distribution of the initial guesses: initial guesses of the parameters are uniformly distributed within a rectangular grid bounded by the inequality in *eqn.*(4.8). This is time-consuming. Fortunately, according to our experience, it is necessary to use multiple guesses only at the beginning of the processing. Usually, from the previous processing, we can get the feel of the bounds within which only one local minimum exists and it happens to be the global minimum of *eqn.*(4.8).

Also, we can use $\hat{\mathbf{\Theta}}_n^k$ as the initial guess for the next estimation of the parameter unless forced maneuvering occurs. To explain this further, we plot the estimated parameter G with error ($\pm 1\ \sigma$) versus time t in two of our simulations in *Figs.*(4.2) and (4.3). The first one is for a vehicle moving in a straight line with constant velocity and the second one for a vehicle always under maneuvering. Both position and velocity observations were used in these two simulations, the velocity of the vehicle in both x and y directions (Vx and Vy) are also plotted for the latter. Considering the error in the estimation of G, we can see that for the first case where no maneuvering happens, the estimated G may be regarded as unchanged; while for the second case where maneuvering occurs from time to time, the estimated G is

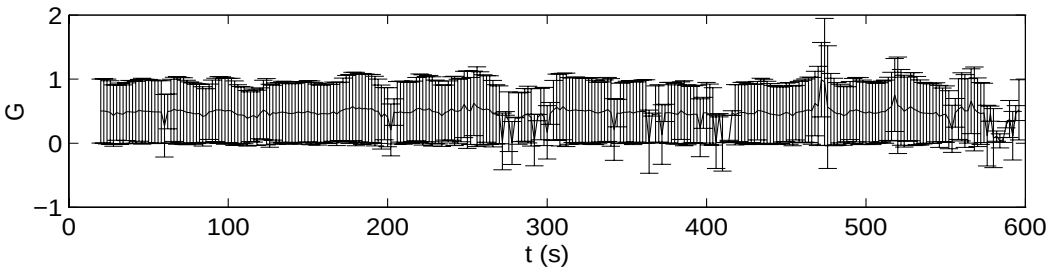

Figure 4.2: Estimated G

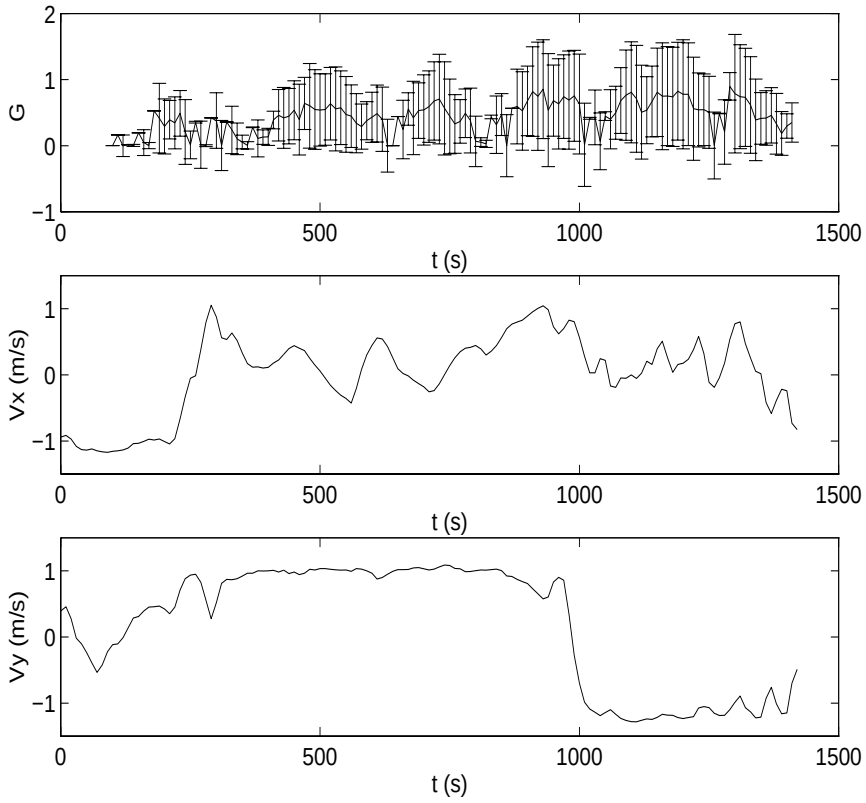

Figure 4.3: Estimated G, and true velocity of the vehicle

somewhat relative to the velocity changes of the vehicle.

4.2.6 evaluation of objective function

As long as the initial state $\mathbf{S}_{n-k}$ of the particle (vehicle) at t_{n-k} is known and the values of parameter $\mathbf{\Theta}$ are given, a trajectory of the particle between the time instants t_{n-k} and t_n can be obtained by using the model (*eqn.*(3.41)). (We have noted in *Chapter* 3 that the model must be updated as soon as a new position fix is produced. That means we need to update the motion model $k+1$ times during the period $[t_{n-k}, t_n]$. The state of the motion at time instant $t_i, n-k \leq i \leq n$ is treated as the initial state for the motion proceeding to time instant t_{i+1}.) The evaluation of the objective function in *eqn.*(4.8) can be done by means of *eqn.*(4.3):

$$\Phi(\mathbf{\Theta}) = \mathbf{v}^T \mathbf{C}^{-1} \mathbf{v} = (\mathbf{f} - \mathbf{l})^T \mathbf{C}^{-1} (\mathbf{f} - \mathbf{l}), \tag{4.26}$$

where each $\mathbf{f}_i$ is a function of the initial state $\mathbf{S}_{n-k}$ of the particle at t_{n-k} and the observations between the time instants t_{n-k} and t_{i-1}.

4.2.7 evaluation of gradients

The gradient of the objective function in the artificial coordinate system Λ_1, Λ_2 can be expressed as

$$\frac{\partial \Phi}{\partial \mathbf{\Lambda}} = 2\mathbf{v}^T \mathbf{C}^{-1} \frac{\partial \mathbf{v}}{\partial \mathbf{\Theta}} \frac{\partial \mathbf{\Theta}}{\partial \mathbf{\Lambda}}. \tag{4.27}$$

According to *eqn.*(4.25),

$$\frac{\partial \mathbf{\Theta}}{\partial \mathbf{\Lambda}} = \begin{pmatrix} (b_1 - a_1)\sin(2\Lambda_1) & 0 \\ 0 & (b_2 - a_2)\sin(2\Lambda_2) \end{pmatrix}. \tag{4.28}$$

Note that for $n-k < i \leq n$, according to *eqn.*(4.2), $\mathbf{v}_i = \mathbf{S}_i - \mathbf{l}_i = \mathbf{f}_i(\mathbf{\Theta}, \mathbf{S}_{i-1}) - \mathbf{l}_i$, we have

$$\frac{\partial \mathbf{v}_i}{\partial \mathbf{\Theta}} = \frac{\partial \mathbf{S}_i}{\partial \mathbf{\Theta}} = \frac{\partial \mathbf{f}_i}{\partial \mathbf{\Theta}}, \tag{4.29}$$

$$\frac{\partial \mathbf{v}}{\partial \mathbf{\Theta}} = \frac{\partial \mathbf{f}}{\partial \mathbf{\Theta}} = -\mathbf{D}, \tag{4.30}$$

where $\mathbf{D}$ is nothing else but one of the design matrices in *eqn.*(4.9). Then *eqn.*(4.27) can be written as:

$$\frac{\partial \Phi}{\partial \mathbf{\Lambda}} = -2\mathbf{v}^T\mathbf{C}^{-1}\mathbf{D}\frac{\partial \mathbf{\Theta}}{\partial \mathbf{\Lambda}}. \tag{4.31}$$

4.2.8 determination of unknown parameters

We use either the variable metric method jointly using BFS and DFP formulae (for more details see Reklaitis, et al., [1983]) or the conjugate direction method using FRP formula (for more details see Reklaitis, et al., [1983]) with variable transformation method to implement the optimization and get the best estimate of the set of parameters in the least squares sense. Both methods are most often used nowadays. It has been said that both have the same convergence rate and the former is more stable but needs more memory during the processing. In our problem, it seems that the numerical results from these two methods do not demonstrate any large differences. This in the other hand, indicates the good health of the filter.

4.3 navigation solutions

4.3.1 least squares filter, predictor and smoother

We have described the filtering options in the end of the *Section* 4.1. Now, because $\hat{\mathbf{\Theta}}_n^k$ is obtained by using the least squares approach, by convention we call it the least squares estimate of the parameter vector, accordingly, $\hat{\mathbf{S}}_n = \hat{\mathbf{l}}_n$ the least squares estimate of the present state of the motion, $\mathbf{S}_{n+1} = \mathbf{f}_{n+1}(\hat{\mathbf{\Theta}}_n^k, \hat{\mathbf{S}}_n)$ the least squares predition of the motion, and $\hat{\mathbf{l}}_i$ $(n-k<i<n)$ the least squares partially smoothed state of the motion. By the navigation convention, the procedures to produce them are respectively called the least squares filter, predictor and smoother.

4.3.2 approximation of covariance matrices

Most of the nonlinear optimization methods usually are reliable and efficient in the determination of the unknown parameters, but they can not offer an effective estimation of corresponding covariance matrices. It is possible that some methods can be further modified to produce also error statistics, but this is beyond our scope here. Our way to obtain the corresponding covariance matrices is, after the accurate estimates of the parameters are obtained, to linearize the observation model *eqn.*(4.5) by Taylor series in the neighborhoods of $\mathbf{l}^{(0)} = \mathbf{l}$ and $\mathbf{\Theta}^{(0)} = \hat{\mathbf{\Theta}}_n^k$ as shown in *Section* 4.2.2. Then *eqns.*(4.13), (4.17) and (4.18) are used to obtain covariance matrices $\mathbf{C}_{\hat{\mathbf{\Theta}}_n^k}$, $\mathbf{C}_{\hat{\mathbf{l}}}$ and $\mathbf{C}_{\hat{\mathbf{\Theta}}_n^k\hat{\mathbf{l}}}$.

To obtain the covariance matrix of the predicted state, first we linearize the prediction model $\mathbf{S}_{n+1}$ with respect to $\mathbf{\Theta}$ and $\mathbf{S}_n$ and get:

$$\delta\mathbf{S}_{n+1} = \mathbf{R}_{\mathbf{\Theta}}\delta\mathbf{\Theta} + \mathbf{R}_{\mathbf{S}_n}\delta\mathbf{S}_n, \tag{4.32}$$

where

$$\mathbf{R}_{\mathbf{\Theta}} = \left.\frac{\partial \mathbf{f}_{n+1}(\mathbf{\Theta}, \mathbf{S}_n)}{\partial \mathbf{\Theta}}\right|_{\substack{\mathbf{\Theta} = \hat{\mathbf{\Theta}}_n^k \\ \mathbf{l} = \hat{\mathbf{l}}}}, \quad \mathbf{R}_{\mathbf{S}_n} = \left.\frac{\partial \mathbf{f}_{n+1}(\mathbf{\Theta}, \mathbf{S}_n)}{\partial \mathbf{S}_n}\right|_{\substack{\mathbf{\Theta} = \hat{\mathbf{\Theta}}_n^k \\ \mathbf{l} = \hat{\mathbf{l}}}}. \tag{4.33}$$

Then using the propagation rule for covariance matrices [Vaníček and Krakiwsky, 1986] with the above linearized model, we can obtain the approximation of the covariance matrix of the predicted state:

$$\mathbf{C}_{\mathbf{S}_{n+1}} = \mathbf{R}_{\mathbf{\Theta}}\mathbf{C}_{\hat{\mathbf{\Theta}}_n^k}\mathbf{R}_{\mathbf{\Theta}}^T + \mathbf{R}_{\mathbf{S}_n}\mathbf{C}_{\hat{\mathbf{l}}_n}\mathbf{R}_{\mathbf{S}_n}^T + \mathbf{R}_{\mathbf{\Theta}}\mathbf{C}_{\hat{\mathbf{\Theta}}_n^k\hat{\mathbf{l}}_n}\mathbf{R}_{\mathbf{S}_n}^T + \mathbf{R}_{\mathbf{S}_n}\mathbf{C}_{\hat{\mathbf{l}}_n\hat{\mathbf{\Theta}}_n^k}\mathbf{R}_{\mathbf{\Theta}}^T. \tag{4.34}$$

4.3.3 blunder detection

As soon as a new position observation $\mathbf{r}$ with its covariance matrix $\mathbf{C}_{\mathbf{r}}$ becomes available, the computer figures out whether the newcomer is compatible with the

corresponding prediction $\hat{\mathbf{r}}$. If the newcomer is not compatible with the prediction, it may be discarded. However, the final decision is up to the navigator.

According to Vaníček and Krakiwsky [1986], if the covariance matrices of observations are known, an error ellipse with confidence level $1-\alpha$ is defined by

$$\Delta\hat{\mathbf{r}}^T\mathbf{C}_{\Delta\hat{\mathbf{r}}}^{-1}\Delta\hat{\mathbf{r}} = \xi_{\chi^2_2,1-\alpha}, \tag{4.35}$$

where $\Delta\hat{\mathbf{r}} = \mathbf{r}-\hat{\mathbf{r}}$, $\mathbf{C}_{\Delta\hat{\mathbf{r}}} = \mathbf{C}_{\mathbf{r}}+\mathbf{C}_{\hat{\mathbf{r}}}$ is its covariance matrix, α is a prescribed significance level and should not be confused with the decaying factor we use throughout this dissertation, $\xi_{\chi^2_2,1-\alpha}$ is the value of the abscissa fixed through the selected value of α to make the probability of a χ^2 statistic (with 2 degrees of freedom) within the interval $(\xi_{\chi^2_2,1-\alpha},\infty)$ to be equal to α. If only cofactor matrices (relative size of the covariance matrices) of the observations are known, an error ellipse with confidence level $1-\alpha$ is defined by

$$\frac{\Delta\hat{\mathbf{r}}^T\mathbf{C}_{\Delta\hat{\mathbf{r}}}^{-1}\Delta\hat{\mathbf{r}}}{(\nu\hat{\sigma}_0^2/\sigma_0^2)/\nu} = \Delta\hat{\mathbf{r}}^T\hat{\mathbf{C}}_{\Delta\hat{\mathbf{r}}}^{-1}\Delta\hat{\mathbf{r}} = 2\xi_{F(2,\nu),1-\alpha}, \tag{4.36}$$

where, σ_0^2 is the variance factor and $\hat{\sigma}_0^2$ is its estimate (for more details see Vaníček and Krakiwsky [1986]), ν is the number of degrees of freedom used to calculate $\hat{\sigma}_0^2$, $\hat{\mathbf{C}}_{\Delta\hat{\mathbf{r}}}$ is the estimated covariance matrix of $\Delta\hat{\mathbf{r}}$, $\xi_{F(2,\nu),1-\alpha}$ is the value of the abscissa which is fixed through the selected value of α to make the probability of a Fisher statistic with the degrees of freedom equal to 2 and ν within the interval $(\xi_{F(2,\nu),1-\alpha},\infty)$ equal to α.

Any tested value $\mathbf{r}$ that falls within the error ellipse centred on $\hat{\mathbf{r}}$ must then be considered compatible with $\hat{\mathbf{r}}$ on the level of $1-\alpha$.

Chapter 5

simulations

The main objective of the simulations was to test the performance of the new filter numerically. We will describe the tests in the following order: effect of k and α, effect of initial velocity, effect of sampling rate, effect of observational errors (random and systematic), finally the ability to accept navigator's intervention. Four simulated tracks, straight, circular, sinusoidal and two straight line with a 90 degree turn in the middle were formed by generating "true" position fixes and adding white Gaussian noise or high frequency sinusoidal systematical noise. The path (estimated, predicted and smoothed) of the vehicle was then computed, by using the SLPND filter formulated in previous chapters. Then the computed positions were compared to the "true" position fixes. We used RMS error (root of mean square error) and maximum estimated position error to measure the "goodness" of the computed track. The RMS error is defined as

$$RMS = \sqrt{\frac{\mathbf{e}^T\mathbf{e}}{u}}, \tag{5.1}$$

where $\mathbf{e}$ is the error vector of the estimated positions (i.e., difference vector between the estimated and the true positions), and u is the number of components in the error vector. The estimated position error is defined as the norm of the difference vector between the estimated and the true positions. The maximum estimated position error

is the maximum among all the estimated position errors.

Equation(3.9) to (3.11) involved all the past position fixes. In fact, because of the presence of the decaying factor term $e^{-\alpha t}$ the real-time position potential field (*eqn.*(3.5)) can be simply constructed by accounting only for the most recent position fix if the value of α is sufficiently large enough. In this chapter, we will show that this simplification is justified by demonstrating that the SLPND requires α to be large to get good results and with this value of α the influence of all the past position fixes can be neglected.

The SLPND can estimate both parameters G and α. In the early simulations we used to let the filter do so and obtained good results, but we were not satisfied with the SLPND's processing speed. In order to improve the processing speed, which is very important to a navigation filter, we now always fix α and let filter to determine G. In the following, all the results are obtained in this manner.

5.1 simulated tracks

straight track: In this case, the vehicle is moving straight with a constant velocity. The straight track is defined by

$$y(t) = 0.0\ m, \quad x(t) = 2t\ m \tag{5.2}$$

where t is time in seconds.

circular track: We also consider the vehicle under a slow and constant maneuvering. It is moving circularly with a uniform angular velocity. The circular track is defined by

$$x = r\cos(\frac{\pi t}{2000}) \tag{5.3}$$

$$y = r\sin(\frac{\pi t}{2000}) \tag{5.4}$$

where $r = 1000\ m$ is the radius of the circle.

sinusoidal track: In the third case, we suppose that the vehicle is under nonconstant maneuvering. Its track is a sinusoidal which is defined by

$$x = 1400\sin(\frac{\pi t}{1760} + \frac{\pi}{90}) \tag{5.5}$$

$$y = 250\sin(\frac{\pi t}{280} + \frac{\pi}{4.5}) + \frac{3t}{8} + 250. \tag{5.6}$$

This track is shown in Fig. 5.1.

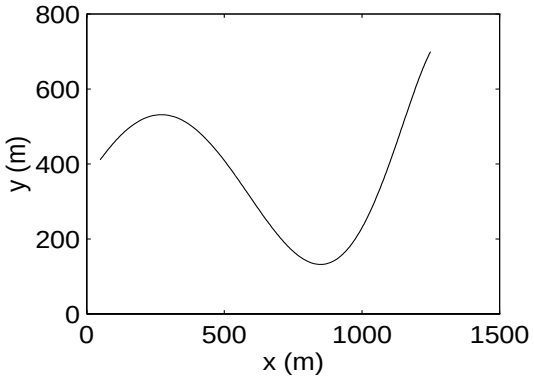

Figure 5.1: sinusoidal simulation track

We assume that only positions of the vehicle are observed with a sampling interval of 2 seconds. Two independent Gaussian noise series (one for the x-component the other for the y-component) are generated (by using function Randn in Matlab), and added to the true track to get the simulated positions. The error set (two series) has zero means and standard deviations $\sigma_x = \sigma_y = 1\ m$ and has a maximum simulated position error of 3.7 m. The initial position ($t = 0$) is supposed to have been obtained by an observation, i.e., it has the same accuracy as the other observed position fixes. Initial velocity $\mathbf{v}_0^T = (x_0, y_0)$ of the motion is obtained in an unspecified way (even by a guess), as

$$\dot{x}_0 = 1.9\ m/s, \quad \dot{y}_0 = 0.1\ m/s. \tag{5.7}$$

The initial velocity is assumed known with errors given by its covariance matrix,

$$\mathbf{C}_{v_0} = \begin{pmatrix} 0.01 & 0.0 \\ 0.0 & 0.01 \end{pmatrix}\ m^2/s^2. \tag{5.8}$$

Unless specified, we will use these assumptions throughout this chapter.

5.2 effect of k and α

We let α and k be fixed in each simulation. The search area for the unknown parameters G is, according to our experience, defined by $0 \leq G \leq 0.1$ for the first track, $0 \leq G \leq 1$ for the second and third tracks. Many simulations are done with different values of k and α. Each simulation produces a value of RMS and a value of the maximum estimated position error. The relations of RMS and maximum estimated position error (max-e) on α (denoted as 'aph' in the figure) and some values of k are depicted in Fig. 5.2 for the first track (straight), in Fig. 5.3 for the second track (circular) and in Fig. 5.4 for the third track (sinusoidal).

Recall that the standard deviation of the errors in the simulated observations is 1.0 m and the maximum simulated position error is 3.7 m. In Figs. 5.2 and 5.3, within a certain interval of α the RMS and the maximum estimated position error are lower than the standard deviation of the error and the maximum simulated position error respectively. But to the sinusoidal track, only with $k = 5$, we can get a slightly better result than the observed positions (see Fig. 5.4). It is because the vehicle is nonconstantly maneuvering and the SLPND is not informed of the kinematical changes of the vehicle (only the position data are input to the SLPND), the SLPND failed to give better results.

Let us discuss the effect of k now. For the straight track, it seems that with larger k the SLPND gives better results. For the circular track, after $k = 10$ the larger k improves the result little. As to the sinusoidal track, among the three values of k ($k = 2,\ 5,\ 10$) only $k = 5$ leads to slightly better results than the 'observed' positions; after $k = 5$ the larger k degrades the SLPND's output. So the best value of k depends on wheter the vehicle is under maneuvering and how severe the maneuvering is: the shaper the maneuvering the smaller the best k is.

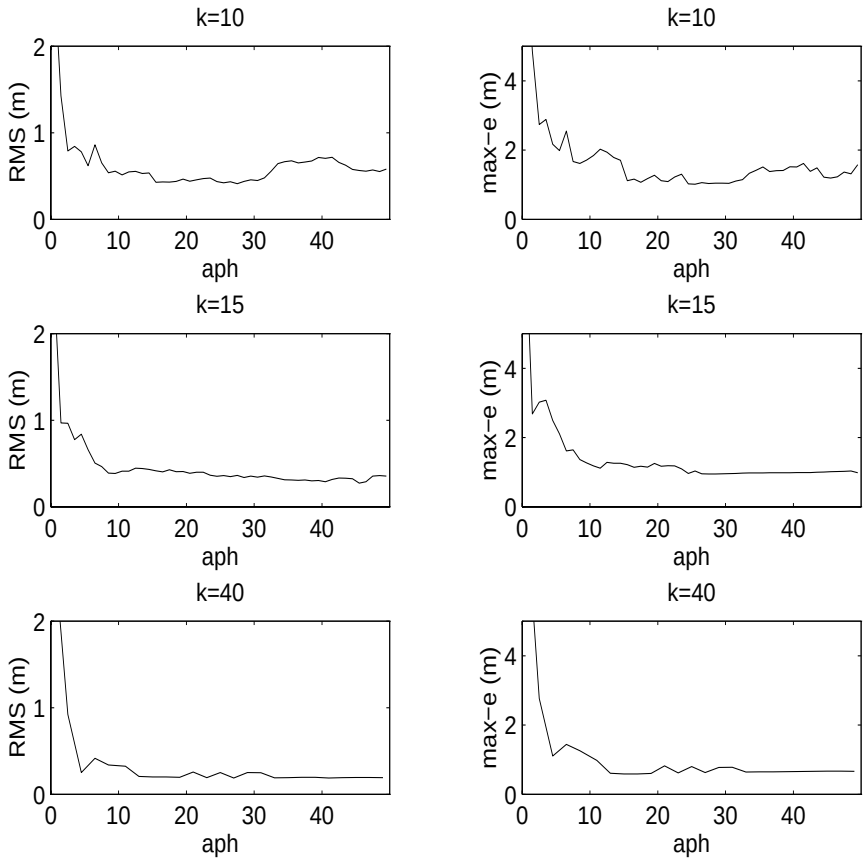

Figure 5.2: RMS and maximum estimated position error, first track

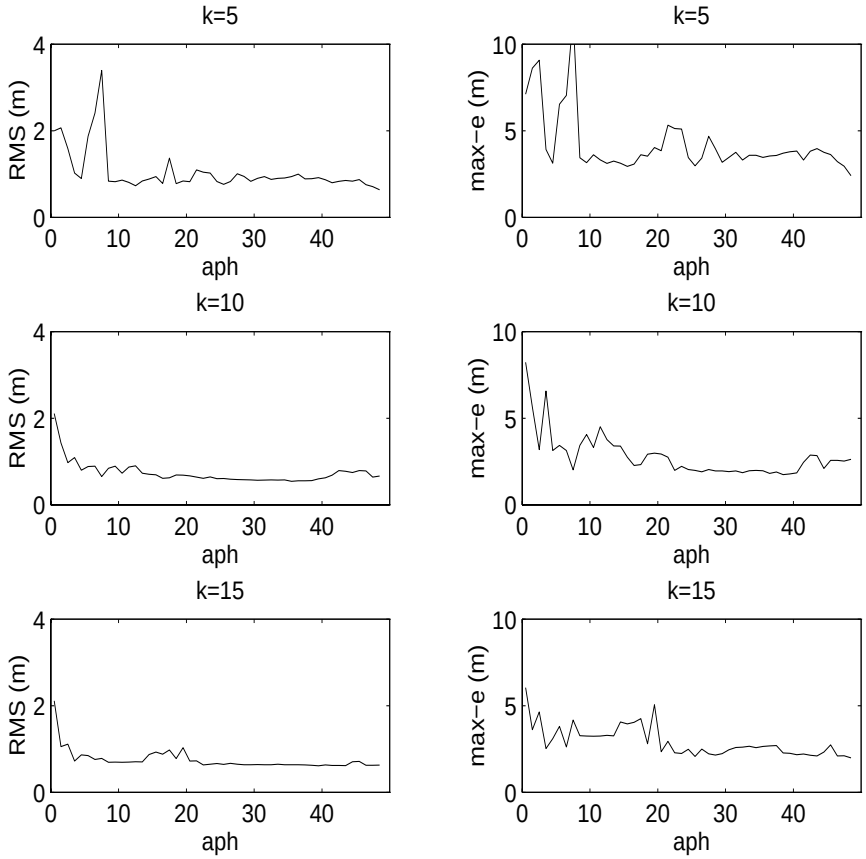

Figure 5.3: RMS and maximum estimated position error, second track

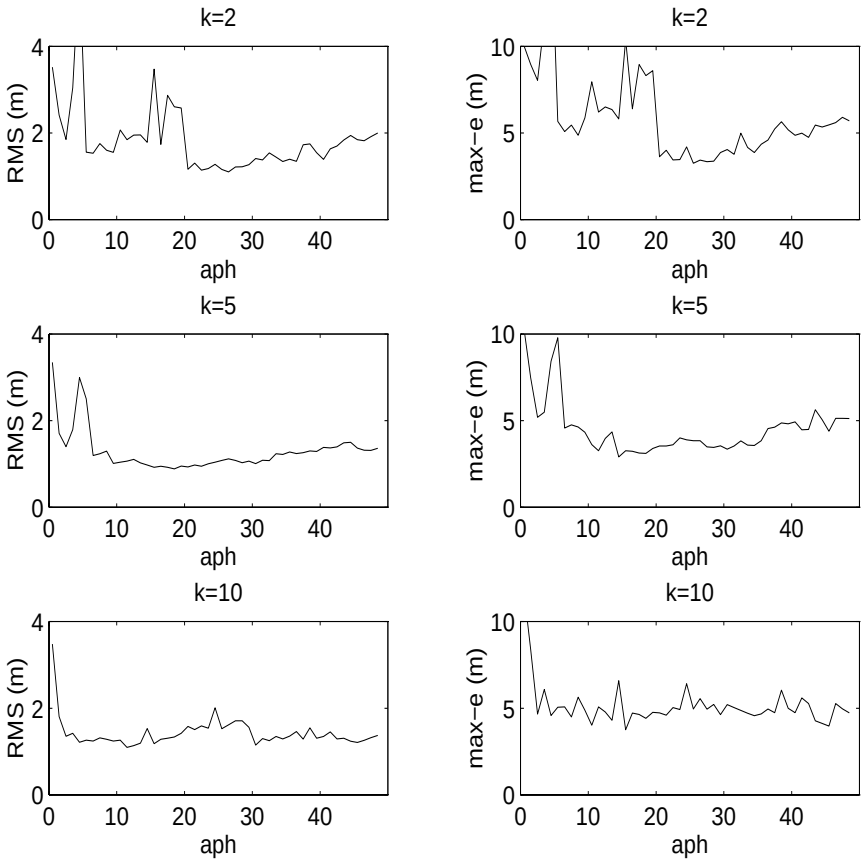

Figure 5.4: Errors of the computed positions, third track

From all the above figures, we can see that if α is too small the results are poor. We need to use a large value of α, say $\alpha = 10$, to get good results. Then for the sampling interval $\Delta t = 2.0$ seconds, at the instant when the present position fix is produced, the position potential function of the preceding position fix will be multiplied by $e^{-10\times 2} = 2.06 \times 10^{-9}$, but that of the present fix by $e^{-5\times 0} = 1.0$ (see *eqn.*(3.5)). So the influence of all the past position fixes can be neglected without much of an effect on the solutions, i.e., only the most recent position fix is really needed in *eqns.*(3.10) and (3.11). The consequence of this is that we can handle the case of non-diagonal covariance matrices by the coordinate transformation (described in **appendix II**) and then use the solution derived in *Chapter* 3. We have repeated the above simulations with the simplified coefficient matrices in the equation of the motion (i.e., including only the most recent position fix in *eqns.*(3.10) and (3.11)). To distinguish these simulations from those with complete coefficient matrices, we call them the "simulations with simplification". The differences caused by the simplification (complete minus simplified) are presented in Figs. 5.5 to 5.7 for the three simulated tracks. In these figures "D-RMS" represents the difference in RMS and "D-max-e" represents the difference in the maximum estimated position error (max-e), positive values among the them indicates that the simplification led to better result and vice versa. From Figs. 5.5 to 5.7, we can see that the simplification causes little

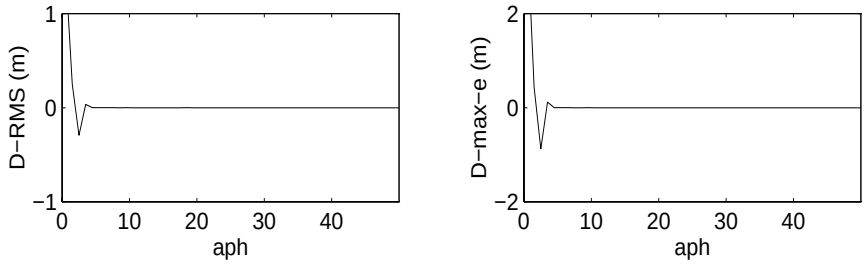

Figure 5.5: Differences caused by simplification, first track, $k = 15$

difference when α is large enough, say about 25.

To conclude this section we would like to point out that:

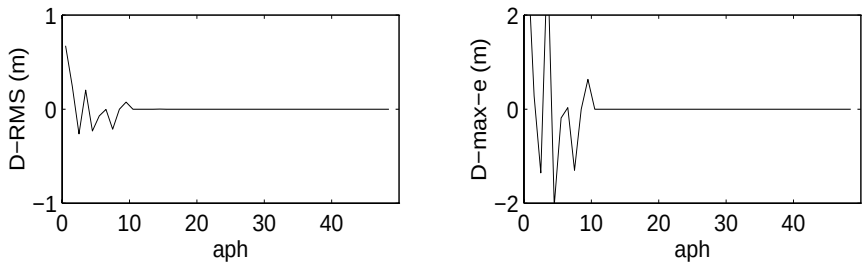

Figure 5.6: Differences caused by simplification, second track, $k = 10$

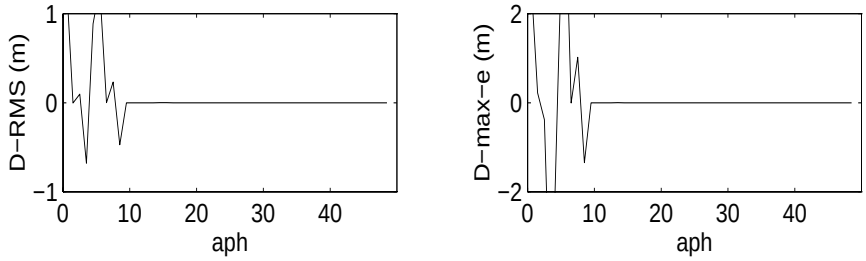

Figure 5.7: Differences caused by simplification, third track, $k = 5$

(1) the selection of k would affect the results, the best value of k depends on whether the vehicle is under maneuvering and how heavy the maneuvering is: the sharper the maneuvering the smaller the best k is.

(2) A small value of α can lead to large errors in results while it is safe to use a large value of α, say $\alpha = 25$. The large value of α enable us the simplification on the solution of the motion of the particle. We have done many simulations with other different tracks about the simplification, the results of these simulations verified the simplification with one accord. In the rest of this dissertation, all the demonstrated the SLPND's results are obtained by the simplified equations.

5.3 effect of initial velocity

To test the SLPND with different initial velocities, now we use another set of initial velocity as follows:

$$\dot{x}_0 = 1.0\ m/s, \quad \dot{y}_0 = -0.5\ m/s \tag{5.9}$$

with a covariance matrix

$$\mathbf{C}_{v_0} = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix} \; m^2/s^2. \tag{5.10}$$

The changes in RMS and maximum estimated position error caused by the change of initial velocity are shown in Fig. 5.8 for the first track. We have done many

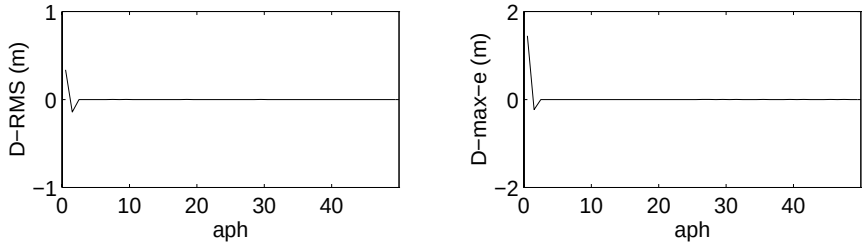

Figure 5.8: Difference caused by different initial velocities

simulations with other initial velocities on other different tracks. All these simulations have revealed that different initial velocities cause little difference in the results if α is large and if the covariance matrix of the initial velocity has well described the uncertainty in the initial velocity.

5.4 effect of sampling rate

In order to test the SLPND with data of different sampling rate, we assume that the position is observed every 10 seconds ($\Delta t = 10\ seconds$) instead of 2 seconds as before and the error set of the simulated data remains unchanged. It is no doubt that for a vehicle under maneuvering, with a larger sampling interval, we can not get better results. So in this section we only use the straight track. With $k = 15$, the RMS and maximum estimated position error versus α are shown in Fig. 5.9. Comparing Fig. 5.9 with the plotting of $k = 15$ in Fig. 5.2, we can see that the results with a larger sampling interval is, as we have expected, not so good as that with a smaller sampling interval but is still much better than the original data.

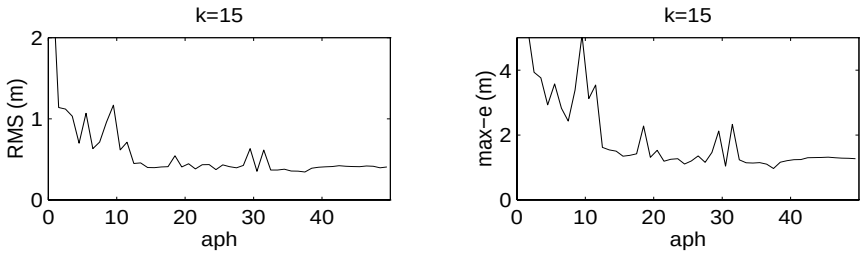

Figure 5.9: RMS and maximum estimated position error, $\Delta t = 10\ s$

5.5 effect of observational errors

5.5.1 random error

Two other error sets which share the same Gaussian population as the previous one but are different realizations are also used. Figure 5.10 depicts the RMS and maximum estimated position errors versus α (for $k = 15$) for the three sets of data of the first track. The three results give the same tendency. We have done similar

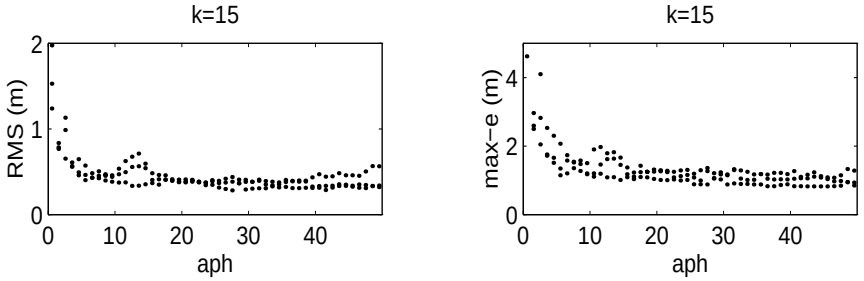

Figure 5.10: RMS and maximum estimated position error

simulations with the second track and have observed the same phenomenon.

Now let us test another error set which is from another Gaussian population: it has also zero means but standard deviations $\sigma_x = \sigma_y = 2\ m$ and $\sigma_{xy} = 0$; the maximum position error of the simulated fixes is 7.8 m (almost being two times as long as the distance between two adjacent fixes). We set $k = 15$. The RMS and maximum estimated position error versus α are depicted in Fig. 5.11 for the first track and Fig. 5.12 for the second. Even with so large uncertainty in the observations,

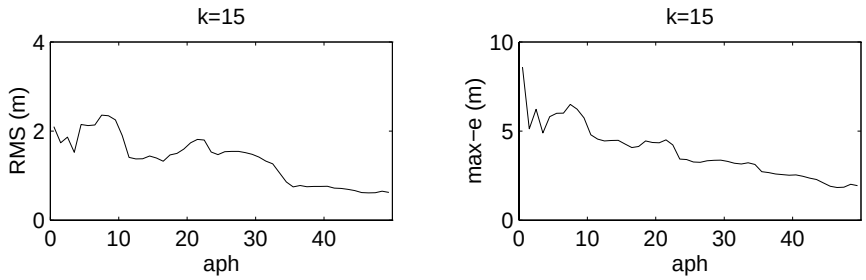

Figure 5.11: RMS and maximum estimated position error, straight track

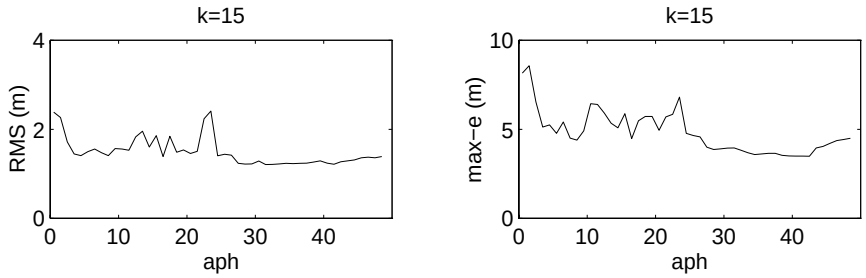

Figure 5.12: RMS and maximum estimated position error, circular track

the SLPND can still offer good results. Like the previous cases of $\sigma_x = \sigma_y = 1\ m$, we need use large α to get good results. But to the second track, in this present case after $k = 10$, an increase of k by 5 (i.e., $k = 15$) improves the result a lot. This may be because the random error in the case is larger than the error caused by having not informed the SLPND the constant maneuvering.

From the simulations in this section, we can see that SLPND can suppress the random noise in observations.

5.5.2 systematical error

Let us suppose now that observed positions are contaminated by a systematic errors in both x and y directions specified by the following equation:

$$e = 2\sin(\frac{\pi t}{3}), \tag{5.11}$$

where e denotes the position error in metres, and t is time in seconds. The errors of so generated positions are depicted in Fig. 5.13. Although the matter using a filter

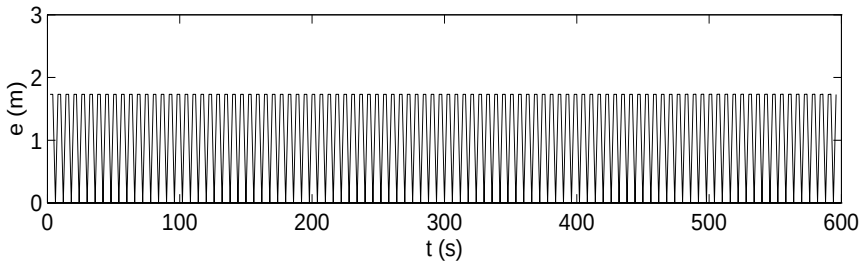

Figure 5.13: Errors of the simulated positions

based on error statistics to this problem might be not proper to some reader, we still use the SLPND to this problem. Even further, we also use RMS for judgment. With $k = 15$, the RMS and maximum estimated position error versus α are depicted in Fig. 5.14 for the first track and in Fig. 5.15 for the second. As indicated by these

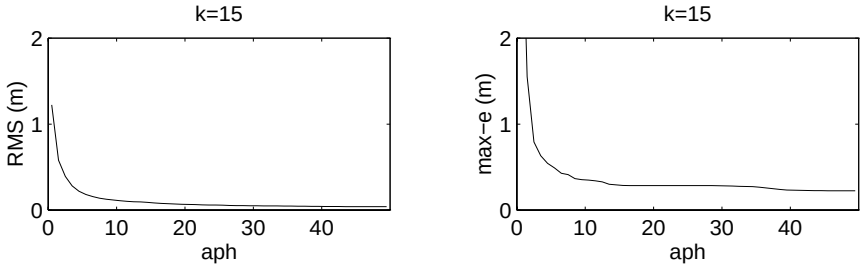

Figure 5.14: RMS and maximum estimated position error, first track

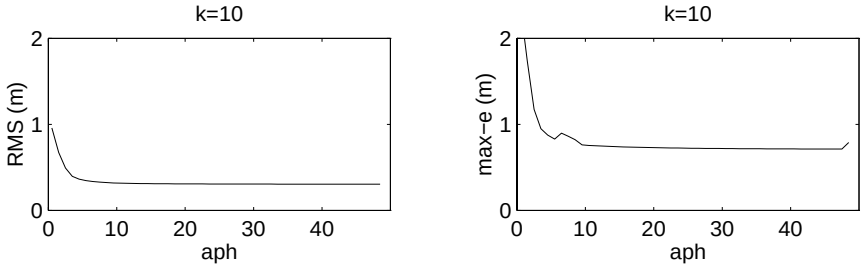

Figure 5.15: RMS and maximum estimated position error, second track

two figures, good results can be obtained with a large α, and the SLPND can also suppress systematical error in observations, with the maximum estimated position error about 0.43 m in the first track and about 0.71 m in the second track compared

with the maximum simulated position error of 1.732 m.

5.6 two straight lines at a right angle

In applications such as acceleration burst at geophysic exploration at sea or tractor operations in agriculture, tracks of vehicles are often straight lines at right angles. In fact, when the vehicles makes a sharp turn, the navigator (man or computer) knows about the change. Then why not let the navigator intervene with the filter operation? The SLPND is designed to make the navigator's interventions possible. In this section, we investigate the SLPND's ability to accommodate interventions.

The simulated track is formed by two straight lines with a right angle turn at the time instant of $t = 298s$. Without navigator's intervention (without informing the filter of the change) at the turning point, right after the turning point, the errors of the estimated positions reachs their worst (6.1 m), and the filter spend more than 50 seconds to come back to normal. It is easier to understand this if we recall the way we built the motion model (*eqn.*(3.41)). The model is totally dependent on the observed position fixes and their error statistics. Certainly, if the vehicle suddenly changes its speed (magnitude and/or direction), the kinematic information from the motion model has one position fix lag to the true kinematics of the vehicle, then the particle would overshoot.

The SLPND can accept the navigator's intervention by getting the velocity value at the turning point or the velocity value with its covariance matrix, or simply a covariance matrix assigned to the SLPND's computed velocity at the turning point. This can easier be understood by summoning *eqn.*(3.41) here:

$$\mathbf{S}_{i+1} = \mathbf{f}_i(\mathbf{\Theta}, \mathbf{S}_i).$$

Suppose that the position fix i is the turning point. The above formula clearly demonstrates that $\mathbf{S}_i$ (the state of the motion at the turning point i) is the initial condition

for the motion evolving to time instant t_{i+1}. If we update the velocity component in $\mathbf{S}_i$, we can implement the intervention. The update can be done either by replacing the SLPND's computed velocity at the turning point (obtained by evolvement from t_{i-1} to t_i) with the velocity observation and its covariance matrix or simply by assigning a covariance matrix to the SLPND's computed velocity. To the latter, this means telling the filter a new uncertainty information of its computed velocity at the turning point. Here, the computed velocity is then regarded as the velocity 'observation', the larger the uncertainty is, the less the computed velocity effects the evolvement from t_i to t_{i+1}.

We give the SLPND the additional velocity observation and its covariance matrix at the turning point. We assume the velocity observation at the turning point to be

$$\mathbf{v}_{t=298} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 0.1 \\ 1.9 \end{pmatrix} \; m/s \qquad \mathbf{C}_{v_{t=298}} = \begin{pmatrix} 0.01 & 0.0 \\ 0.0 & 0.01 \end{pmatrix} \; m^2/s^2. \quad (5.12)$$

The true forward velocity at the turning point was $\dot{x} = 0.0 \; m/s$ and $\dot{y} = 2.0 \; m/s$. We set $k = 15$ and $\alpha = 25.5$. The errors of the computed (estimated, predicted and partially smoothed) positions are pictured in Fig. 5.16. Here, the errors (i.e., e in the figure) of the computed position are the the norm of the difference vectors between the true positions and the computed positions, and the smoothed position is about the middle position fix among those used for parameter estimation.

Now we try to intervene the SLPND by assigning the following covariance matrix $\mathbf{C}_{v_{t=298}}$ to its computed velocity at the turning point:

$$\mathbf{C}_{v_{t=298}} = \begin{pmatrix} 9.0 & 0.0 \\ 0.0 & 9.0 \end{pmatrix} \; m^2/s^2. \quad (5.13)$$

The errors of the computed positions are depicted in Fig. 5.17.

It seems that with the navigator's intervention the SLPND can provide much better results after the turning point: the maximum estimated position error is 1.1 m in Fig. 5.16 and 1.5 m in Fig. 5.17 compared with 6.1 m in the case without intervention.

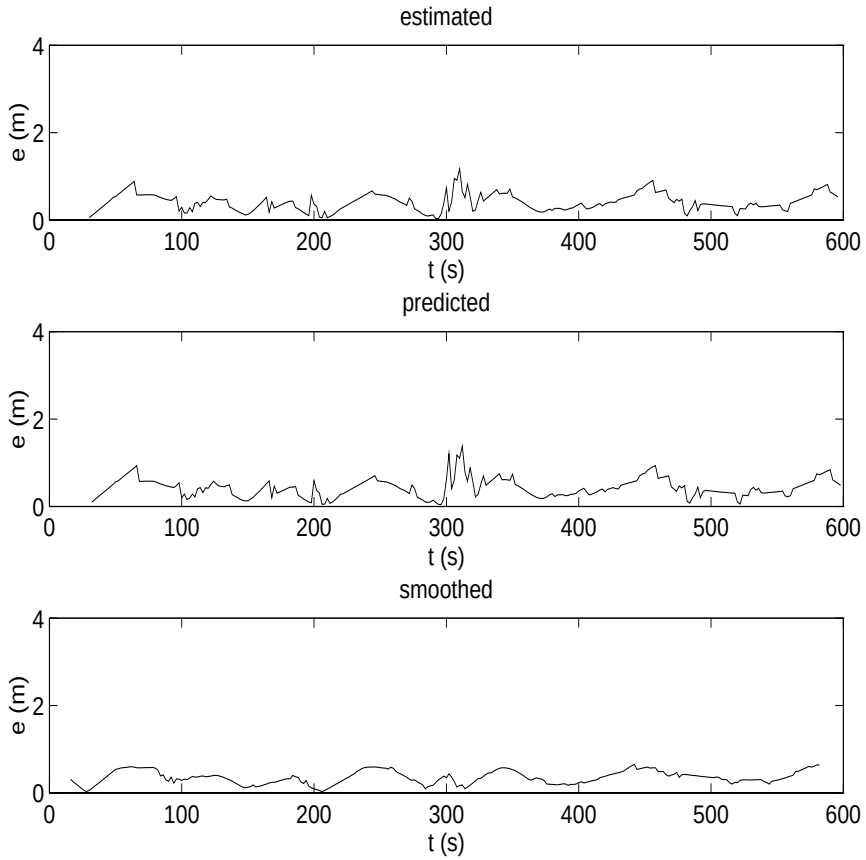

Figure 5.16: Errors of the computed positions

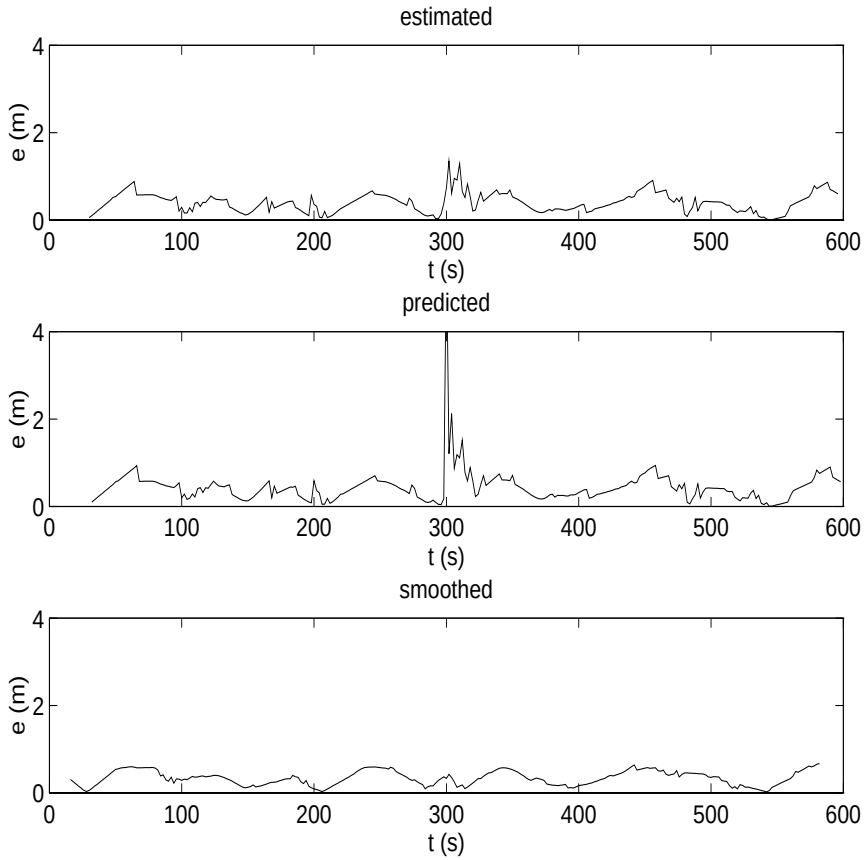

Figure 5.17: Errors of the computed positions

5.7 SLPND's features, (I)

From the above simulations, we get some feelings about the SLPND as follows.

1. Even without velocity information, the SLPND can learn, to a certain extent, about the kinematic of the vehicle when the vehicle is under a slow and constant (eg circular track) maneuvering. But when the vehicle is under a heavy maneuvering (eg sinusoidal track), what it learned may be not enough to improve position accuracy. In this kind of situation it is not enough to supply only position information to the SLPND. We expect that the SLPND would give better results if it is fed with both position and velocity observations.

2. The selection of k would effect the results and there is no general k for various tracks. The best value of k depends on whether the vehicle is under maneuvering and how heavy the maneuvering is: the sharper the maneuvering the smaller the best k is.

3. A small value of α can lead to large errors while it is safe to use a large value of α, say $\alpha = 25$. The large value of α leads to that only the most recent position fix needs to stay in *eqns.*(3.10) and (3.11) and enables us the simplification on the solution of the motion of the particle.

4. Different initial velocities will cause little difference in the results if α is large and if the covariance matrix of the initial velocity has well described the uncertainty in the initial velocity.

5. The SLPND can suppress random errors and high frequency systematic errors.

6. An intervention from the navigator can be easily implemented. When the vehicle is under sharp maneuvering, giving the SLPND an extra information about the kinematic change, even a guess at the vehicle's velocity or simply uncertainty information on its computed velocity is very helpful.

Chapter 6

applications to real tracks

This chapter is devoted for the test of the SLPND with real track, using both position and velocity information. The SLPND is used on two tracks of MARY O (the research boat of the University of New Brunswick) observed by the Ocean Mapping Group of the Department of Geodesy and Geomatics Engineering of UNB. The observed positions of these two tracks contains biases. This prevents us from testing the SLPND's covariance matrix estimation. To overcome this, we generate simulated data based on the true tracks of MARY O (obtained from the post-processing and regarded as the true tracks). First, we apply the SLPND on the simulated data. Then, the real position data with the simulated velocity data (no real velocity data available) are used. The SLPND is tested as an estimator, predictor, smoother and blunder detector.

6.1 track description

The first track was observed on September 12, 1993 in Saint John harbor and the second March 26, 1994 in the Bay of Fundy. Both had a sampling rate of 10 seconds. When MARY O was working, its positions were determined by Differential Global Positioning System (DGPS) with PNAV, the Precise Differential GPS Navigation

and Surveying software (Ashtech, 1993). C/A-code pseudo-range and carrier phase were observed. The real-time positions of MARY O were computed by PNAV, using carrier phase smoothed C/A-code pseudo-range. The available accuracies and the corresponding requirements for PNAV have been listed in Table 1.4. It has been claimed to have a processing accuracy of 1-3 metres for the real-time output, using the mode of smoothed C/A-code pseudo-range with PDOP < 4.0. More accurate results can be obtained by PNAV's post processing procedure. It has been said that the post processing accuracy of the PNAV is on the level of 0.05-0.3 metres, with mode of pseudo-ranges plus carrier phase and float integer ambiguities and by forward and backward processing method (Ashtech, 1993). According to the post processing results, kindly offered by the Ocean Mapping Group, the internal accuracy of positions after the post processing is on the level of 0.05 m. We assume that there is no bias in the position output of the post processing and regard them as the true positions. Figure 6.1 depicts the two tracks of MARY O; it is obvious that the boat was always under maneuvering during the two campaigns.

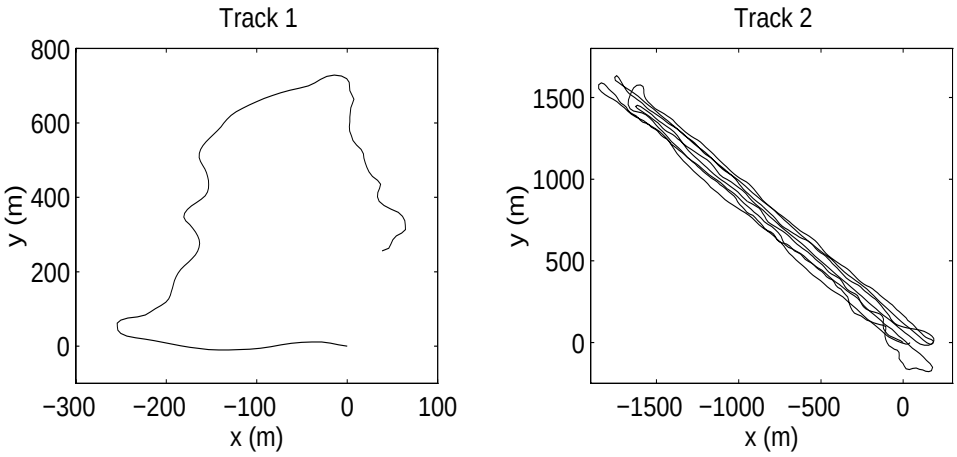

Figure 6.1: Two Tracks of MARY O

6.2 results with simulated data

6.2.1 data description

To get simulated position data of the two tracks, we add Gaussian noise to PNAV's post processed positions. Two Gaussian noise series (one for the x-component the other for the y-component) for the positions have mean 0 and standard deviation 1 m, and are not correlated with each other. We difference PNAV's post processed positions and then add Gaussian noise to get simulated velocity data. The two Gaussian noise series added to the velocities have mean 0 and standard deviation 0.05 m/s and are also not correlated with each other.

When velocity data are available, the dead-reckoning technique may be used to navigate the vehicle as long as the initial position of the vehicle is known. The model for this technique can be expressed as follows.

$$\mathbf{r}_n = \mathbf{r}_{n-1} + \dot{\mathbf{r}}_{n-1}\Delta t, \tag{6.1}$$

where, $\mathbf{r}_n$ represents the position of the vehicle at t_n, $\dot{\mathbf{r}}_n$ is the velocity vector at t_n, and Δt is the time increment from t_{n-1} to t_n.

The simulated position errors for the both tracks are depicted in Fig. 6.2 and Fig. 6.4. The maximum simulated position error is 3.9 m for the first track and 4.1 m for the second. The position errors of the dead-reckoning technique are depicted in Fig. 6.3 with a maximum position errors of 7.7 m for the first track and Fig. 6.5 with a maximum position errors of 14.8 m for the second.

6.2.2 positioning results

A search area for unknown parameters G is defined by $0 \leq G \leq 1.0$. The RMS and maximum estimated position error versus α and k are depicted in Fig. 6.6 for the first track and Fig. 6.7 for the second. When α is large enough, the RMS for both tracks are lower than 1 m, and the maximum estimated position errors are lower than

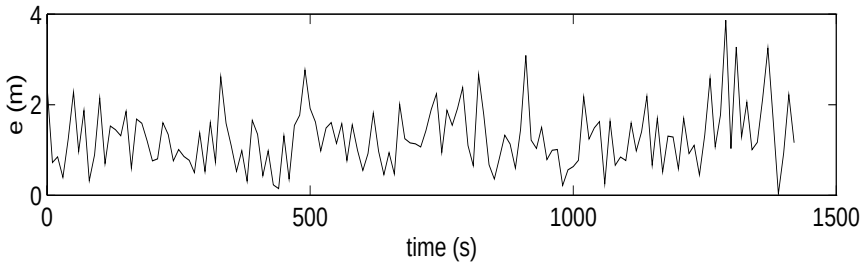

Figure 6.2: Errors of simulated positions, Track 1

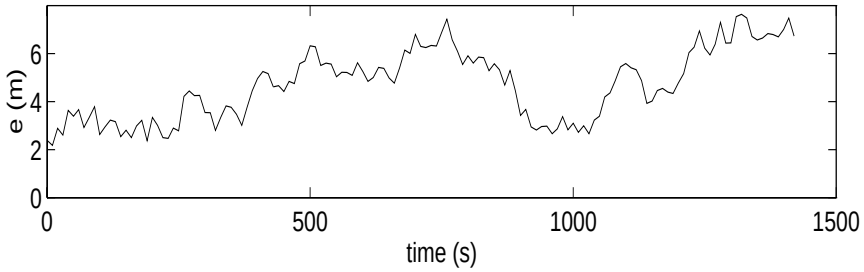

Figure 6.3: Position errors using dead-reckoning, Track 1

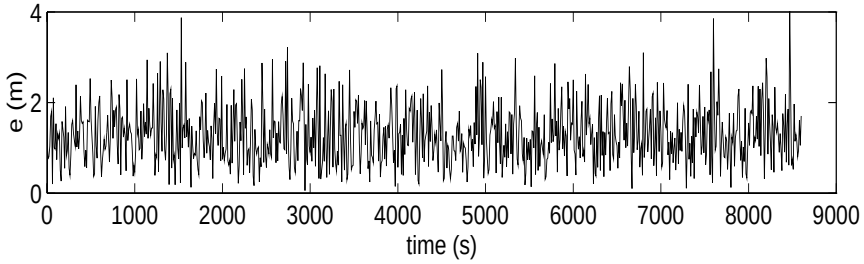

Figure 6.4: Errors of simulated positions, Track 2

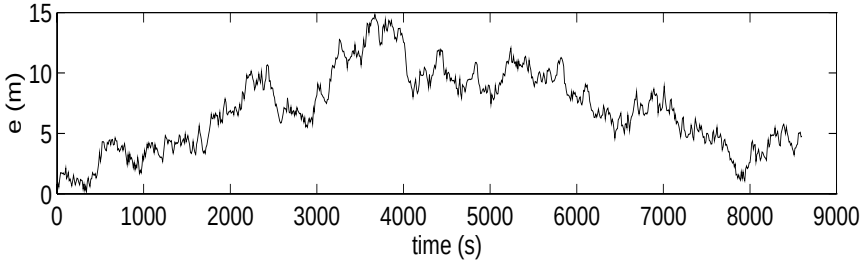

Figure 6.5: Position errors using dead-reckoning, Track 2

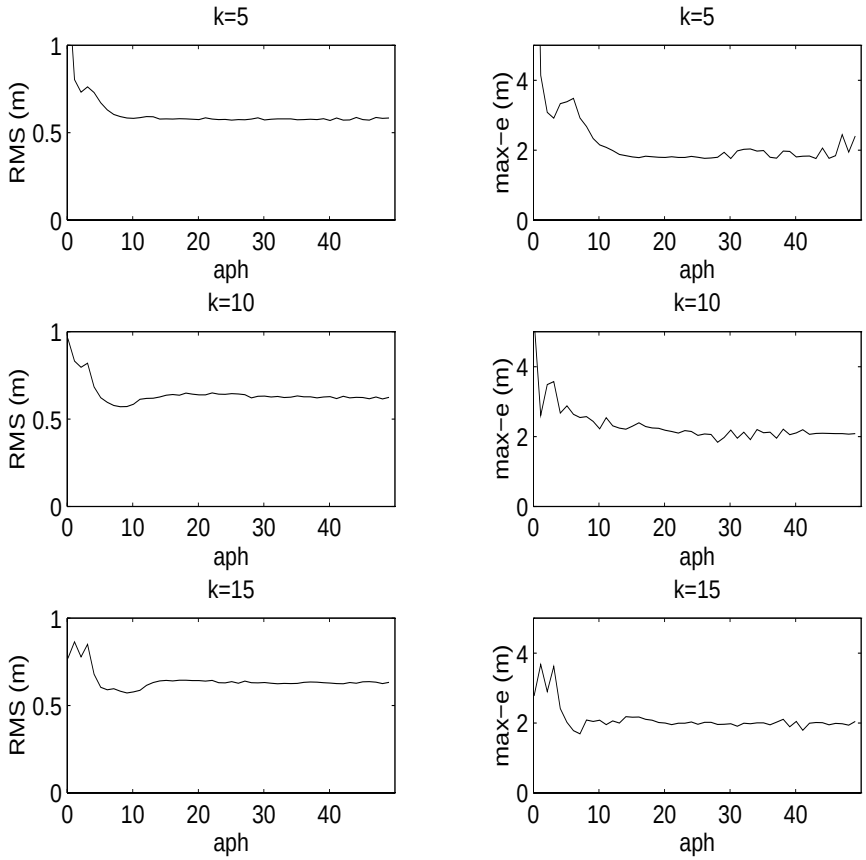

Figure 6.6: RMS and maximum estimated position error, Track 1

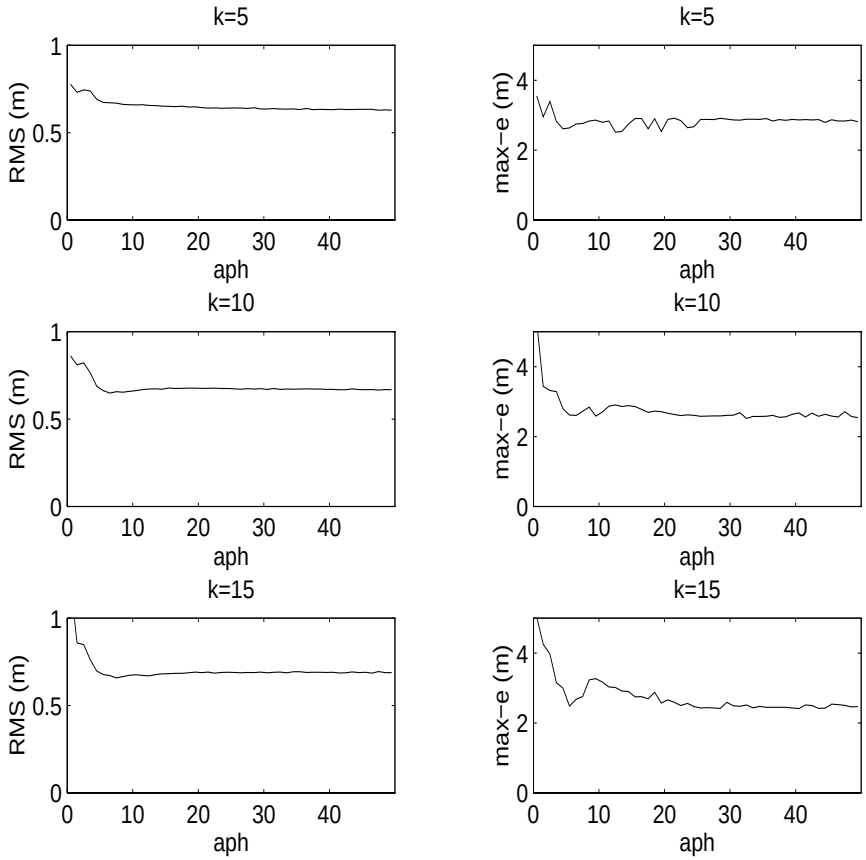

Figure 6.7: RMS and maximum estimated position error, Track 2

the maximum simulated position errors (3.9 m for the first track and 4.1 m for the second). It seems that the larger k might not give better results. This supports our conclusion on the k issue in the end of the last chapter. With $\alpha = 25.5$ and $k = 10$, the errors of the computed positions (the estimated, predicted and smoothed positions) are shown in Fig. 6.8 for the first track and Fig. 6.9 for the second. The errors of the smoothed positions depicted is about the middle fix (the sixth one) among the 10 involved position fixes used to estimate the parameter G. Among the smoothed, estimated and predicted positions, as they should be, the smoothed positions have the highest accuracy, and then the estimated positions.

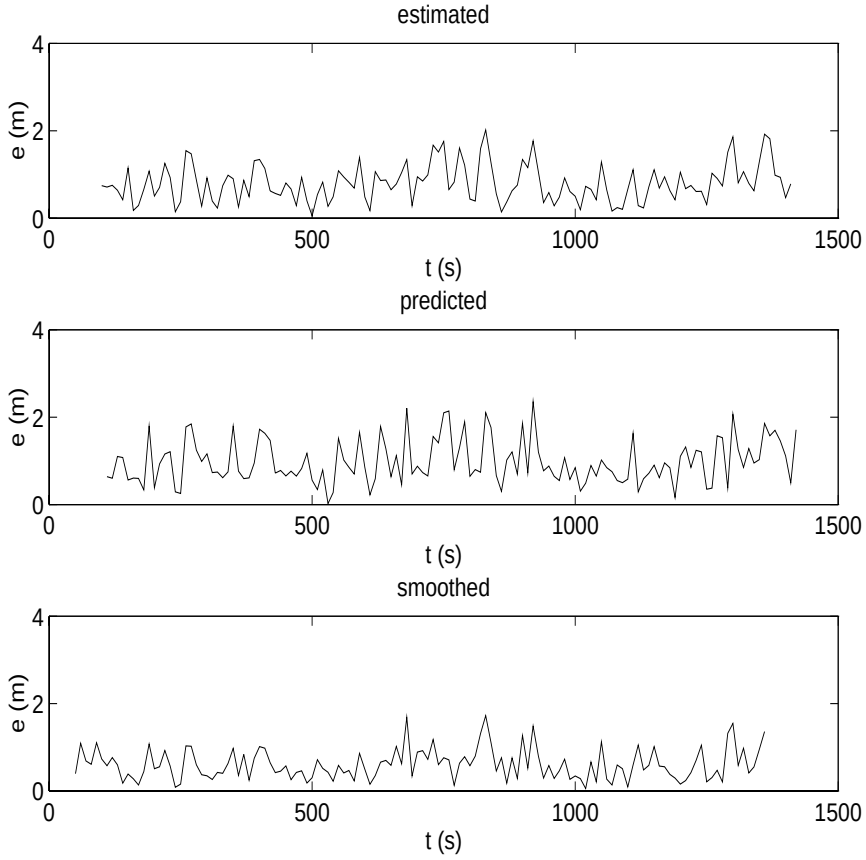

Figure 6.8: Errors of the computed positions, Track 1

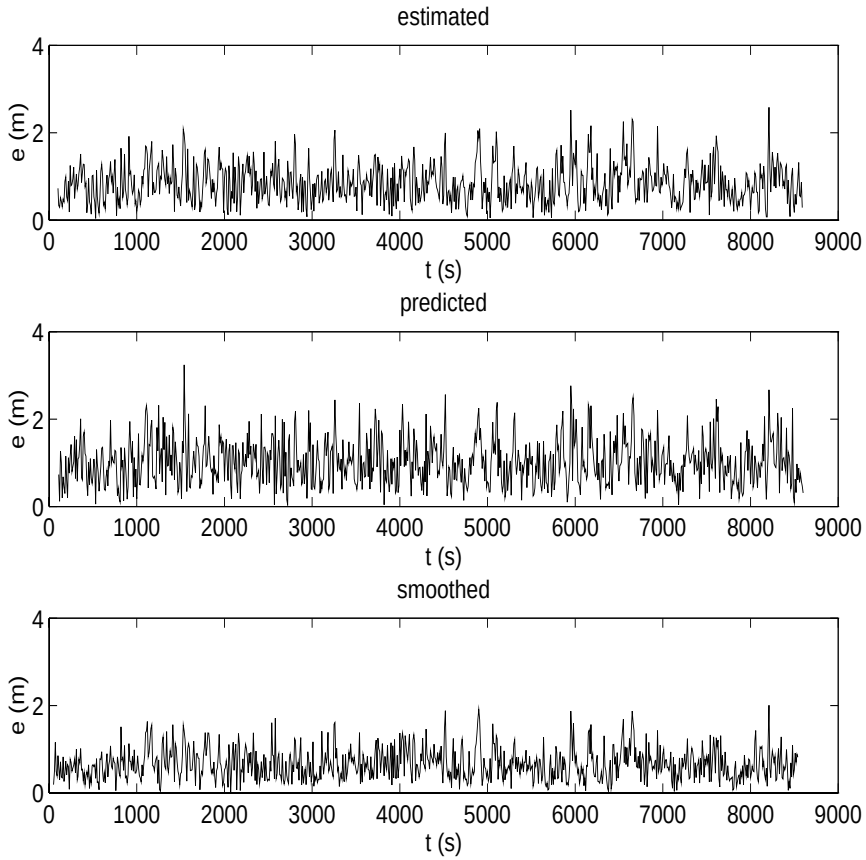

Figure 6.9: Errors of the computed positions, Track 2

6.2.3 test of the covariance matrix estimation

We make error ellipses [Vaníček and Krakiwsky, 1986] at different confidence levels for each computed position based on the corresponding covariance matrix. Then we test the compatibility between the computed position and the corresponding true position and enumerate the compatible pairs. Table 6.1 shows the percentages of the compatible pairs. For the first track, the number of tested position fixes is 132, and for the second, the number is 850. The reasonable percentages mark the validity of the estimation of the covariance matrices.

Table 6.1: Percentage of compatibility by χ^2-test, $k = 10$

Track	Sample	Type	Confidence level (%)							
			99.9	99.0	98.0	95.0	90.0	80.0	70.0	50.0
1	132	Est.	100.0	97.0	94.7	91.7	83.3	74.2	65.9	44.7
		Pre.	98.5	98.5	96.2	94.7	84.8	77.3	71.2	51.5
		Smo.	100.0	99.2	98.5	95.5	90.2	82.6	75.0	59.1
2	850	Est.	99.2	97.4	95.3	89.5	82.5	69.6	57.8	40.0
		Pre.	99.6	98.5	96.5	92.7	88.1	77.2	66.7	44.4
		Smo.	99.8	98.4	97.2	94.4	88.4	80.2	69.9	49.9

Table 6.2: Percentage of compatibility by Fisher-test, $k = 10$

Track	Sample	Type	Confidence level (%)				
			99.9	99.0	97.5	95.0	90.0
1	132	Est.	100.0	100.0	99.2	97.7	93.9
		Pre.	99.2	99.2	99.2	96.9	92.3
		Smo.	100.0	100.0	100.0	97.7	95.4
2	850	Est.	100.0	99.3	98.2	96.2	90.6
		Pre.	100.0	99.6	98.6	96.5	92.7
		Smo.	100.0	100.0	98.6	97.6	94.8

6.2.4 blunder detection

Two blunders are added to the first set of data. The first one is added to at $t = 400$ seconds with a magnitude of 10 metres in x direction, and second one at $t = 800$ seconds with the same magnitude as the first one but in y direction. The errors of the simulated positions are depicted in Fig. 6.10. We set a confidence level $1 - \alpha = 0.999$,

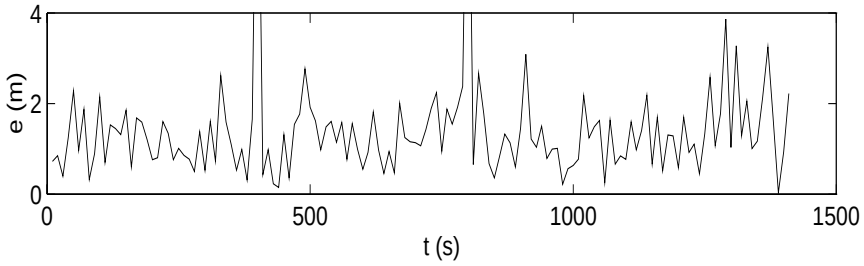

Figure 6.10: Errors of simulated positions

then the corresponding threshold value is $\xi_{\chi^2(2),1-\alpha} = 13.816$. With $\alpha = 25.5$ and $k = 10$ the errors of estimated positions without detection and with detection are shown in Fig. 6.11 and Fig. 6.12 respectively. With the blunder detection function used,

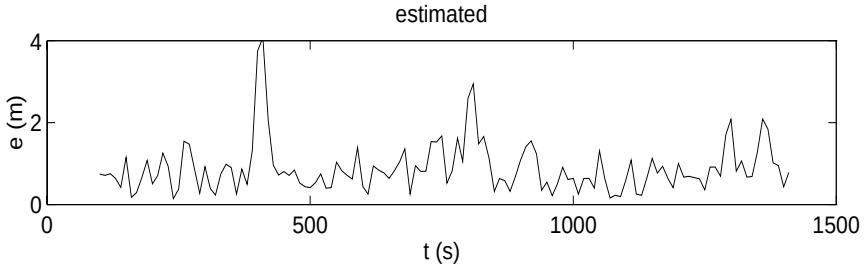

Figure 6.11: Errors of estimated positions, without detection

the two blunders have been successfully detected and eliminated, and the SLPND gives good results, see Fig. 6.12 (cf. Fig. 6.10). We can also see from Fig. 6.11 that even without blunder detection, the effect of the blunders is suppressed.

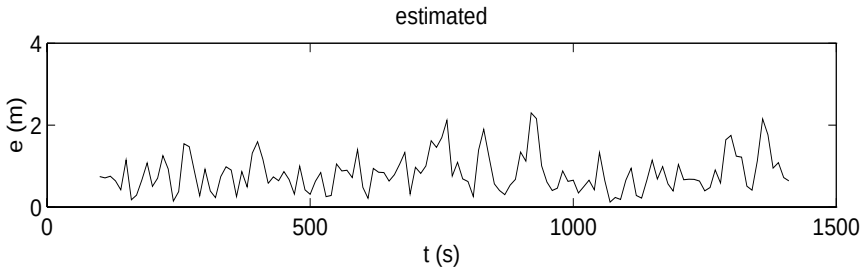

Figure 6.12: Errors of estimated positions, with detection

6.3 results with real position and simulated velocity data

6.3.1 data description

Now we take PNAV's real-time position outputs as the position observations for further processing with the SLPND. Due to the lack of real velocity observations, we use simulated velocity data obtained by the same way as we did in the last section. The errors in the velocity data have a standard deviation of 0.01 m/s in both x and y components and no cross correlation between these two components. The observed position errors are depicted in Fig. 6.13 with a maximum observed position error of 2.2 m for the first track, and Fig. 6.14 with a maximum observed position error of 3.2 m for the second. The position errors by dead reckoning are depicted in Fig. 6.15

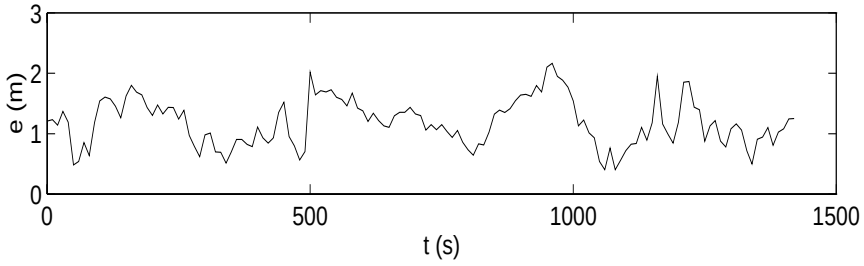

Figure 6.13: Errors of the observed positions, Track 1

for the first track and Fig. 6.16 for the second. The maximum position error is 2.3 m

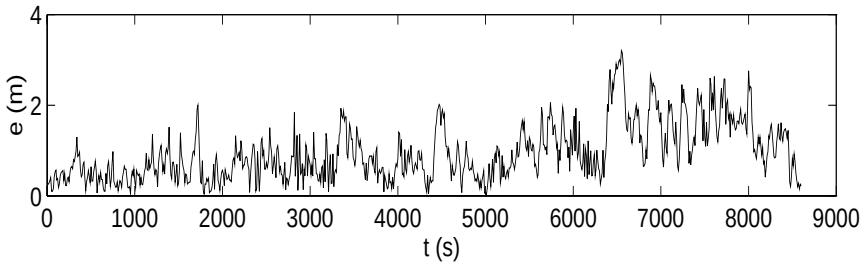

Figure 6.14: Errors of the observed positions, Track 2

in Fig. 6.15 and 4.4 m in Fig. 6.16.

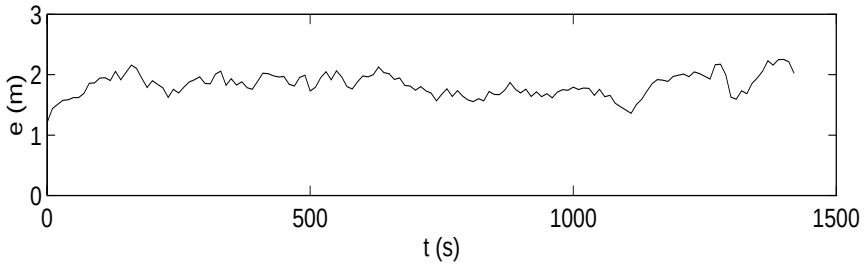

Figure 6.15: Position errors by dead reckoning, Track 1

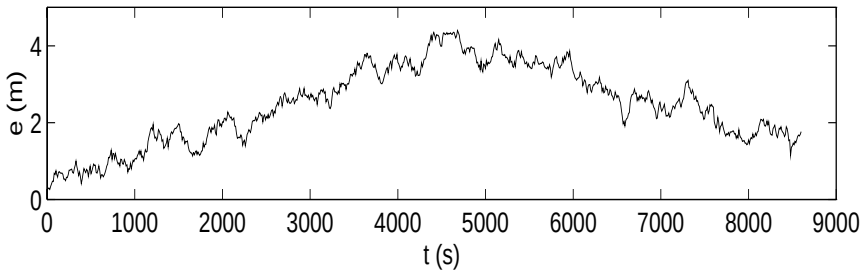

Figure 6.16: Position errors by dead reckoning, Track 2

6.3.2 positioning results

With $k = 10$, the RMS and maximum estimated position error versus α are depicted in Fig. 6.17 for the first track and Fig. 6.18 for the second. For $\alpha = 50$ and $k = 10$, the errors of the computed positions are depicted in Fig. 6.19 for the first track and Fig. 6.20 for the second. The maximum position error is reduced from 2.2 m to 1.6

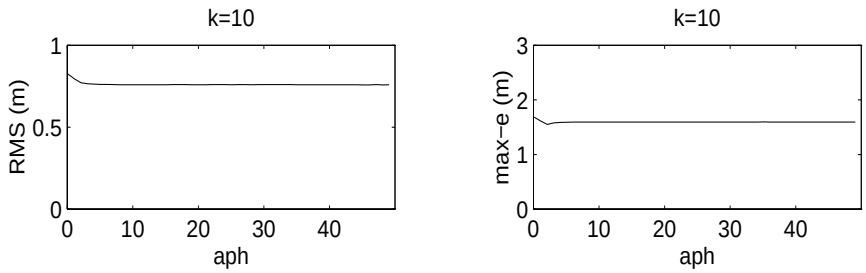

Figure 6.17: RMS and maximum estimated position error, Track 1

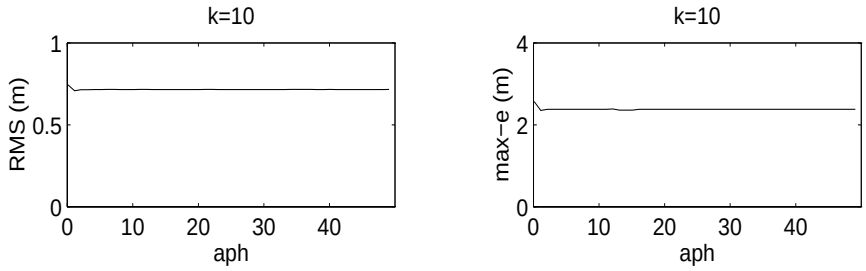

Figure 6.18: RMS and maximum estimated position error, Track 2

m for the first track and from 3.2 m to 2.4 m for the second. In fact, α might be any value larger than 2 for a result as good as the above.

6.3.3 test of the covariance matrix estimation

The compatibility test between the computed positions and the positions obtained by PNAV's post processing is made by Fisher-test [Vaníček and Krakiwsky, 1986]. The results of the test are shown in Table 6.3.

It has been indicated by this table that a large number of computed positions are not compatible with the PNAV's post processing results. What is the reason? The requirements for $eqn.(4.36)$ are that $(\mathbf{r}-\hat{\mathbf{r}})^T\mathbf{C}^{-1}_{\Delta\hat{\mathbf{r}}}(\mathbf{r}-\hat{\mathbf{r}})$ is a χ^2 statistic with 2 degrees of freedom, $(\nu\hat{\sigma}_0^2/\sigma_0^2)$ is also a χ^2 statistic but with ν degrees of freedom and these two statistics are independent. According to Vaníček and Krakiwsky [1986], these two statistics may not be χ^2 distributed for a number of reasons:

(a) the non-normal density of the $(\mathbf{r}-\hat{\mathbf{r}})$,

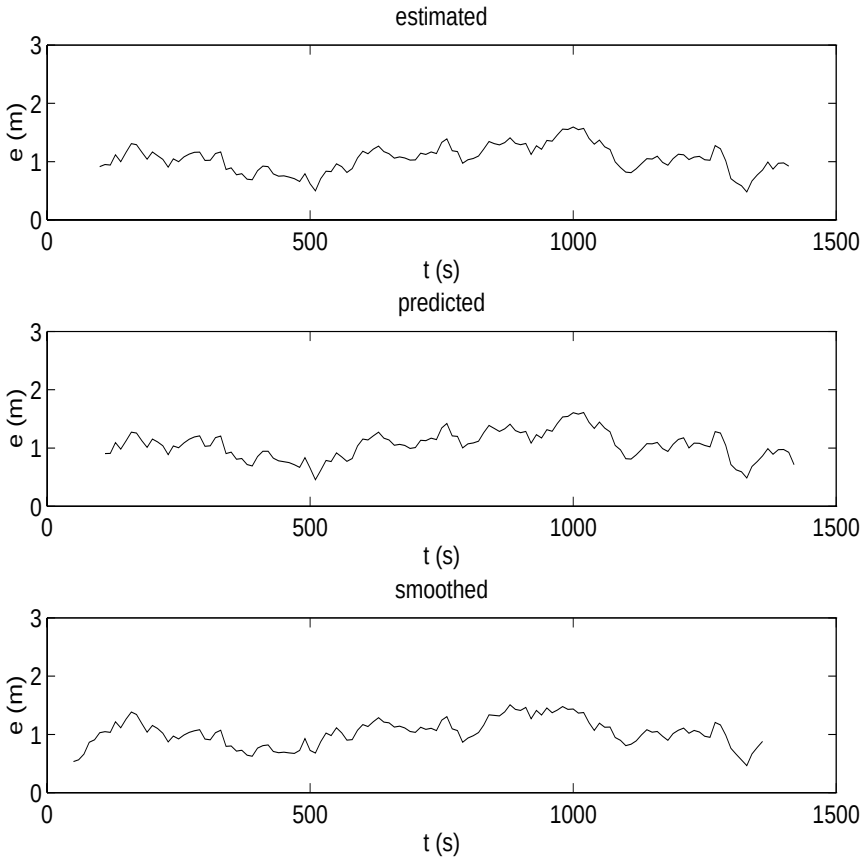

Figure 6.19: Errors of computed positions, Track 1

Table 6.3: Percentage of compatibility by Fisher-test

Track	Sample	Type	Confidence level (%)				
			99.9	99.0	97.5	95.0	90.0
1	132	Est.	19.8	6.0	0.0	0.0	0.0
		Pre.	35.9	14.5	5.0	2.0	0.0
		Smo.	5.0	0.0	0.0	0.0	0.0
2	850	Est.	38.9	28.4	22.6	17.4	12.5
		Pre.	59.5	45.8	35.5	28.2	22.9
		Smo.	24.3	14.5	12.2	10.0	8.0

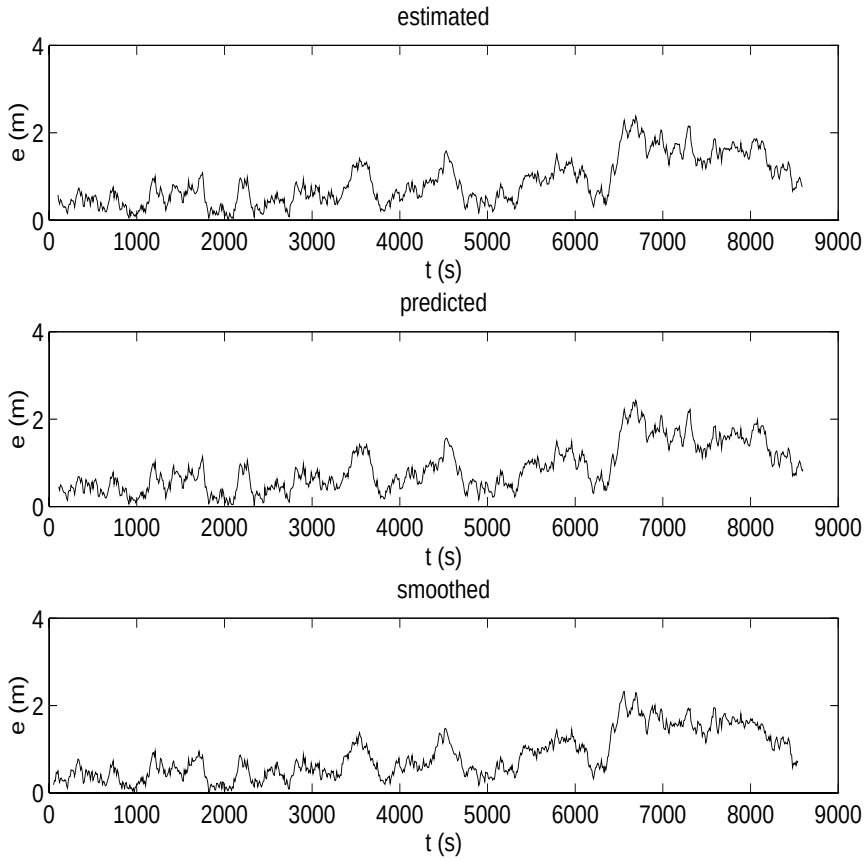

Figure 6.20: Errors of computed positions, Track 2

(b) the incorrect methematical model,

(c) the presence of systematic errors in the observations, and

(d) the incorrect a priori covariance matrix of the observations.

The successes on the error estimations and the blunder detections with the simulated data in the last section eliminated the possibility of reason (b). So there must be something wrong with either the observed data or the PNAV's post processing results. We plot the differences between the observed position data and those obtained from PNAV's post processing in both x and y directions in Fig. 6.21 for the first track and in Fig. 6.22 for the second one. Indeed, if PNAV's post processing has given

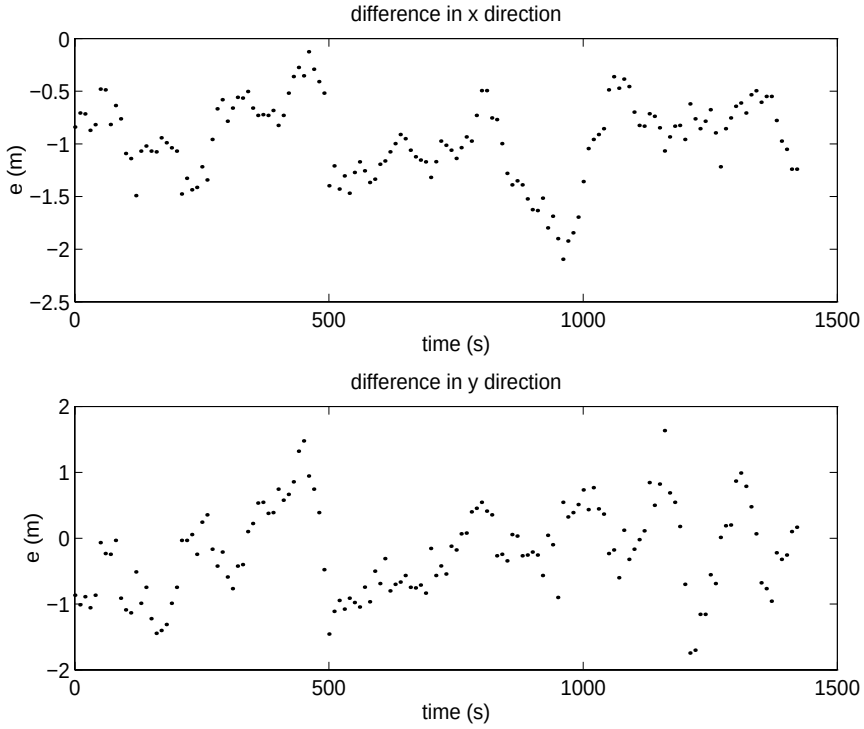

Figure 6.21: Difference between real-time and post processing results, Track 1

very accurate positions, both sets of the observed data were contaminated by the systematic errors. In the first track, the observed positions in x direction have about one metre bias. In the second track, the observed positions have bias in x direction beginning at about $t = 3300$ and in y direction beginning at about $t = 1500$ seconds.

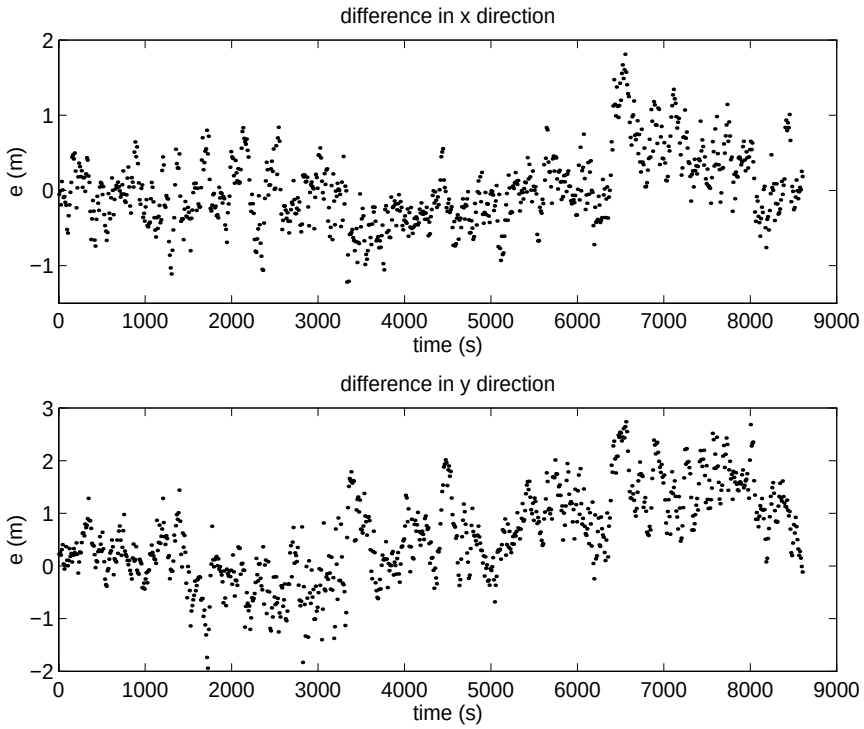

Figure 6.22: Difference between real-time and post processing results, Track 2

The systematic errors are the main reason for the unrealistic error estimations.

6.3.4 blunder detection

The SLPND's blunder detection is based on the covariance matrices of the predicted positions and the χ^2-test if the variance factor σ_0^2 for observation is known, or the Fisher-test if only its estimated value is known. If the obtained covariance matrices are not realistic, in fact in real navigation we may not know of this, what will happen with blunder detection? Two blunders are added to the first set of data (Track 1) in the same way as we did in the *Section* 6.2.4, and the errors of the simulated position are depicted in Fig. 6.23. And the errors of estimated positions without detection are shown in Fig. 6.24. With the confidence level $1 - \alpha = 0.999$, the corresponding threshold value $\xi_{F(2,19),1-\alpha} = 10.16$, not only are the two blunders detected but also many normal observations are regarded as blunders. We then continually increase

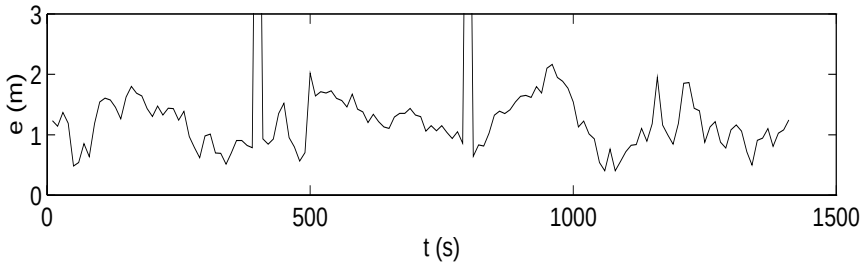

Figure 6.23: Errors of simulated positions

the threshold value $\xi_{F(2,19),1-\alpha}$ until only the two blunders are detected. The final threshold value is $\xi_{F(2,19),1-\alpha} = 22.5$. The errors of the estimated positions with detection are depicted in Fig. 6.25. We have set $\alpha = 50$ and $k = 10$.

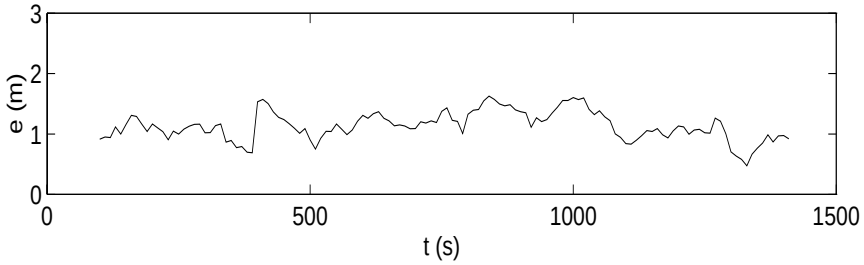

Figure 6.24: Errors of the estimated positions, without detection

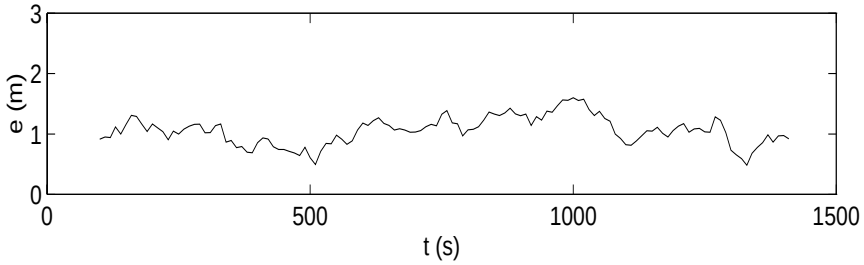

Figure 6.25: Errors of the estimated positions, with detection

The failure of the blunder detection at the confidence level $1 - \alpha = 0.999$ indicates that the values of the diagonal elements of the estimated covariance matrices are too small. The detections of the blunders are independent on PNAV's post processing results, so we suspect the failures of the Fisher-tests and blunder detections have been

more relevant to the observed data, i.e., PNAV's real-time solutions.

6.4 SLPND's features, (II)

1. When a vehicle is always under maneuvering, supplying not only position observations but also velocity observations to the SLPND can leads to good results.

2. The SLPND can act as an integrator to combine position and velocity observations, and then can supply navigation better than that using either positions alone or velocities alone (dead-reckoning). It is supposed to be able to integrate range or azimuth observations too: then the range observation will be regarded as a special error ellipse that has its minor axis of a limited length in the range direction and its major axis of an infinite length; and the azimuth observation as an ellipse that has its major axis of an infinite length aligned with the azimuth and its minor axis of a limited length orthogonal to the azimuth.

3. The SLPND requires the uncertainty of the position fixes only to be known in a relative size (cofactor matrix instead of covariance matrix).

4. The SLPND can also act as a blunder detector.

Chapter 7

comparisons between the SLPND and the existing navigation filters

In this chapter, we first compare the SLPND with the Kalman filtering technique and the polynomial filter conceptually. Then we compare the numerical results obtained respectively by the SLPND and the Kalman filter.

7.1 a conceptual comparison

In *Chapter* 2 we briefly described two of the existing navigation filters: the often used Kalman filter and the polynomial filter. Both these filters try to benefit from the kinematic information about the motion of the vehicle coded in the "process" models (*eqns.*(2.1) and (2.4)), and from the observation information written in a observation models (*eqns.*(2.2) and (2.5)). Similarly, the SLPND relies on *eqns.*(3.41) and (4.5). Table 7.1 lists the basic models used in these three filters.

The "process" model of the Kalman filter is a random model in which both the state vector and the "process" noise are regarded as independent random variables, separately modeled by two or more (corresponding to position, velocity and acceleration, etc.) different random processes. On the other hand, the "process" model of

Table 7.1: A Comparison of the Models

Name	Model	Formula	Type
Standard Kalman	Process	$\mathbf{S}_{n+1} = \mathbf{\Phi}_n \mathbf{S}_n + \mathbf{G}_n \mathbf{w}_n$	random, linear
	Observation	$\mathbf{l}_{n+1} = \mathbf{H}_{n+1} \mathbf{S}_{n+1} + \mathbf{v}_{n+1}$	random, linear
Polyn.	Process	$s_{n+1} = \sum_{i=0}^{m} \beta_i \phi_i(t_{n+1})$	deterministic, linear
	Observation	$\mathbf{l} + \mathbf{v} = \mathbf{\Phi}\beta,$	random, linear
SLPND	Process	$\mathbf{S}_{n+1} = \mathbf{f}_n(\mathbf{\Theta}, \mathbf{S}_n)$	deterministic, nonlinear
	Observation	$\mathbf{l} + \mathbf{v} = \mathbf{f}(\mathbf{\Theta}, \mathbf{l} + \mathbf{v})$	random, nonlinear

the polynomial filter is deterministic. The SLPND's motion model is deterministic too, with the understanding that it is based on uncertainties of the observed position fixes.

The standard Kalman filter allows no unknown parameter in its "process" model. The user has to give the filter all the parameters in advance, according to his/her experience. An adaptive Kalman filter may allow one or more unknown parameters and let the filter determine them from the observation series. The polynomial filter requires unknown parameters "β" in its "process" model. These parameters are, of course, determined from the observation series. The user has to determine only the order of the polynomial in advance. Selections of the parameters in the Kalman filter, or the degree of the polynomial in the polynomial filter are very important to navigation.

The SLPND's motion model has two parameters G and α. Both can be determined from the observation series, however, in order to increase the SLPND's processing speed, we usually fix α and let the filter determine G. Parameter G needs to be updated as soon as a new position fix is available. When the vehicle is maneuvering sharply, all three models may give incorrect kinematic information. But, the SLPND can easily accept the user's intervention in this case.

The observation model of the Kalman filter only involves the last observation, and the covariance matrix of the observations is required. The polynomial filter uses a

series of historical observations to determine the unknown parameters, only a cofactor matrix (unscaled covariance matrix) of the observations is required.

Like the polynomial filter, the SLPND uses a sub-series of past observations to estimate the unknown parameters G and/or α, and the user has to choose the size of this sub-series. The SPLPND can also accept an unscaled covariance matrix of the observations. It is understandable that both the polynomial filter and the SLPND have a somewhat heavier computing requirements than the Kalman filter has. To meet the real-time requirements of navigation, the user may be forced to allow fewer unknown parameters and shorter observation sub-series than he/she would prefer to use.

Both "process" and observation models of the Kalman filter and of the polynomial filter are linear. Both the SLPND's motion model and observation model are nonlinear.

Both the polynomial filter and the SLPND can offer partially smoothed positions with a lag of a few steps behind the present estimation. The Kalman filter needs an independent run to compute smoothed position fixes after the sailing.

7.2 a comparison of numerical results from the SLPND and the Kalman filter

Let us apply the Kalman filter to the two sets of simulated navigation data about the two real tracks as those used in *Chapter* 6. The process model is (Gutman and

Velger, 1990):

$$\begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{pmatrix}_n = \begin{pmatrix} 1 & 0 & \Delta t & 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \frac{1}{2}\Delta t^2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{pmatrix}_{n-1} + \begin{pmatrix} \frac{1}{6}\Delta t^3 & 0 \\ 0 & \frac{1}{6}\Delta t^3 \\ \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} w_{n-1}, \quad (7.1)$$

where Δt is the sampling rate and w is a zero-mean white noise sequence with covariance matrices

$$\mathbf{q}_{ij} = E(w_i w_j) = \begin{cases} q \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix} & \text{if } i = j \\ \mathbf{0} & \text{if } i \neq j \end{cases} \quad (7.2)$$

The observation model is:

$$\begin{pmatrix} l_x \\ l_y \\ l_{\dot{x}} \\ l_{\dot{y}} \end{pmatrix}_n = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \\ v_{\dot{x}} \\ v_{\dot{y}} \end{pmatrix}_n, \quad (7.3)$$

where $\mathbf{v} = (v_x \; v_y \; v_{\dot{x}} \; v_{\dot{y}})_n^T$ is the observation error vector (zero-mean white noise vector) with a known covariance matrix $\mathbf{C}_n$. We assumed no correlation between w and $\mathbf{v}$.

We apply this Kalman filter on the above mentioned two sets of data, keeping the covariance matrices of the observations fixed and changing only the value of q, run by run from 1.0^{-15} to 4.0 in order to find its most suitable value. The RMS and the maximum position error as function of q are listed in Table 7.2 for q from 1.0^{-9} to 1.0. The most suitable value for q for Track 1 is 10^{-7}, for Track 2 is 10^{-8}. The best

Table 7.2: RMS and maximum position error in metre

Track		q									
		10^{-9}	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1.0
1	RMS	2.14	2.07	2.07	2.09	2.11	2.15	2.24	2.27	2.28	2.28
	max-e	8.07	7.93	7.78	7.96	8.13	8.60	8.66	8.58	8.56	8.56
2	RMS	3.67	3.52	3.52	3.55	3.57	3.60	3.63	3.65	3.65	3.65
	max-e	20.6	17.9	18.5	19.2	19.4	19.6	19.7	19.7	19.7	19.7

results achieved are depicted in Fig. 7.1 for the first track and Fig. 7.2 for the second.

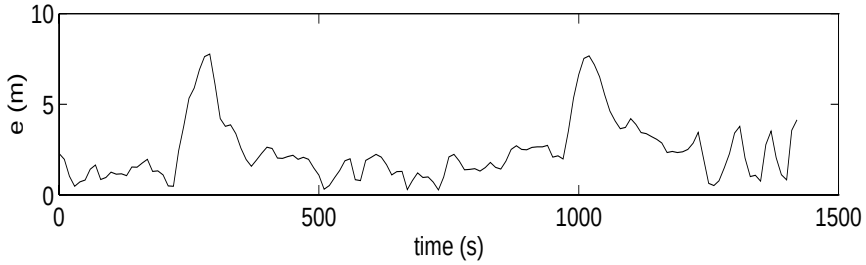

Figure 7.1: Errors of the estimated positions by Kalman filter, track 1

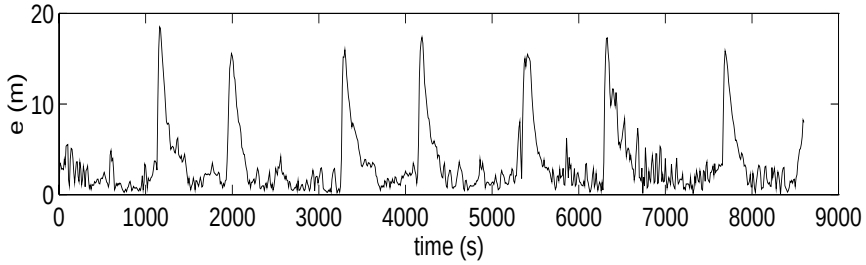

Figure 7.2: Errors of the estimated positions by Kalman filter, track 2

From Figs. 7.1 and 7.2, we can see that the Kalman filter failed to improve the position accuracy, while with the same data the SLPND was able to give much better results, the maximum estimated position error being about 2 m for Track 1 and about 2.3 m for Track 2. The error peaks experienced by the Kalman filter occur when Mary O was making sharp turns. It is not surprising because at those points the constant

acceleration model $eqn.(7.1)$ does not fit the real "process" any more, yet the filter still tries to fit the observations to this incorrect model. It is user's responsibility to give the filter the correct values of q; the technique should not be blamed for user's failure to do so. When the vehicle is under maneuvering, q may have to be changed. From Fig. 7.2, for example, we can see that when Mary O sailed straight, the Kalman filter was successful in reducing the effect of large errors caused by the turns.

Because the SLPND's motion model gets updated from fix to fix it could offer better results. This makes the SLPND run slower. Using the same data and the same computer (Sunstation with SunOS release 4.1.3.), the Kalman filter needed only 0.02 seconds of processing time for each estimated position (about Track 2), while the SLPND needed 0.11 seconds, for $k = 5$, and 0.23 seconds for $k = 10$.

Chapter 8

summary, conclusions and recommendations

This final chapter summarizes the research and recommends some points for further investigations.

8.1 summary of the research

The basic idea behind the SLPND is two-fold: (1) A sequence of observed position fixes contains true kinematic information about the vehicle, (2) A motion model based on the error statistics of the position fixes is able to convert this information into a successful navigation filter. The general strategy used in designing the SLPND was as follows.

First a position potential function was selected (*eqn.*(3.3)) and a real-time position potential field of the sequence of observed position fixes was formulated (*eqn.*(3.5)).

Second, a free particle was put into the real-time potential field, and its equation of motion was constructed (*eqn.*(3.9)). This equation was solved and *eqns.*(3.19) to (3.22) for the state vector resulted. Then, in turn, served as the basis for the derivation of the motion model (*eqn.*(3.41)).

Third, nonlinear least squares method had been formulated for the unknown parameter Θ estimation by fitting the motion model to a sequence of observations (*eqn.*(4.5)). Based on the estimated parameter vector, the estimated positions at present, in the future and the partially smoothed positions in the past can be computed.

To get the covariance matrices of the computed positions, *eqn.*(4.5) was linearized around the estimated parameter vector and the linear-least squares method was used. A blunder detector, based on the χ^2-test or Fisher-test was built within the prediction mode.

Having developed the SLPND apparatus, we made some simulations which lead to an adoption of a simplification of the coefficient matrices $\mathbf{A}$ and $\mathbf{B}$ in *eqn.*(3.9). The SLPND was then tested as an estimator, predictor, smoother, and blunder detector on two real ship tracks (*Chapter* 6). The SLPND was further reformulated to accept navigator's intervention. This feature was tested in *Chapter* 5.

Conceptual comparisons between the SLPND and two of the existing navigation filters, the polynomial filter and the Kalman filter as well as comparisons of numerical results from the SLPND and the Kalman filter were made in *Chapter* 7.

8.2 conclusions of the research

Based on the research we have done, the following conclusions can be drawn.

1. The SLPND's motion model based on the error statistics of the position fixes is able to convert the true kinematic information contained in the sequence of observed positions into a successful navigation filter. Updating the motion model position fix by position fix and leaving parameter(s) unfixed enable the motion model keep pace with the kinematic change of the vehicle to a large extent. But, compared with the Kalman filter, these cause the SLPND's heavier computational burden.

2. The simplification of the coefficient matrices $\mathbf{A}$ and $\mathbf{B}$ in *eqn.*(3.9), i.e., only

the most recent position fix needs to stay in *eqns.*(3.10) and (3.11), is valid.

3. Even without velocity information, the SLPND can learn, to a certain extent, about the kinematic of the vehicle when the vehicle is slowly and constantly maneuvering (eg circular track). But when the vehicle is heavily maneuvering (eg sinusoidal track), what it learned may be not enough to improve position accuracy. Then, to get a good navigation the SLPND needs not only position but also velocity observations.

4. The selection of k would effect the SLPND's output and there is no general k for various tracks. The best value of k depends on whether the vehicle is under maneuvering and how heavy the maneuvering is: the sharper the maneuvering the smaller the best k is.

5. A small value of α can lead to large errors in the SLPND's output while it is safe to use a large value of α, say $\alpha = 25$.

6. Different initial velocities will cause little difference in the results if α is large and if the covariance matrix of the initial velocity has well described the uncertainty in the initial velocity.

7. The interventions from navigators can be easily realized. When the vehicle is under a large maneuvering, giving the SLPND extra information about the kinematic change, even a guess of the vehicle's velocity or simply uncertainty information on its computed velocity at the change point, is very helpful.

8. The SLPND can act as an integrator to combine position and velocity observations, and then can supply navigation better than those using either positions alone or velocities alone (dead-reckoning). It is supposed to be able to integrate range or azimuth observations too, then the range or azimuth will be regarded as a special error ellipse (ellipsoid).

9. The SLPND can suppress random errors and high frequency systematic errors in observations and can act as an estimator, predictor, smoother and blunder detector. The partially smoothed positions can be obtained with little extra effort.

10. The SLPND requires the uncertainties of the position fixes to be known only

relatively (cofactor matrix), and the covariance matrices of the computed positions can still be achieved.

11. To save processing time, we need to give the SLPND a well defined searching area for the parameters. This requires some experience.

12. Compared with the Kalman filter, the SLPND has a slower processing speed.

After all, the basic idea behind this research works. We have reached the purpose of the research.

8.3 recommendations

To create a completely new and useful navigation filter is a very complex job. The SLPND is still at its primitive stage, everything in it may not be perfect and many further investigations are needed. The following subjects are recommended for the further investigations.

1. To get more feeling about this new filter, more simulations and applications (using both real observed position and velocity data) are needed.

2. The software should be improved to improve the processing speed and use less memory space.

3. To get the best position potential function (force model), other position potential functions should be tested and evaluated, for example, the potential function which produce force proportional to square of the distance from the source.

4. The effort to develop nonlinear optimization methods which can offer error estimation should be made. Gaussian method in the family of the nonlinear optimization methods is strongly relative to the linear least squares method. It is possible to be adapted to include the error estimation.

5. To extend this filter to three dimensional navigation problems is also recommended.

References

Abidin, H.A (1993). *Computational and geometrical aspects of on-the fly ambiguity resolution.* Ph.D. dissertation, Dept. of Surveying Engineering, Technical Report No.164, University of New Brunswick, Fredericton, New Brunswick, Canada, 314 pp.

Ackroyd, N. and Lorimer, R. (1990). *Global navigation-A GPS user's guide.* Lloyd's of London press Ltd.

Ashtech (1993). *Ashtec Precise Differential GPS Navigation and Surveying Software User's Manual.* Document Number 600200, Rev A Pre-release version 3, Publication date 3 May, 1993.

Anderson, E.W. (1966). *The Principles of Navigation.* Hollins and Carter, London.

Bachri S. (1990). *Assessments of the polarfix system.* Thesis for Master of Engineering in the Dept. of Surveying Engineering, The University of New Brunswick, Canada.

Bard, Y. (1974). *Nonlinear Parameter Estimation.* Academic Press, New York and London.

Barton, D. (1980). "On Taylor series and stiff equations." *ACM Trans. Math. Software.* 6, 3, 280-294.

Barton, D., Willers, I.M., and Zahar, R.V.M. (1971). "Taylor series methods for ordinary differential equations - An evaluation." *Mathematical Software.* John Rice (Ed.), Academic Press, New York, pp. 369-390.

Blais,J.A.R. (1982). "Synthesis of Kriging estimation methods." *Manuscripta Geodaetica*, Vol.7, pp.325-352.

Brown, R.G. and Hwang, P.Y.C. (1992). *Introduction to random signals and applied Kalman filtering.* second edition, John Wiley and Sons, Inc.

Cannon M.E (1990). "High accuracy GPS semi-kinematic positioning: modeling and results." *Navigation*, Vol.37, No.1.

Chang, Y.F. (1986). "The Atomic Toolbox." *Byte*, Vol. 11, No.4.

Corliss, G. and Chang, Y.F. (1982). "Solving Ordinary Differential Equations Using Taylor Series." *ACM Transactions on Mathematical Software*, Vol. 8, No. 2: 114-144.

Deloach, S.R., Wells, D. and Dodd, D. (1995). "Why On-the-Fly?" *GPS world*, May, 1995.

Dodd, D.W. (1992). "HDCS Navigation Filter Program Algorithm Description." paper for SE 6101, Department of Surveying Engineering, University of New Brunswick, Fredericton, Canada E3B 5A3.

Dove, M.J. and Miller, K.M. (1989). "Kalman filters in navigation systems.' *Navigation (UK)*, Vol. 42, No.2.

Enge, P.K. and McCullough, J.R. (1988). "Aiding GPS with calibrated Loran-C." *Navigation (US)*, Vol. 35.

Gao, Y., Krakiwsky E.J. and Liu Z.W. (1992). "A new algorithm for filtering a correlated measurement sequence." *manuscripta geodaetica*, Vol.17: 96-103.

Gelb, A. (1974). *Applied optimal estimation.* MIT-Press, Cambridge, Ma.

Grant, S.T. (1976). *Integration of passive ranging Loran-C satellite navigation, ship's log, and ship's Gyrocompass.* M. S. E. thesis, Dept. of Surveying Engineering, University of New Brunswick, Fredericton, N.B. Canada.

Gutman, P-O. and Velger, M. (1990). "Tracking Targets Using Adaptive Kalman Filtering." *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 26, No. 5, pp 691-698.

Hofmann-Wellenhof, B., Lichtenegger, H. and Collins, J. (1992). *Global Positioning System.* Springer-Verlag Wien New York.

Hwang, P.Y.C, Brown, R.G. (1990). "GPS navigation: Combining pseudo-range with continuous carrier phase using a Kalman filter." *Navigation*, Vol.37, No.2.

Inzinga, T. and Vaníček, P. (1985). "A two-dimensional navigation algorithm using a probabilistic force field.", Presented at Third International Symposium on Inertial Technology for Surveying and Geodesy, Banff, Canada, 1985.

Kuebler, W. and Sommers, S. (1982). "A Critical Review of the Fix Accuracy and Reliability of Electronic Marine Navigation Systems." *Navigation*, Vol. 29, No.2.

Liu, Z.W, Krakiwsky E.J, and Gao Y. (1992). "An analysis of three methods for filtering a correlated measurement sequence." *manuscripta geodaetica*, 17: 87-95.

Lu, G. and Lachapelle G. (1992). "Statistical quality control for kinematic GPS positioning." *manuscripta geodaetica*, Vol.17: 270-281.

McGraw-Hill (1978). *Dictionary of Scientific and Technical Terms.* Second Edition, McGraw-Hill, 1978.

Mertikas, S. (1985). *Error distributions and accuracy measures in Navigation: an overview.* Technical report No. 113, Dept of Surveying Engineering, University of New Brunswick, Fredericton, Canada.

Mikhail, E.M. (1976). *Observations and least squares.* University Press of America.

Minkler, G. and Minkler, J. (1993). *Theory and Application of Kalman Filtering.* Magellan Book Company, Palm Bay, FL.

Morrison, N. (1969). *Introduction to Sequential Smoothing and Prediction.* McGRAW-HILL BOOK COMPANY.

Napier, M. (1990). "Integration of satellite and inertial positioning systems." *Navigation (UK)*, Vol.43, No. 1.

Pierre, D.A. (1969). *Optimization Theory with Applications.* John Wiley and Sons, Inc.

Reklaitis, G.V. and Ravindran, A. and Ragsdell, K.M. (1983). *Engineering Optimization: Methods and Applications.* A Wiley-Interscience Publication John Wiley and Sons.

Rektorys, K. (editor) (1969). *Survey of Applicable Mathematics.* The M. I. T. Press.

Salzmann M.A. (1988). *Some aspects of Kalman filtering.* Technical report No. 140, Dept. of Surveying Engineering, University of New Brunswick, Fredericton, Canada.

Schwarz, K.P., Cannon, M.E., and Wong, R.V.C., (1989). "A comparison of GPS kinematic models for the determination of position and velocity along a trajectory." *manuscripta geodaetica*, Vol.14: 345-353.

Sonnenberg, G.J. (1988). *Radar and Electronic navigation.* Butterworth & Co. Ltd.

Stich, H., Blanchard, W., Lechner, W., Kayser, D., Gaillard, H. (1994). "Assessment of the Potential of Satellite Navigation Systems for Europe: Result of a Study for the European Commission." *Navigation (UK)*, Vol.47, No.3, Sept. 1994.

Tetley, L. and Calcutl, D. (1986). *Electronic aids to navigation.* Edward Arnold Ltd.

Teunissen, P.J.G. (1990). "Nonlinear Least Squares." *manuscripta geodaetica*, 15: 137-150.

Torn, A. and Zilinskas A. (1989). *Global Optimization.* Springer-Verlag Berlin Heidelberg.

Vaníček, P. and Krakiwsky, E.J. (1986). *Geodesy: the concepts.* second edition, Elsevier Science Publishers B. V. 1992.

Wells, D.E. (1976). "Improved marine navigation error modelling through system integration." Presented at Canadian Geophysical Union Symposium on Modern Trends in Geodesy, Quebec City, 16 June, 1976.

Wells, D.E. and Grant, S.T. (1981). "An adaptable integrated navigation system: BIONAV." Presented at Canadian Petroleum Association Colloquium III.

Wong R.V.C (1982). *A Kalman filter-smoother for an inertial navigation system of local level type.* UCSE Report No. 20001, The University of Calgary.

Wong R.V.C, Schwarz K.P (1983). *Dynamic positioning with an integrated GPS-INS: Formulae and baseline tests.* UCSE Report No. 30003, The University of Calgary.

Wong R.V.C, Schwarz K.P, Cannon M.E (1988). "High accuracy kinematic positioning by GPS-INS." *Navigation*, Vol. 35, No.2.

Appendix I

solutions for the motion of the particle by taylor series

I.1 motion of the particle

We have no intention to explain in detail the background theory of this subject here, and also do not claim the completeness of this study. An interested reader can find them in Barton, Willers and Zahar [1971], Barton [1980] and Corliss and Chang [1982].

Taylor series in the neighborhood of $t_0 = 0$, i.e., Maclaurin series, has been defined as,

$$\mathbf{r}(t) = \mathbf{r}_0 + \sum_{n=1}^{\infty} \frac{\mathbf{r}_0^{(n)}}{n!} t^n \tag{I.1}$$

Taking the time derivative of $eqn.(3.9)$ we get:

$$\dot{\mathbf{r}}(t) = \dot{\mathbf{r}}_0 + \sum_{n=1}^{\infty} \frac{\mathbf{r}_0^{(n+1)}}{n!} t^n. \tag{I.2}$$

Before we generate a Taylor series for a function, we have to determine how many terms are suitable for the function and the procedure we will use. Using a Taylor series for a solution of an ordinary differential equation makes us face a possible

singularity problem. A singularity is the point where the Taylor series will behave dramatically. If a generated Taylor series had singularities around the solution points, the Taylor series would fail to give a correct results. In other words, the series would diverge. One principle of selecting the number of terms is that the length of the Taylor series should be enough for the study of the convergence property of the series. The convergence property of a short Taylor series cannot be determined by any means. There is insufficient information in the few terms of a short Taylor series for a reliable analysis of the truncation error. Hence, the short series methods such as Runge-Kutta are forced into taking small incremental steps. The many small steps give rise to the propagation of machine round-off errors. Thus, for a difficult problem such as those associated with exponential terms, the short-series methods are bound to produce results with accuracy much poorer than the machine would allow. While small steps give rise to round-off errors, large steps give rise to truncation errors. When the order of the Taylor series is high, say about 30, it is possible to find the exact positions of those catastrophic singularities in the solution function [Chang, 1986]. In our research we take the order of the Taylor series equal to 30.

To establish a Taylor series for a function, the derivatives of the function with respect to associated variables, time variable in our case, are required up to desired orders. There is no common formula to produce the derivatives for various functions. The basic means to calculate all required derivatives is Leibnitz's rule. Leibnitz's rule can be written as: if $u(t) = f(t)g(t)$, $u'(t) = f'(t)g(t) + f(t)g'(t)$ [Rektorys, 1969].

If we would simply use this rule, the computing burden would be very heavy. Fortunately, we have found a special recurrent formula for the calculation. By taking successive derivatives of $eqn.(3.9)$, we get a sequence of formulas as follows:

$$\begin{aligned}
\mathbf{r}^{(3)} &= -\alpha\mathbf{r}^{(2)} - e^{-\alpha t}\mathbf{A}\mathbf{r}^{(1)} \\
\mathbf{r}^{(4)} &= -\alpha\mathbf{r}^{(3)} - e^{-\alpha t}\mathbf{A}(\mathbf{r}^{(2)} - \alpha\mathbf{r}^{(1)}) \\
\mathbf{r}^{(5)} &= -\alpha\mathbf{r}^{(4)} - e^{-\alpha t}\mathbf{A}(\mathbf{r}^{(3)} - 2\alpha\mathbf{r}^{(2)} + \alpha^2\mathbf{r}^{(1)})
\end{aligned}$$

$$
\begin{aligned}
\mathbf{r}^{(6)} &= -\alpha\mathbf{r}^{(5)} - e^{-\alpha t}\mathbf{A}(\mathbf{r}^{(4)} - 3\alpha\mathbf{r}^{(3)} + 3\alpha^2\mathbf{r}^{(2)} - \alpha^3\mathbf{r}^{(1)}) \\
\mathbf{r}^{(7)} &= -\alpha\mathbf{r}^{(6)} - e^{-\alpha t}\mathbf{A}(\mathbf{r}^{(5)} - 4\alpha\mathbf{r}^{(4)} + 6\alpha^2\mathbf{r}^{(3)} - 4\alpha^3\mathbf{r}^{(2)} + \alpha^4\mathbf{r}^{(1)}) \\
\mathbf{r}^{(8)} &= -\alpha\mathbf{r}^{(7)} - e^{-\alpha t}\mathbf{A}(\mathbf{r}^{(6)} - 5\alpha\mathbf{r}^{(5)} + 10\alpha^2\mathbf{r}^{(4)} - 10\alpha^3\mathbf{r}^{(3)} + 5\alpha^4\mathbf{r}^{(2)} - \alpha^5\mathbf{r}^{(1)}) \\
&\cdots\cdots,
\end{aligned}
$$

where, $\mathbf{r}^{(i)}$ represents the i-th derivative of $\mathbf{r}$ with respect to t. The list of the coefficients in the second and following terms on the right hand side in the above formulas form a well known Pascal triangle [Rektorys, 1969], i.e.,

$$
\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1 \\
 & & & & & \cdots\cdots & & & & &
\end{array}
$$

The coefficients in the second terms are equal to the combination C_{n-3}^{k-1}, where n is the order of the derivative and k is the position of the coefficient in the bracket. So, the n-th order $(n > 2)$ derivative of $\mathbf{r}$ with respect to time t can be expressed as

$$
\mathbf{r}^{(n)} = -\alpha\mathbf{r}^{(n-1)} - e^{-\alpha t}\mathbf{A}\sum_{i=1}^{n-2}(-1)^{i-1}C_{n-3}^{i-1}\alpha^{i-1}\mathbf{r}^{(n-i-1)}. \tag{I.3}
$$

By using $(I.3)$, we can get the desired derivatives up to any order.

Our next task is to estimate the series radius of convergence, the location and order of the primary singularity.

It has been said that series for solutions to ordinary differential equations follow a few very definite patterns which are characterized by the locations of primary singularities, and series which are real-valued on the real axis can have poles, logarithmic

branch points, and essential singularities only on the real axis or in conjugate pairs, further, the effects of all secondary singularities disappear if sufficiently long series are used (Corliss, Chang, 1982). Based on the above arguments, Corliss and Chang have recommended the following ways to estimate the series radius of convergence and the order of the singularity.

(1) If the series have a single primary singularity, use

$$\frac{h}{R_c} = \frac{h}{a} = k\frac{V(k+1)}{V(k)} - (k-1)\frac{V(k)}{V(k-1)} \tag{I.4}$$

$$order = s = k\frac{V(k+1)}{V(k)}(\frac{R_c}{h}) - k + 1 \quad k = n-1, \tag{I.5}$$

where, $R_c = a$ is the radius of convergence, $V(i)$ is reduced derivatives defined as $V(i) = \mathbf{r}^{(i-1)}(t_0)h^{i-1}/(i-1)!$, $h = t - t_0$ is the stepsize of the Taylor series. In order to detect when the series has singularities which are not of this form, two estimates for h/R_c using different terms of the series are need. If the two estimates do not agree, then the series does not have one real primary singularity, so the presence of a conjugate pair of primary singularities is investigated.

(2) If it has only a conjugate pair of primary singularities, use

$$\begin{aligned} kV(k+1) &= (k-1)V(k)x_1 + V(k)x_2 - (k-2)V(k-1)x_3 \\ &\quad -2V(k-1)x_4 \end{aligned} \tag{I.6}$$

where, $x_1 = (h/a)\cos\theta$, $x_2 = s(h/a)\cos\theta$, $x_3 = (h/a)^2$, and $x_4 = s(h/a)^2$, $k = n-1, \ldots, n-4$ and n is the length of the series being analyzed. Then a measure of the relative accuracy of the solution is obtained from the residual of another copy of $eqn.(I.6)$ with $k = n-5$. If the residual is small, then the series radius of convergence, as well as the order and location of the conjugate pair of singularities $ae^{\pm i\theta}$, is computed from $x_1, x_2, x_3, \text{and} x_4$. If the residual is large, if $x_3 < 0$, or if computed $|\cos\theta| > 1$, then this means that the series has secondary singularities. Then,

(3) Use the top-line heuristic. This procedure gives a conservative estimate for R_c from the slope of a linear upper envelope (a straight line fitting the points from above) of a graph of $ln|V(k)|$ versus k. the slope approaches $ln|R_c|$ as $k \to \infty$. It has been said that the upper envelope of the graph of $ln|V(k)|$ versus k has the following properties.

(1) If the order of the primary singularity is 1, then the slope is $ln|h/R_c|$.

(2) If the order is not 1, then the slope converges to $ln|h/R_c|$ as $k \to \infty$ at a rate roughly proportional to $1/k$.

(3) If the order is not 1, then the upper envelope is not linear. For orders larger than 1, the graphs open downward. The concavity approaches zero very rapidly as $k \to \infty$. For orders less than 1, the graphs are concave up, the slope underestimates $ln|h/R_c|$, and R_c is overestimated. In case of this, the series is differentiated termwise to reduce the second derivative of the graph, and a new top-line is fit. This process is repeated until the graph is linear, opens downward, or until seven termwise differentiations have been done. If the series for y''' still opens upward, the estimate for R_c is reduced by 10 percent [Corliss and Chang, 1982].

Having completed the estimation of the convergence radius of the series, the machine (computer) computes the largest integration stepsize which may be taken subject to error control constraints. The truncation error is estimated from the magnitude of the first neglected term in the Taylor series. By adjusting the integration stepsize, the truncation error can be kept below a specified limit.

Because of the stepsize control in order to avoid the singularities and to realize the error control, the solution at the desired point (time instant) may not be got in one step. Then the immediately preceding solution is used as new initial condition and the above described procedure will be repeated until the solution at the desired solution point is obtained.

I.2 Evaluation of gradients

By derivating *eqn.*$(I.1)$ and *eqn.*$(I.2)$ with respect to $\mathbf{\Theta}$ respectively, we get

$$\frac{\partial \mathbf{r}(t)}{\partial \mathbf{\Theta}} = \frac{\partial \mathbf{r}_0}{\partial \mathbf{\Theta}} + \sum_{n=1}^{\infty} \frac{\partial \mathbf{r}_0^{(n)}}{\partial \mathbf{\Theta}} \frac{t^n}{n!} \tag{I.7}$$

$$\frac{\partial \dot{\mathbf{r}}(t)}{\partial \mathbf{\Theta}} = \frac{\partial \dot{\mathbf{r}}_0}{\partial \mathbf{\Theta}} + \sum_{n=1}^{\infty} \frac{\partial \mathbf{r}_0^{(n+1)}}{\partial \mathbf{\Theta}} \frac{t^n}{n!}. \tag{I.8}$$

$\partial \mathbf{r}_0 / \partial \mathbf{\Theta}$ and $\partial \dot{\mathbf{r}}_0 / \partial \mathbf{\Theta}$ in above formula can be obtained from the last evolution. Derivating the equation of motion of the particle, i.e., *eqn.*(3.9) with respect to G and α respectively and setting $t = 0$, we can get

$$\frac{\partial \mathbf{r}_0^{(2)}}{\partial G} = -(\mathbf{A}\mathbf{r}_0 - \mathbf{B})/G - \mathbf{A}\frac{\partial \mathbf{r}_0}{\partial G} \tag{I.9}$$

$$\frac{\partial \mathbf{r}_0^{(2)}}{\partial \alpha} = -(\frac{\partial \mathbf{A}}{\partial \alpha}\mathbf{r}_0 + \mathbf{A}\frac{\partial \mathbf{r}_0}{\partial \alpha} - \frac{\partial \mathbf{B}}{\partial \alpha}) \tag{I.10}$$

where

$$\frac{\partial \mathbf{A}}{\partial \alpha} = G\sum_{i=1}^{n} e^{\alpha(t_i - t_n)}(t_i - t_n)\mathbf{C}_i^{-1}$$

$$\frac{\partial \mathbf{B}}{\partial \alpha} = G\sum_{i=1}^{n} e^{\alpha(t_i - t_n)}(t_i - t_n)\mathbf{C}_i^{-1}\mathbf{r}_i^0.$$

To get higher order of partial derivatives $(i \geq 3)$, we derivate *eqn.*$(I.3)$, with $t = 0$, with respect to G and α respectively and get

$$\begin{aligned}\frac{\partial \mathbf{r}_0^{(i)}}{\partial G} &= -\alpha\frac{\partial \mathbf{r}_0^{(i-1)}}{\partial G} - \mathbf{A}\sum_{k=1}^{i-2}(-1)^{k-1}C_{i-3}^{k-1}\alpha^{k-1}\frac{\partial \mathbf{r}_0^{(i-k-1)}}{\partial G} \\ &\quad -\frac{\mathbf{A}}{G}\sum_{k=1}^{i-2}(-1)^{k-1}C_{i-3}^{k-1}\alpha^{k-1}\mathbf{r}_0^{(i-k-1)} \end{aligned} \tag{I.11}$$

$$\begin{aligned}\frac{\partial \mathbf{r}_0^{(i)}}{\partial \alpha} &= -\mathbf{r}_0^{(i-1)} - \alpha\frac{\partial \mathbf{r}_0^{(i-1)}}{\partial \alpha} - \frac{\partial \mathbf{A}}{\partial \alpha}\sum_{k=1}^{i-2}(-1)^{k-1}C_{i-3}^{k-1}\alpha^{k-1}\mathbf{r}_0^{(i-k-1)} \\ &\quad -\mathbf{A}\sum_{k=1}^{i-2}(-1)^{k-1}C_{i-3}^{k-1}\alpha^{k-2}((k-1)\mathbf{r}_0^{(i-k-1)} + \alpha\frac{\partial \mathbf{r}_0^{(i-k-1)}}{\partial \alpha}. \end{aligned} \tag{I.12}$$

Appendix II

orthogonal coordinate transformation

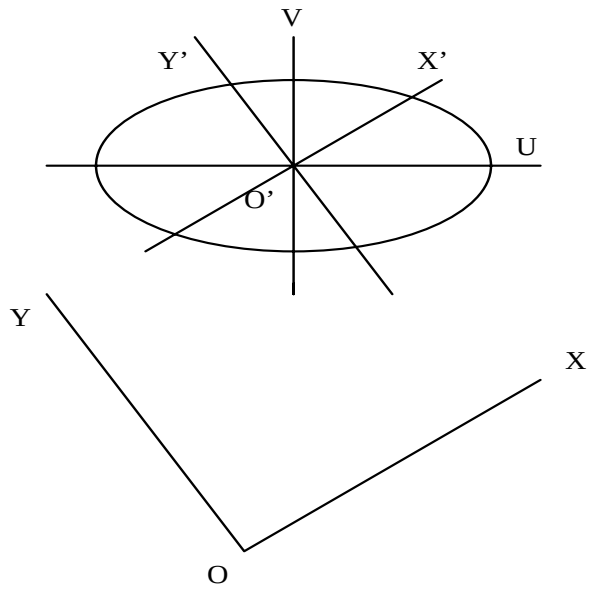

Figure II.1: Orthogonal coordinate transformation

We assume that an error ellipse in coordinate system XOY is defined as follows,

$$(\mathbf{r} - \mathbf{r}^0)^T \mathbf{C}^{-1} (\mathbf{r} - \mathbf{r}^0) = h, \tag{II.1}$$

where h is a constant,

$$\mathbf{r} = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad \mathbf{r_0} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix},$$

and the covariance matrix $\mathbf{C}$ is given as:

$$\mathbf{C}_i = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}.$$

First, we shift the origin of the coordinate system XOY to $\mathbf{r}^0$ by transformation: $\mathbf{r}' = \mathbf{r} - \mathbf{r}^0$. In the new coordinate system $X'O'Y'$, the equation of the error ellipse is:

$$\mathbf{r}'^T \mathbf{C}^{-1} \mathbf{r}' = h. \tag{II.2}$$

Now we rotate $X'O'Y'$ by an angle θ and get a new coordinate system $UO'V$. The corresponding coordinate transformation is then defined by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} \tag{II.3}$$

If the rotation angle θ is given as:

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2}\right) \qquad -\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}, \tag{II.4}$$

the equation of the error ellipse becomes [Mertikas, 1985]

$$\frac{u^2}{\lambda_1} + \frac{v^2}{\lambda_2} = h, \tag{II.5}$$

where

$$\lambda_1 = \frac{1}{2}[\sigma_x^2 + \sigma_y^2 + \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4{\sigma_{xy}}^2}] \tag{II.6}$$

$$\lambda_2 = \frac{1}{2}[\sigma_x^2 + \sigma_y^2 - \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4{\sigma_{xy}}^2}]. \tag{II.7}$$

Appendix III

derivatives of f_i with respect to parameter vector Θ

Let us take the derivatives of *eqns.*(3.19), (3.20), (3.21) and (3.22) with respect to G and α. Because the constants c_1, c_2, c_1' and c_2' are determined from the initial state of the motion at last time instant by *eqns.*$(3.23)-(3.26)$, c_1, c_2, c_1' and c_2' are functions of the parameters G and α, with an exception that the time instant of the last position fix is t_{n-k-1}, when the initial state of the motion is inherited from the last fitting process . Noting that

$$\frac{dJ_0(z)}{dz} = -J_1(z), \quad \frac{dN_0(z)}{dz} = -N_1(z)$$
$$\frac{dJ_1(z)}{dz} = J_0(z) - \frac{J_1(z)}{z}, \quad \frac{dN_1(z)}{dz} = N_0(z) - \frac{N_1(z)}{z}$$

we get the components of $\frac{\partial \mathbf{S}}{\partial \mathbf{\Theta}}$ as follows.

$$\frac{\partial x}{\partial G} = \frac{\partial c_1}{\partial G}J_0(s_x) - (c_1 J_1(s_x) + c_2 N_1(s_x))\frac{\partial s_x}{\partial G} + \frac{\partial c_2}{\partial G}N_0(s_x) \qquad \text{(III.1)}$$

$$\frac{\partial y}{\partial G} = \frac{\partial c_1'}{\partial G}J_0(s_y) - (c_1' J_1(s_y) + c_2' N_1(s_y))\frac{\partial s_y}{\partial G} + \frac{\partial c_2'}{\partial G}N_0(s_y) \qquad \text{(III.2)}$$

$$\frac{\partial x}{\partial \alpha} = \frac{\partial (B_x/A_x)}{\partial \alpha} + \frac{\partial c_1}{\partial \alpha}J_0(s_x) - (c_1 J_1(s_x) + c_2 N_1(s_x))\frac{\partial s_x}{\partial \alpha} + \frac{\partial c_2}{\partial \alpha}N_0(s_x) \qquad \text{(III.3)}$$

$$
\begin{aligned}
\frac{\partial y}{\partial \alpha} &= \frac{\partial (B_y/A_y)}{\partial \alpha} + \frac{\partial c_1'}{\partial \alpha} J_0(s_y) - (c_1' J_1(s_y) + c_2' N_1(s_y)) \frac{\partial s_y}{\partial \alpha} \\
&\quad + \frac{\partial c_2'}{\partial \alpha} N_0(s_y) \qquad \text{(III.4)} \\
\frac{\partial \dot{x}}{\partial G} &= \frac{\dot{x}}{2G} + \frac{\alpha s_x}{2} \{ \frac{\partial c_1}{\partial G} J_1(s_x) + [c_1 (J_0(s_x) - \frac{J_1(s_x)}{s_x}) \\
&\quad + c_2 (N_0(s_x) - \frac{N_1(s_x)}{s_x})] \frac{\partial s_x}{\partial G} + \frac{\partial c_2}{\partial G} N_1(s_x) \} \qquad \text{(III.5)} \\
\frac{\partial \dot{y}}{\partial G} &= \frac{\dot{y}}{2G} + \frac{\alpha s_y}{2} \{ \frac{\partial c_1'}{\partial G} J_1(s_y) + [c_1' (J_0(s_y) - \frac{J_1(s_y)}{s_y}) \\
&\quad + c_2' (N_0(s_y) - \frac{N_1(s_y)}{s_y})] \frac{\partial s_y}{\partial G} + \frac{\partial c_2'}{\partial G} N_1(s_y) \} \qquad \text{(III.6)} \\
\frac{\partial \dot{x}}{\partial \alpha} &= \frac{1}{2} (\frac{T_{xt}}{A_x} - t) \dot{x} + \frac{\alpha s_x}{2} \{ \frac{\partial c_1}{\partial \alpha} + [c_1 (J_0(s_x) - \frac{J_1(s_x)}{s_x}) \\
&\quad + c_2 (N_0(s_x) - \frac{N_1(s_x)}{s_x})] \frac{\partial s_x}{\partial \alpha} + \frac{\partial c_2}{\partial \alpha} N_1(s_x) \} \qquad \text{(III.7)} \\
\frac{\partial \dot{y}}{\partial \alpha} &= \frac{1}{2} (\frac{T_{yt}}{A_y} - t) \dot{y} + \frac{\alpha s_y}{2} \{ \frac{\partial c_1'}{\partial \alpha} + [c_1' (J_0(s_y) - \frac{J_1(s_y)}{s_y}) \\
&\quad + c_2' (N_0(s_y) - \frac{N_1(s_y)}{s_y})] \frac{\partial s_y}{\partial \alpha} + \frac{\partial c_2'}{\partial \alpha} N_1(s_y) \} \qquad \text{(III.8)}
\end{aligned}
$$

where

$$
\begin{aligned}
s_x &= \frac{2}{\alpha} e^{-\frac{\alpha t}{2}} \sqrt{G A_x} \\
s_y &= \frac{2}{\alpha} e^{-\frac{\alpha t}{2}} \sqrt{G A_y} \\
\frac{\partial s_x}{\partial G} &= \frac{s_x}{2G} \\
\frac{\partial s_y}{\partial G} &= \frac{s_y}{2G} \\
\frac{\partial s_x}{\partial \alpha} &= -s_x (\frac{1}{\alpha} + \frac{t}{2} - \frac{T_{xt}}{2A_x}) \\
\frac{\partial s_y}{\partial \alpha} &= -s_y (\frac{1}{\alpha} + \frac{t}{2} - \frac{T_{yt}}{2A_y}) \\
\frac{\partial (B_x/A_x)}{\partial \alpha} &= \frac{T_{bxt} A_x - B_x T_{xt}}{A_x^2} \\
\frac{\partial (B_y/A_y)}{\partial \alpha} &= \frac{T_{byt} A_y - B_y T_{yt}}{A_y^2} \\
T_{xt} &= \sum_{i=1}^{n} e^{\alpha (t_i - t_n)} (t_i - t_n) p_{x_i}
\end{aligned}
$$

$$
\begin{aligned}
T_{yt} &= \sum_{i=1}^{n} e^{\alpha(t_i - t_n)}(t_i - t_n)p_{y_i} \\
T_{bxt} &= \sum_{i=1}^{n} e^{\alpha(t_i - t_n)}(t_i - t_n)p_{x_i}x_i^0 \\
T_{byt} &= \sum_{i=1}^{n} e^{\alpha(t_i - t_n)}(t_i - t_n)p_{y_i}y_i^0.
\end{aligned}
$$

The derivatives of c_1, c_2, c_1' and c_2' with respect to G and α can be obtained by taking derivatives of $eqns.(3.23) - (3.26)$.

$$
\begin{aligned}
\frac{\partial c_1}{\partial G} &= -\frac{\pi}{\alpha}\{\dot{x}_0 N_1(s_{x0})m_x\alpha^{-1} + i_x[\frac{m_x}{2}N_1(s_{x0}) \\
&\quad +(N_0(s_{x0}) - \frac{1}{s_{x0}}N_1(s_{x0}))\frac{T_x}{\alpha}]\} \qquad \text{(III.9)} \\
\frac{\partial c_2}{\partial G} &= \frac{\pi}{\alpha}\{\dot{x}_0 J_1(s_{x0})m_x\alpha^{-1} + i_x[\frac{m_x}{2}J_1(s_{x0}) \\
&\quad +(J_0(s_{x0}) - \frac{1}{s_{x0}}J_1(s_{x0}))\frac{T_x}{\alpha}]\} \qquad \text{(III.10)} \\
\frac{\partial c_1'}{\partial G} &= -\frac{\pi}{\alpha}\{\dot{y}_0 N_1(s_{y0})m_y\alpha^{-1} + i_y[\frac{m_y}{2}N_1(s_{y0}) \\
&\quad +(N_0(s_{y0}) - \frac{1}{s_{y0}}N_1(s_{y0}))\frac{T_y}{\alpha}]\} \qquad \text{(III.11)} \\
\frac{\partial c_2'}{\partial G} &= \frac{\pi}{\alpha}\{\dot{y}_0 J_1(s_{y0})m_y\alpha^{-1} + i_y[\frac{m_y}{2}J_1(s_{y0}) \\
&\quad +(J_0(s_{y0}) - \frac{1}{s_{y0}}J_1(s_{y0}))\frac{T_y}{\alpha}]\} \qquad \text{(III.12)} \\
\frac{\partial c_1}{\partial \alpha} &= -\frac{c_1}{\alpha} - \frac{\pi}{\alpha}\{-\frac{\partial(B_x/A_x)}{\partial\alpha}\sqrt{GA_x}N_1(s_{x0}) + i_x \\
&\quad [\frac{T_{xt}}{2m_x}N_1(s_{x0}) + \sqrt{GA_x}(N_0(s_{x0}) - \\
&\quad \frac{1}{s_{x0}}N_1(s_{x0}))\frac{\partial s_{x0}}{\partial\alpha}] + \dot{x}_0 N_1(s_{x0})\frac{\partial s_{x0}}{\partial\alpha}\} \qquad \text{(III.13)} \\
\frac{\partial c_2}{\partial \alpha} &= \frac{c_2}{\alpha} + \frac{\pi}{\alpha}\{-\frac{\partial(B_x/A_x)}{\partial\alpha}\sqrt{GA_x}J_1(s_{x0}) + i_x \\
&\quad [\frac{T_{xt}}{2m_x}J_1(s_{x0}) + \sqrt{GA_x}(J_0(s_{x0}) - \\
&\quad \frac{1}{s_{x0}}J_1(s_{x0}))\frac{\partial s_{x0}}{\partial\alpha}] + \dot{x}_0 J_1(s_{x0})\frac{\partial s_{x0}}{\partial\alpha}\} \qquad \text{(III.14)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial c_1'}{\partial \alpha} = & -\frac{c_1'}{\alpha} - \frac{\pi}{\alpha}\{-\frac{\partial (B_y/A_y)}{\partial \alpha}\sqrt{GA_y}N_1(s_{y0}) + i_y \\
& [\frac{T_{yt}}{2m_y}N_1(s_{y0}) + \sqrt{GA_y}(N_0(s_{y0}) - \\
& \frac{1}{s_{y0}}N_1(s_{y0}))\frac{\partial s_{y0}}{\partial \alpha}] + \dot{y}_0 N_1(s_{y0})\frac{\partial s_{y0}}{\partial \alpha}\} \qquad \text{(III.15)} \\
\frac{\partial c_2'}{\partial \alpha} = & \frac{c_2'}{\alpha} + \frac{\pi}{\alpha}\{-\frac{\partial (B_y/A_y)}{\partial \alpha}\sqrt{GA_y}J_1(s_{y0}) + i_y \\
& [\frac{T_{yt}}{2m_y}J_1(s_{y0}) + \sqrt{GA_y}(J_0(s_{y0}) - \\
& \frac{1}{s_{y0}}J_1(s_{y0}))\frac{\partial s_{y0}}{\partial \alpha}] + \dot{y}_0 J_1(s_{y0})\frac{\partial s_{y0}}{\partial \alpha}\} \qquad \text{(III.16)}
\end{aligned}
$$

where

$$
\begin{aligned}
m_x &= \sqrt{\frac{A_x}{G}} \\
m_y &= \sqrt{\frac{A_y}{G}} \\
i_x &= x_0 - B_x/A_x \\
i_y &= y_0 - B_y/A_y \\
s_{x0} &= \frac{2}{\alpha}\sqrt{GA_x} \\
s_{y0} &= \frac{2}{\alpha}\sqrt{GA_y} \\
\frac{\partial s_{x0}}{\partial \alpha} &= \frac{1}{\alpha}(\frac{T_{xt}}{m_x} - s_{x0}) \\
\frac{\partial s_{y0}}{\partial \alpha} &= \frac{1}{\alpha}(\frac{T_{yt}}{m_y} - s_{y0}).
\end{aligned}
$$