

**ON THE APPLICATION OF DATA ASSIMILATION  
IN THE SINGAPORE REGIONAL MODEL**

**SUN YABIN**

*(M.Sc., TJU)*

**A THESIS SUBMITTED  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY  
DEPARTMENT OF CIVIL ENGINEERING  
NATIONAL UNIVERSITY OF SINGAPORE**

**2010**

## **Acknowledgements**

I would like to express my sincere gratitude to my supervisor, Professor Chan Eng Soon, for his continuous support on my research. His immense knowledge and constructive criticisms have been of great value for this study. Without his guidance, this work would not have been possible.

I am deeply grateful to my co-supervisor, Assoc. Professor Vladan Babovic, who guided me throughout this research, and gave me the opportunity to work with other researchers in Singapore-Delft Water Alliance. His rigorous attitude and eternal enthusiasm in research have exerted a remarkable influence on me, and will accompany me in my entire career.

My sincere thanks also go to Professor Liong Shie-Yui, Professor Ong Say Leong, Professor Cheong Hin Fatt and Dr. Herman Gerritsen, for their insightful comments and excellent suggestions on my thesis.

Special thanks to Dr. Sisomphon, who introduced me to Delft3D modelling, and proposed numerous inspiring ideas on my research. The stimulating discussions with her have established a solid basis for this thesis. Thanks are extended to my colleagues in Singapore-Delft Water Alliance, Mr. Klaas Pieter, Ms. Tay Hui Xin, Ms. Arunoda, Ms. Wang Xuan, Mr. Alamsyah Kurniawan, Mr. Pavlo Zemskyy, Dr. Rao Raghu and Dr. SK

Ooi, as well as my colleagues in Deltares, Dr. Daniel Twigt and Dr. Firmijn Zijl, for the enjoyable working experience we share together and their help on my thesis.

I am also thankful to Mr. Krishna and Ms. Norela from the Hydraulic Lab, for their essential assistance in various aspects.

The financial support from the National University of Singapore is gratefully acknowledged.

Additional thanks to my friends, Dr. Liu Dongming, Mr. Lin Quanhong, Mr. Chen Haoliang, Mr. Zhang Wenyu, Dr. Gu Hanbin, Mr. Xu Haihua, Dr. Dulakshi, Dr. Ma Peifeng, Dr. Wang Zengrong, Dr. Cheng Yonggang, Dr. Zhou Xiaoquan, Mr. Zhang Xu and Mr. Wang Li, for all the great time we spent together and the everlasting friendship we have.

Heartfelt thanks to my dear parents and my wife, who continuously support me with their love. Without their understanding and encouragement, it would have been impossible for me to accomplish this work.

# Table of Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Table of Contents</b>	<b>iii</b>
<b>Summary</b>	<b>viii</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Symbols</b>	<b>xvii</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Background.....	1
1.2 Review of Data Assimilation.....	3
1.2.1 Classification.....	3
1.2.2 Methodology.....	5
1.3 Overview of Singapore Regional Model.....	6
1.4 Objectives of Present Study.....	8
1.5 Organization of Thesis.....	10
<b>Chapter 2 Chaos Theory</b>	<b>13</b>

2.1 Introduction	14
2.2 Time-delay Embedding Theorem	15
2.3 System Characterization	16
2.4 Phase Space Reconstruction	18
2.4.1 Time Delay $\tau$	19
2.4.2 Embedding Dimension $m$	20
2.5 Time Series Prediction	22
2.5.1 Local Model	22
2.5.2 Standard Approach	24
2.5.3 Inverse Approach	24
2.5.4 Lorenz Time Series Prediction	26
<b>Chapter 3 Artificial Neural Networks</b>	<b>36</b>
3.1 Introduction	36
3.2 Neuron	37
3.3 Activation Function	38
3.4 Multilayer Perceptron	39
3.5 Back-propagation Algorithm	40
3.6 Application of Multilayer Perceptron	41
3.6.1 Network Architecture	41
3.6.2 Lorenz Time Series Prediction	42

<b>Chapter 4 Kalman Filter</b>	<b>47</b>
4.1 Linear Kalman Filter	47
4.2 Extended Kalman Filter	50
4.3 Steady-state Kalman Filter	52
4.4 Application of Kalman Filter in Error Distribution	53
<b>Chapter 5 Singapore Regional Model</b>	<b>56</b>
5.1 Delft3D-FLOW	56
5.1.1 Introduction	56
5.1.2 Governing Equations	57
5.1.3 Numerical Aspects	60
5.2 Singapore Regional Model	62
5.2.1 Model Set-up	62
5.2.2 Numerical Simulation	63
<b>Chapter 6 Error Prediction with Local Model and Multilayer Perceptron</b>	<b>72</b>
6.1 Introduction	72
6.2 Application of Local Model in Error Prediction	73
6.2.1 Chaos Identification	73
6.2.2 Parameter Determination	73
6.2.3 Results	74
6.3 Application of Multilayer Perceptron in Error Prediction	75

6.3.1 Methodology	75
6.3.2 Results	77
6.4 Comparison between Local Model and Multilayer Perceptron	77
<b>Chapter 7 Error Distribution with Kalman Filter and Multilayer Perceptron</b>	<b>94</b>
7.1 Introduction	94
7.2 Application of Kalman Filter in Error Distribution	95
7.2.1 Error Statistics Approximation	95
7.2.2 Results	97
7.3 Application of Multilayer Perceptron in Error Distribution	97
7.3.1 Methodology	97
7.3.2 Results	99
7.4 Comparison between Kalman Filter and Multilayer Perceptron	100
<b>Chapter 8 Use of Data Assimilation in Understanding Sea Level Anomalies</b>	<b>111</b>
8.1 Introduction	111
8.2 Overview of Sea Level Anomalies	112
8.2.1 Sources of Marine Data	112
8.2.2 Extraction of Sea Level Anomalies	113
8.2.3 Statistical Analysis of Sea Level Anomalies	115
8.2.4 RADS SLA vs. DUACS SLA	116
8.2.5 Altimeter SLA vs. In-situ SLA	117

8.3 Assimilation of Sea Level Anomalies into Singapore Regional Model.....	118
8.3.1 Prediction of SLA at Open Boundaries.....	119
8.3.1.1 Preprocess of SLA Time Series.....	119
8.3.1.2 Methodology.....	119
8.3.1.3 Results.....	121
8.3.2 Numerical Simulation of Internal SLA.....	121
8.4 Research in Progress and Future.....	122
<b>Chapter 9 Conclusions and Recommendations</b>	<b>139</b>
9.1 Conclusions.....	139
9.2 Recommendations.....	141
<b>References</b>	<b>143</b>
<b>Appendix A</b>	<b>151</b>
<b>Appendix B</b>	<b>161</b>
<b>List of Publications</b>	<b>166</b>



## Summary

One primary objective of this study is to develop and implement applicable data assimilation methods to improve the forecasting accuracy of the Singapore Regional Model. A novel hybrid data assimilation scheme is proposed, which assimilates the observed data into the numerical model in two steps: (i) predicting the model errors at the measurement stations, and (ii) distributing the predicted errors to the non-measurement stations. Specifically, three approaches are studied, the local model approach (LM), the multilayer perceptron (MLP), and the Kalman filter (KF).

At the stations where observations are available, both the local model approach and the multilayer perceptron are utilized to forecast the model errors based on the patterns revealed in the phase spaces reconstructed by the past recordings. In cases of smaller prediction horizons, such as  $T = 2, 24$  hours, the local model approach outperforms the multilayer perceptron. However, due to the less competency of the local model approach in capturing the trajectories of the state vectors in the higher-dimensional phase spaces, the prediction accuracy of the local model approach decreases by a wider margin when  $T$  progresses to 48, 96 hours. Averaged over 5 different prediction horizons, both methods are able to remove more than 60% of the root mean square errors (RMSE) in the model error time series, while the multilayer perceptron performs slightly better.

To extend the updating ability to the remainder of the model domain, Kalman filter and the multilayer perceptron are used to spatially distribute the predicted model errors to the non-measurement stations. When the outputs of the Singapore Regional Model at the non-measurement stations and the measurement stations are highly correlated, such as at Bukom and Raffles, both approaches exhibit remarkable potentials of distributing the predicted errors to the non-measurement stations, resulting in an error reduction of more than 50% on average. However, the performance of Kalman filter in error distribution deteriorates at a rapid pace when the correlation decreases, with only about 40% of the root mean square errors removed at Sembawang and 20% at Horsburgh. Comparatively, the multilayer perceptron is less sensitive to the correlations with a more consistent performance, which removes more than 40% of the root mean square errors at Sembawang and Horsburgh. In addition, the error distribution study demonstrates for the first time that distributing the predicted errors from more measurement stations does not necessarily produce the best results due to the misleading information from less correlated stations. As suggested by this finding, to conduct a prior correlation analysis among possible sites is favorable when planning the future layout of the measurement stations.

Another major objective of this study is to analyze and predict the sea level anomalies by means of data assimilation. Sea level anomalies are extracted based on tidal analysis from both altimeter data and in-situ measurements. A reasonable fit between the altimeter sea level anomalies and the in-situ sea level anomalies can be observed, indicating the coherence and consistency of different data sources. As a demonstration of the proposed

data assimilation scheme, the sea level anomalies explored in this study are the spatially and temporally interpolated DUACS sea level anomalies.

At the open boundaries of the Singapore Regional Model, the sea level anomaly time series are predicted using multilayer perceptron with prediction horizon  $T = 24$  hours. Multilayer perceptron successfully captures the motion dynamics of the sea level anomalies, with more than 90% of the root mean squares (RMS, quadratic mean) removed on average. The sea level anomalies inside the model domain are then numerically modelled by imposing the sea level anomalies predicted at the open boundaries as driving force to the Singapore Regional Model. A reasonable correspondence are observed between the modelled sea level anomalies and the DUACS sea level anomalies, verifying that the internal sea level anomalies can be decently modelled through numerical simulation provided that the sea level anomalies are properly prescribed at the open boundaries.

## List of Tables

Table 2.1 Parameters in the inverse approach for Lorenz model.	35
Table 5.1 Statistics of model errors at the measurement stations.	71
Table 6.1 Parameter settings in genetic algorithm.	89
Table 6.2 Embedding parameters ( $m, \tau, k$ ) in local model.	90
Table 6.3 Statistics of residual errors at the measurement stations (local model).	91
Table 6.4 Embedding parameters ( $m, \tau$ ) in multilayer perceptron.	92
Table 6.5 Statistics of residual errors at the measurement stations (multilayer perceptron).	93
Table 7.1 Correlation coefficient between the SRM outputs at the measurement stations and the non-measurement stations.	106
Table 7.2 Statistics of residual errors at Bukom (Kalman filter; *: best case).	107
Table 7.3 Statistics of residual errors at Raffles (Kalman filter; *: best case).	107
Table 7.4 Statistics of residual errors at Sembawang (Kalman filter; *: best case).	108
Table 7.5 Statistics of residual errors at Horsburgh (Kalman filter; *: best case).	108
Table 7.6 Statistics of residual errors at Bukom (multilayer perceptron; *: best case).	109
Table 7.7 Statistics of residual errors at Raffles (multilayer perceptron; *: best case).	109

Table 7.8 Statistics of residual errors at Sembawang (multilayer perceptron; *: best case).	110
Table 7.9 Statistics of residual errors at Horsburgh (multilayer perceptron; *: best case).	110
Table 8.1 General aspects of Jason-1 and Envisat.	137
Table 8.2 Summary of statistical analysis results of the sea level anomalies.	138

## List of Figures

Figure 1.1	Variational data assimilation approach.	11
Figure 1.2	Sequential data assimilation approach.	11
Figure 1.3	Schematic diagram of simulation and forecasting with emphasis on the four different updating methodologies	12
Figure 2.1	Lorenz time series.	28
Figure 2.2	Fourier power spectrum of Lorenz time series.	29
Figure 2.3	Correlation integral analysis for Lorenz time series.	29
Figure 2.4	Average mutual information of Lorenz time series.	30
Figure 2.5	False nearest neighbors analysis for Lorenz time series.	30
Figure 2.6	Reconstructed phase space for Lorenz model.	31
Figure 2.7	Conceptual sketch of the local model approach.	32
Figure 2.8	Flow diagram of genetic algorithm.	33
Figure 2.9	Schematic illustration of evolving process in genetic algorithm.	33
Figure 2.10	Lorenz time series prediction using local model (standard approach; $T=2$ ).	34
Figure 2.11	Lorenz time series prediction using local model (inverse approach; $T=2$ ).	34
Figure 3.1	Nonlinear model of a neuron.	44
Figure 3.2	Model of a single-layer perceptron.	44

Figure 3.3	Architectural graph of a multilayer perceptron with two hidden layers.	45
Figure 3.4	Lorenz time series prediction using multilayer perceptron (T=2).	46
Figure 4.1	Linear Kalman filter algorithm.	55
Figure 4.2	Extended Kalman filter algorithm.	55
Figure 5.1	Staggered grid of Delft3D-FLOW.	66
Figure 5.2	Extent, grid and bathymetry of Singapore Regional Model.	67
Figure 5.3	Measurement stations around Singapore.	68
Figure 5.4	SRM outputs, observations and model errors at Jurong.	69
Figure 5.5	SRM outputs, observations and model errors at Horsburgh.	69
Figure 5.6	Model errors at Jurong.	70
Figure 5.7	Model errors at Horsburgh.	70
Figure 6.1	Correlation integral analysis for the model error time series at Jurong.	80
Figure 6.2	Reconstructed phase space for the model errors at Jurong (T=2 hours).	81
Figure 6.3	Error prediction with local model at Jurong (T=2 hours).	82
Figure 6.4	Error prediction with local model at Jurong (T=96 hours).	82
Figure 6.5	Error prediction with local model at Horsburgh (T=2 hours).	83
Figure 6.6	Error prediction with local model at Horsburgh (T=96 hours).	83
Figure 6.7	Scatter diagrams of SRM outputs at Jurong.	84
Figure 6.8	Scatter diagrams of LM corrected outputs at Jurong (T=2 hours).	84
Figure 6.9	Average mutual information of the model errors at Jurong.	85

Figure 6.10 False nearest neighbors analysis for the model errors at Jurong.	85
Figure 6.11 Architecture of multilayer perceptron in error prediction.	86
Figure 6.12 Error prediction with multilayer perceptron at Jurong (T=2 hours).	87
Figure 6.13 Error prediction with multilayer perceptron at Jurong (T=96 hours).	87
Figure 6.14 RMSE vs. prediction horizon at Jurong.	88
Figure 6.15 RMSE vs. prediction horizon at Horsburgh.	88
Figure 7.1 Error distribution with Kalman filter at Horsburgh (T=2 hours; Case 3).	102
Figure 7.2 Error distribution with Kalman filter at Horsburgh (T=96 hours; Case 3).	102
Figure 7.3 Architecture of multilayer perceptron in error distribution.	103
Figure 7.4 Error distribution with multilayer perceptron at Horsburgh (T=2 hours; Case 3).	104
Figure 7.5 Error distribution with multilayer perceptron at Horsburgh (T=96 hours; Case 3).	104
Figure 7.6 RMSE vs. prediction horizon at Horsburgh.	105
Figure 8.1 Jason-1 (upper) and Envisat (lower) ground tracks.	124
Figure 8.2 Locations of the UHSLC stations.	125
Figure 8.3 Amplitudes (upper) and phases (lower) of M2 from RADS altimeter data and from in-site measurements.	126
Figure 8.4 Along track RADS sea level anomalies for period from 14 <sup>th</sup> to 29 <sup>th</sup> November 2005.	127
Figure 8.5 Gridded DUACS sea level anomalies for period from 16 <sup>th</sup> to 30 <sup>th</sup> November 2005.	128
Figure 8.6 Comparison of sea level anomalies obtained from the RADS and DUACS data sets with sea level anomalies obtained from UHSLC in-situ measurements (Kelang/140; 2005).	129



Figure 8.7	Comparison of sea level anomalies obtained from the RADS and DUACS data sets with sea level anomalies obtained from UHSLC in-situ measurements (Cendering /320; 2005).	129
Figure 8.8	Extent, bathymetry of the Singapore Regional Model with 17 boundary support points.	130
Figure 8.9	Extracted SLA at selected Singapore Regional Model SCS, Andaman Sea, and Java Sea boundary support points.	131
Figure 8.10	Architecture of multilayer perceptron in sea level anomaly prediction.	132
Figure 8.11	SLA prediction with multilayer perceptron at SCS boundary (ID 9; T=24 hours).	133
Figure 8.12	SLA prediction with multilayer perceptron at Andaman Sea boundary (ID 4; T=24 hours).	133
Figure 8.13	SLA prediction with multilayer perceptron at Java Sea boundary (ID 15; T=24 hours).	134
Figure 8.14	SRM simulated SLA (red line) compared to DUACS SLA (blue asterisks) at Tanjong Pagar.	135
Figure 8.15	SRM simulated SLA (left panels) compared to DUACS SLA maps (right panels).	136
Figure A.1	Signal-flow graph of output neuron $j$ .	159
Figure A.2	Signal-flow graph of hidden neuron $j$ connected to output neuron $k$ .	159
Figure A.3	Back-propagation algorithm cycle.	160

## List of Symbols

$A_o$	mean water level
$A_i$	amplitude of a constituent
$\mathbf{A}_k$	matrix that relates the state vectors
$b_k$	bias
$\mathbf{B}_k$	matrix that relates the forcing term to the state
$Cr$	Courant number
$C(\varepsilon)$	correlation integral
$d$	correlation dimension
$\vec{d}$	external forces
$E_{av}$	average squared error energy
$E(n)$	instantaneous error energy
$\hat{\mathbf{E}}_M$	model errors forecasted at the measurement stations
$\hat{\mathbf{E}}_N$	distributed errors at the non-measurement stations
$f$	Coriolis coefficient
$f(\cdot)$	nonlinear model operator

$f_T(\cdot)$	mapping function
$F_i$	amplitude factor of a constituent
$F_\xi, F_\eta$	turbulent momentum fluxes in $\xi$ and $\eta$ directions
$g_T(\cdot)$	alternative mapping function
$G_i$	phase lag of a constituent
$\sqrt{G_{\xi\xi}}, \sqrt{G_{\eta\eta}}$	coefficient transforming orthogonal curvilinear co-ordinates to Cartesian rectangular co-ordinates
$h(\cdot)$	nonlinear measurement operator
$H(\cdot)$	Heaviside step function
$H(t)$	astronomic tidal level
$\mathbf{H}_k$	matrix that relates the state to the measurement
$i$	index of a constituent
$I_{AB}$	average mutual information between $A$ measurements and $B$ measurements
$I(\tau)$	average mutual information between $x_i$ and $x_{i-\tau}$
$k$	no. of nearest neighbors / no. of relevant constituents
$\mathbf{K}_k$	Kalman gain
$m$	embedding dimension
$M_\xi, M_\eta$	sources/sinks of momentum in $\xi$ and $\eta$ directions

$\mathbf{M}$	matrix of the numerical model outputs at the measurement stations
$N$	length of the time series
$\mathbf{N}$	matrix of the numerical model outputs at the non-measurement stations
$P_A(a)$	individual probability density for measurements $A$
$P_{AB}(a,b)$	joint probability density for measurements $A$ and $B$
$P_B(b)$	individual probability density for measurements $B$
$P_i$	population of chromosomes
$P_\xi, P_\eta$	hydrostatic pressure gradients in $\xi$ and $\eta$ directions
$\mathbf{P}_k^a$	error covariances for the analysis estimate
$\mathbf{P}_k^f$	error covariances for the forecast estimate
$Q$	global source/sink per unit area
$\mathbf{Q}_k$	model error covariance
$r$	correlation coefficient
$R_i(m)^2$	square of the Euclidian distance between $\mathbf{x}_i$ and $\mathbf{x}_i^{NV}$
RMS	root mean square / quadratic mean
RMSE	root mean square error
$\mathbf{R}_k$	measurement error covariance
$t$	time
$T$	lead time (prediction horizon)
$u, v, w$	flow velocities in $x$ , $y$ and $z$ directions

$u_k$	linear combiner output
$\mathbf{u}_k$	forcing term
$U, V$	depth-averaged velocity in $\xi$ and $\eta$ directions
$\nu$	correlation exponent
$v_k$	induced local field (activation potential)
$\mathbf{v}_k$	measurement noise
$(V_o + u)_i$	astronomical argument of a constituent
$w_{kj}$	synaptic weights of neuron $k$
$\mathbf{w}_{k-1}$	model noise
$x_i$	scalar time series
$x_j$	input signals
$\mathbf{x}_i$	phase space vector
$\mathbf{x}_i^{NN}$	nearest neighbor of $\mathbf{x}_i$
$\mathbf{x}_k$	state vector
$\mathbf{x}_k^a$	analysis state estimate
$\mathbf{x}_k^f$	forecast state estimate
$\mathbf{x}_t$	current state
$\mathbf{x}_{t+T}$	future state
$\hat{\mathbf{x}}_{t+T}$	'expected' future state

$y_k$	output signal
$z_k$	measurement vector
$\alpha$	momentum constant
$\varepsilon$	threshold distance
$\delta_j(n)$	local gradient
$\zeta$	free surface elevation above the horizontal reference plane
$\eta$	learning rate
$\eta_i$	observed values
$\eta_i'$	Singapore Regional Model outputs
$\lambda$	linearized bottom friction coefficient
$\nu_{3D}$	eddy viscosity due to 3D turbulence
$\nu_{mol}$	kinematic viscosity
$\nu_V^{back}$	background vertical eddy viscosity
$\xi, \eta$	horizontal orthogonal curvilinear co-ordinates
$\rho$	spatial correlation for the model errors
$\rho_0$	reference water density
$\sigma$	vertical co-ordinate
$\sigma_{mei}$	standard deviation for the measurement errors
$\sigma_{mo}$	standard deviation for the model errors
$\tau$	time delay

$\varphi(\cdot)$	activation function
$\phi(\cdot)$	mapping function
$\omega$	flow velocity in $\sigma$ direction
$\omega_i$	angular velocity of a constituent

# **Chapter 1**

## **Introduction**

### **1.1 Background**

Oceanographic system forecasting is of prime importance for safe navigation and offshore operations as well as understanding oceanographic physics, such as ocean waves, ocean currents, transport and mixing characteristics. Great effort has been devoted to developing different approaches to forecast the oceanographic system. These approaches can be classified into three general categories: numerical models, data mining and data assimilation.

With the development of computer science, the use of numerical models that are governed by a set of mathematical equations is the preferred way for researchers to predict the future of oceanographic system. Numerous numerical models have been developed under different numerical environments to describe the movement of local water or even the circulation of entire ocean (Pugh, 1996; Palacio et al., 2001; Marchuk et.al, 2003). The improvement of numerical calculation and the increasing power of computers made people extremely confident in the competence of the numerical models. It was believed that numerical models could become complex enough to reach any level of precision, simply by refining the model scales and calculating for long enough.



However, some researchers have indicated that the numerical models are far from being perfect as they are indeed only models of reality (Madsen et al., 2003; Babovic et al., 2005; Mancarella et al., 2007). The prediction capability of the numerical models could be diminished due to certain inherent delimiting factors, such as simplifying assumptions employed in the numerical models, errors in the numerical schemes, inaccuracy in the model parameters and uncertainty in the prescribed forcing terms. Therefore, numerical models tend to produce imperfect model results even if the governing laws can model the prediction framework with good aptness.

The opposite approach to numerical models in oceanographic forecasting is encompassed in the term data mining. The original philosophy behind data mining is the attempt to circumvent the numerical models. Data mining has become an important tool to transform data into information as a process of extracting hidden patterns from data. In domains where the numerical models are poor and data have been collected over long periods, through data mining the researchers would be able to capture and reproduce the dynamics of the system just by analyzing the data (Cipolla, 1995; Wang, 1999; Poncelet et al., 2007). However, the performance of data mining critically relies on the data quality and availability. Sometimes the size and complexity of the data make it difficult to find useful information (Kamath, 2006; Hong et al., 2009). Discarding the experience accumulated by the refinement of theories also makes data mining less convincing to the researchers who wonder about the science still undiscovered in the data.

With the objective to take the best of both numerical models and observed data, a method referred to as data assimilation was designed, following the terminology in

meteorology (Daley, 1991). As defined by Robinson et al. (1998), data assimilation is a methodology that can optimize the extraction of reliable information from observed data, and assimilate it into the numerical models to improve the quality of the estimate. Due to the outstanding accuracy in forecasting the natural systems, data assimilation has recently attracted extensive research effort with a wide range of applications, such as physics, economics, earth sciences, hydrology and oceanography (Hartnack and Madsen, 2001; Haugen and Evensen, 2002; Reichle, 2008).

In the following sections, an attempt is made to review in general terms the most well-known and applied data assimilation techniques, followed by a brief review of the Singapore Regional Model (SRM), the objectives of present study and the organization of thesis.

## **1.2 Review of Data Assimilation**

### **1.2.1 Classification**

According to the way the system is updated, data assimilation can be divided into two different categories.

- Variational data assimilation:

Variational data assimilation is based on the optimal control theory. Optimization is performed by minimizing a given cost function that measures the model to data misfit. As illustrated in Figure 1.1, variational data assimilation corrects the initial conditions of the model in order to obtain the best overall fit of the state to the observations based on all

the data available during the assimilation period, from the start of the modelling until the present time.

The most widely applied variational data assimilation is the adjoint method (Le Dimet and Talagrand, 1986; Nechaev and Yaremchuk, 1994; Luong et al., 1998). The adjoint method computes the gradient of a quadratic function with respect to the variables to be adjusted, and then approaches the exact trajectory of the state by propagating backwards the differences with the adjoint equations. The adjoint method has been applied for off-line estimation of model parameters. However, the complexity of the adjoint methods makes it a difficult task to apply such methods in on-line forecasting procedures.

- Sequential data assimilation:

Sequential data assimilation is usually associated with estimation theory, where the system state is estimated sequentially by propagating information only forward in time. As illustrated in Figure 1.2, sequential data assimilation corrects the present state of the model as soon as the observations are available. In contrast to variational data assimilation, sequential data assimilation usually leads to discontinuities in the time series of the corrected state.

Many sequential data assimilation methods have been proposed in recent years, such as in Cañizares (1999), Pham (2000), Verlaan and Heemink (2001). Sequential data assimilation avoids driving numerical models backwards, which makes it more applicable for updating the system state and hence results in more research effort directed to its development.

### 1.2.2 Methodology

Referred to as process models in WMO (1992) and Refsgaard (1997), Numerical models can be described as a set of equations that contain state variables and parameters. In classical numerical stimulation, state variables vary with time whereas parameters remain constant. According to the variables modified during the updating process, four different methodologies of data assimilation have been defined as follows (see Figure 1.3):

- Updating of input variables:

Updating of Input variables is the classical method, justified by the fact that input uncertainties may be the dominant error source in operational forecasting.

- Updating of state variables:

State variables are a set of variables that represent the state of a general system. The adjustment of the state variables can be done in different ways. The theoretically most comprehensive methodology is based on Kalman filter (KF, Kalman, 1960). Kalman filter was originally proposed as the optimal updating procedure for linear systems, but with some modifications, Kalman filter also provides approximate solutions for nonlinear systems.

- Updating of model parameters:

As the operation of any numerical system cannot significantly change over the short interval of time, recalibration of the model parameters at every time step has no real advantages for numerical models of nontrivial complexity, Therefore, updating of model parameters remains debatable and is least popular as a data assimilation method.

- Updating of output variables:

The deviations between the forecasted and the observed data are called model errors. The model errors are usually found to be serially correlated, making it possible to forecast the future values of these errors. Predicting the model errors and then superimposing on the numerical model outputs usually simulate the system with a better accuracy. This method is most often referred to as error prediction.

### **1.3 Overview of Singapore Regional Model**

Motivated by different interests involved in safety, ecology and economy, Singapore has a great thirst for accurate water level prediction. With the intention to provide reliable hydrodynamic information of the water surrounding Singapore, the Singapore Regional Model (SRM) was developed in 2004 by WL | Delft hydraulics, the Netherlands (Kernkamp and Zijl, 2004).

The Singapore Regional Model was constructed within the Delft3D modelling system, which is Deltares' state-of-the-art framework for the modelling of surface water systems (Deltares, 2009). The Singapore Regional Model has been intensively calibrated, and is able to predict the water levels for any selected period with reasonably good accuracy. However, noticeable errors can still be observed between the model output and the water level measurements due to certain limitations in the model setup and in the numerical modelling.

At the open boundaries of the Singapore Regional Model, 8 tidal constituents, i.e. Q1, O1, P1, K1, N2, M2, S2 and K2, are prescribed to generate water level time series as the forcing terms to the numerical model. The generated water levels propagate according to

the numerical rule from the open boundary to the model domain. In tide theory, the astronomical component of water levels can be decomposed into 234 tidal constituents in total (Kantha and Clayson, 1999). Although the 8 tidal constituents prescribed account for most portions of water levels, the missing of other constituents can still sacrifice the forecasting accuracy to a great extent.

Wind stress on the sea surface is an important factor which affects the water levels. When the wind blows in one direction, it will push against the water and cause the water to pile up higher than the normal sea level. This pile of water is pushed and propagated in the direction of wind, generating the meteorological component of sea level referred to as a storm surge. However, due to the lack of available wind information, wind is not included in the setup of the Singapore Regional Model. This distinction from real condition neglects the contribution from the storm surge, and hence generates discrepancies between the observed water levels, especially in the two significant monsoon seasons.

The Delft3D modelling system consists of a set of partial differential equations, describing how the state variables evolve in time. Solving these equations requires discretization in space and time, which entails that only processes with scales larger than grid sizes and time steps can be reproduced reliably. In addition, the Singapore Regional Model contains model parameters, such as model bathymetry, bottom roughness and viscosity coefficients. These parameters are not known exactly and determined empirically.

The error sources stated above would accumulate to generate model errors in the Singapore Regional Model output. Inaccurate water levels predicted may lead to concerning issues, such as unnecessary high fuel consumption due to sub-optimal route, increased port operating costs due to delays and rescheduling, and uncertainties in the trajectory track of sediment transport, etc.

#### **1.4 Objectives of Present Study**

One primary objective of this study is to develop and implement applicable data assimilation methods to improve the forecasting accuracy of the Singapore Regional Model. Depending on the availability of the observed water levels, this objective is specifically achieved in two steps, i.e. model error prediction and then model error distribution.

At the stations where observations are available in the model domain, future values of the model errors can be directly forecasted based on the past recordings. Two state-of-art time series prediction methods are herein adopted, i.e. local model (LM) based on chaos theory, and multilayer perceptron (MLP) in artificial neural networks (ANN). Local model and multilayer perceptron are widely used in time series prediction due to their favourable applicability, but no research has been done to compare their performance. In this study, both methods are applied to predict the model error time series, with a thorough performance comparison conducted afterwards.

The effect of error prediction is confined within the measurement stations. To extend the updating ability to the remainder of the computational domain, two approaches of

error distribution are explored, i.e. Kalman filter and multilayer perceptron. Kalman filter is a recursive algorithm to estimate the system state, whereas multilayer perceptron determines the variable relationships by simulating the human brains. This study applies both Kalman filter and multilayer perceptron to distribute the model errors to the non-measurement stations, and also compares their performance afterwards.

Sea level anomalies (SLA) are important phenomena in the Singapore and Malacca Straits. At times sea level anomalies can overtake the regular tidal flow conditions, causing serious troubles for ship navigation and port operation. Research reveals that sea level anomalies mostly result from persistent basin-scale monsoon winds and their short scale variations over the South China Sea and Andaman Sea. Failing to consider the influence from the wind, the Singapore Regional Model is incompetent to numerically capture the dynamics of the sea level anomalies. This motivates another major objective of this study, i.e. to analyze and predict sea level anomalies by means of assimilating the sea level anomaly measurements into the numerical model.

Sea level anomalies are extracted based on tidal analysis from both altimeter data and in-situ measurements, whereas the altimeter sea level anomalies are explored in this study as a demonstration of the data assimilation scheme. At the open boundaries of the Singapore Regional Model, the sea level anomaly time series are predicted using multilayer perceptron. The sea level anomalies inside the model domain are then numerically modelled by imposing the sea level anomalies predicted at the open boundaries as driving force to the Singapore Regional Model. To assess the efficiency of



the data assimilation scheme, the predicted sea level anomalies and the modelled sea level anomalies will be compared with the altimeter sea level anomalies.

## **1.5 Organization of Thesis**

Chapters 2, 3, and 4 review in detail the techniques involved, i.e. chaos theory, artificial neural networks and Kalman filter.

Chapter 5 first introduces the numerical modelling system – Delft3D-FLOW, including conceptual description and numerical aspects, whereafter the dedicated Singapore Regional Model is described.

Chapter 6 applies local model and multilayer perceptron in model error prediction. Detailed comparison results on the prediction performance are also presented.

Chapter 7 demonstrates the application of Kalman filter and multilayer perceptron in error distribution, with a performance comparison conducted thereafter.

Chapter 8 studies the features of the sea level anomalies, and applies data assimilation techniques on the prediction of sea level anomalies.

Chapter 9 draws conclusions resulting from the present study. A number of recommendations for the further research are given in the end.

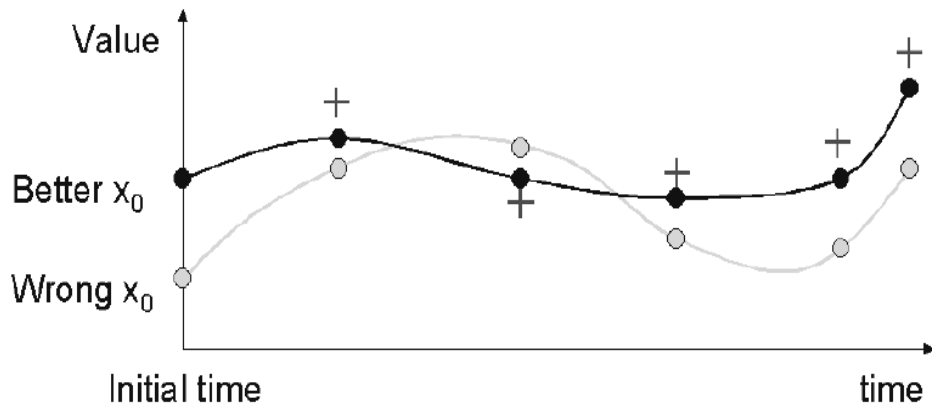


Figure 1.1 Variational data assimilation approach. The original model run (grey line and dots) is given better initial conditions that lead to a new model run (black line and dots) closer to the observations (+).

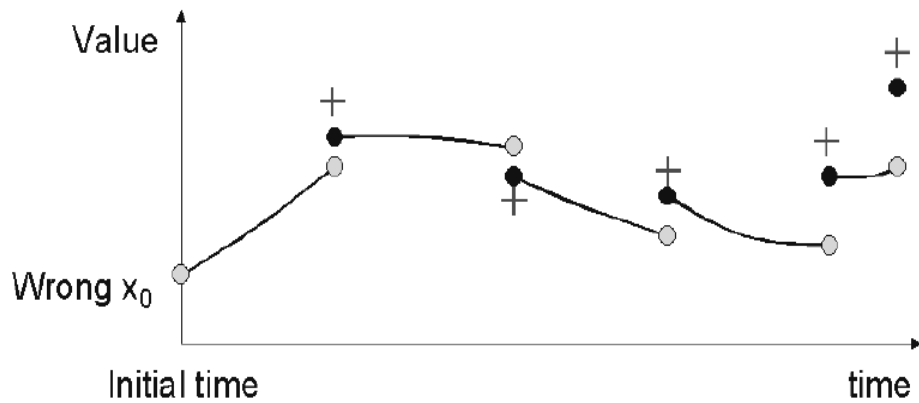


Figure 1.2 Sequential data assimilation approach. When an observation (+) is available, the model forecast (grey dot) is updated to a value closer to the observation (black dot) that is used to make the next model forecast.

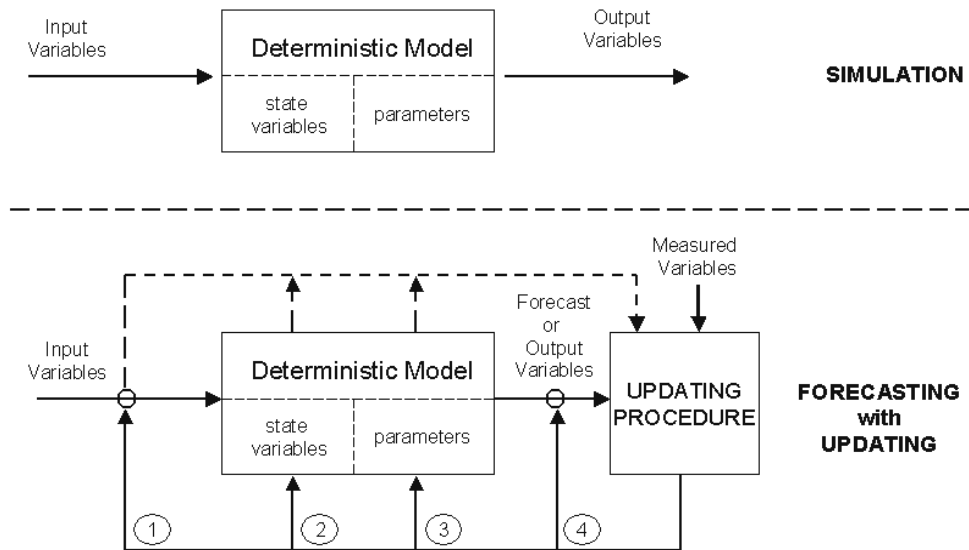


Figure 1.3 Schematic diagram of simulation and forecasting with emphasis on the four different updating methodologies (Adapted from Refsgaard, 1997).

## Chapter 2

### Chaos Theory

Time series prediction plays an important role in various fields, ranging from economics through physics to engineering. Fundamentally, the goal of time series prediction is to estimate some future value based on current and past data samples. Mathematically stated,

$$x_{t+T} = f(x_{t-a}, x_{t-b}, x_{t-c} \dots), \quad (2.1)$$

where  $x_{t+T}$  is the future value of a discrete time series  $x_i$ . The mapping function  $f(\cdot)$  in Equation (2.1) is required to be determined, such that the predicted future value  $\hat{x}_{t+T}$  is unbiased and consistent.

The traditional statistical fitting methods, such as autoregressive (AR), moving average (MA) and autoregressive moving average (ARMA) models, have once dominated the fields of time series analysis (Box and Jenkins, 1976). In these models, the future values of the time series are expressed as a linear combination of the current and past data samples weighted by a set of coefficients plus residual white noise. However, due to the inherent linearity assumptions, such appealing simplicity can be entirely inapplicable in the complex systems where weak nonlinearities occur (Pasternack, 1999; Ding et al., 2008).

With the recent development in chaos theory, numerous nonlinear systems have been identified to arise from purely deterministic dynamics despite their random behaviors. Time series analysis within the chaotic dynamic system has hence gained popularity in a variety of applications (Ott, 1993; Alligood et al., 1997; Babovic et al., 2001; Sprott, 2003; Karunasinghe and Liong, 2006).

## 2.1 Introduction

Chaos is not a rare phenomenon. Chaotic behaviors have been widely observed in the laboratory and nature, such as molecular vibrations, chemical reactions, magnetic fields and fluid dynamics. Defined by Williams (1997), chaos is a sustained and disorderly-looking evolution that satisfies certain special mathematical criteria and that occurs in a deterministic nonlinear system.

An early pioneer of chaos theory was Edward Lorenz, whose interest in chaos came about accidentally through his work on weather prediction (Lorenz, 1963). Lorenz discovered that even tiny changes in initial conditions could produce large changes in the long-term weather prediction. This finding is popularly known as the “Butterfly Effect”, as Lorenz stated that ‘the flap of a butterfly’s wings in Brazil may set off a tornado in Texas’. This quote essentially reveals the extreme sensitivity of chaos to its initial conditions.

Lorenz model is a system of 3 ordinary differential equations abstracted by Lorenz from the Galerkin approximation to the partial differential equations of thermal convection in the lower atmosphere derived by Salzman (1962). The equations read,

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= -xz + rx - y, \\ \dot{z} &= xy - bz\end{aligned}\tag{2.2}$$

where  $\sigma$ ,  $r$  and  $b$  are parameters with standard values  $\sigma = 16$ ,  $b = 4$  and  $r = 45.92$ . The  $x(t)$  component is solved using a straightforward fourth order Runge-Kutta method with time step of  $\Delta t = 0.01$ . As plotted in Figure 2.1, the orbits of the  $x(t)$  component exhibit non-periodic motion with chaotic characteristics. Lorenz model is a typical example of the chaotic system, and will be used as prototype of time series prediction in Chapters 2 and 3.

## 2.2 Time-delay Embedding Theorem

Takens' time-delay embedding theorem (Takens, 1981) paved the way for the analysis of chaotic time series in the chaotic systems. This theorem establishes that, given a scalar time series  $x_i$  from a chaotic system, it is possible to reconstruct a phase space in terms of the phase space vectors  $\mathbf{x}_i$  expressed as

$$\mathbf{x}_i = \left( x_i, x_{i-\tau}, \dots, x_{i-(m-1)\tau} \right),\tag{2.3}$$

where  $m$  is the embedding dimension, and  $\tau$  is the time delay. The time-delay embedding theorem essentially indicates that the underlying structures in the chaotic time series cannot be seen in the scalar space, but can only be equivalently viewed when unfolded into the phase space.

In general, analysis of chaotic time series can be divided into three phases:

- System characterization;
- Phase space reconstruction;
- Time series prediction.

System characterization investigates whether a time series is chaotic or not. Being identified chaotic, the time series can be projected into a phase space, which is reconstructed through the optimization of the time delay  $\tau$  and the embedding dimension  $m$ . Based on the underlying structures revealed in the phase space, the chaotic time series can be correspondingly predicted.

### **2.3 System Characterization**

For the systems evolving with deterministic equations, broadband power spectra are sufficient to identify chaos. However, identification of chaos is a difficult task in real world where the governing equations are not always available. As the stochastic time series also has broadband power spectra, Fourier analysis alone is not sufficient to recognize chaotic behaviors. A number of methods have emerged to distinguish the chaotic time series from the stochastic time series, such as the Kolmogorov entropy method (Grassberger and Procaccia, 1983a), the Lyapunov exponent method (Wolf et al., 1985) and the surrogate data method (Schreiber and Schmitz, 1996). Among these methods, the correlation dimension method, proposed by Grassberger and Procaccia (1983b, c), is the most popular with wide applications in meteorology, geology and hydrology.

The correlation dimension method is also called the correlation integral analysis (CIA), as the correlation integral is usually used to estimate the correlation dimension. The correlation integral is the mean probability that the states at two different times are close. Consider a set of state vectors  $\mathbf{x}_i$ , the correlation integral can be expressed by

$$C(\varepsilon) = \lim_{N \rightarrow \infty} \frac{1}{N^2} \sum_{i,j=1}^N H(\varepsilon - |\mathbf{x}_i - \mathbf{x}_j|) \quad (2.4)$$

where  $N$  is the number of considered states,  $H(\cdot)$  is the Heaviside step function,  $\varepsilon$  is a threshold distance, and  $\sum_{i,j=1}^N H(\varepsilon - |\mathbf{x}_i - \mathbf{x}_j|)$  counts the number of pairs  $(i, j)$  whose distance  $|\mathbf{x}_i - \mathbf{x}_j| < \varepsilon$ . As the number of points tends to infinity, the correlation integral, for small values of  $\varepsilon$ , will take the form

$$C(\varepsilon) \sim \varepsilon^\nu \quad \text{when } N \rightarrow \infty \quad (2.5)$$

where  $\nu$  is referred to as the correlation exponent. The correlation dimension (denoted by  $d$ ) is then defined as

$$d = \lim_{\varepsilon \rightarrow 0} \nu = \lim_{\varepsilon \rightarrow 0} \frac{\ln C(\varepsilon)}{\ln \varepsilon} \quad (2.6)$$

The correlation dimension  $d$  is a measure of the dimensionality of the space occupied by the random points.

Caputo et al. (1986) suggested that, the correlation dimension  $d$  of a system can be estimated as the saturated correlation exponent  $\nu$  in the plot of  $\ln C(\varepsilon)$  against  $\ln \varepsilon$ . If the correlation dimension increases without bound, the system is supposed to be stochastic. If the correlation dimension leads to a finite value, the system is thought to be



governed by deterministic dynamics. If the correlation dimension is small and non-integer, the system is governed by chaos.

Figure 2.2 displays the Fourier power spectrum of the Lorenz time series. As expected from a non-periodic signal, the spectrum is continuous with broadband. Figure 2.3 shows the results of correlation integral analysis to identify chaos. The lines with different colors represent the analysis results in different embedding dimensions  $m$  within a predefined range [1, 10]. In the saturation region, the correlation dimension satisfies  $2 < d < 3$ , which verifies that Lorenz model is governed by low dimensional chaos.

Although the chaos identification methods are widely accepted, some debates can be observed on the claims that certain phenomena are chaotic (Grassberger, 1986; Theiler, 1990; Lorenz, 1991; Pasternack, 1999). As stated by Kantz and Schreiber (2004), too much effort has been directed to investigate whether a system is chaotic or not, whereas chaos based techniques can be applied in situations where determinism cannot be established, even to analyze systems with stochastic behaviors. With this understanding, more emphasis should be placed on the other two phases, i.e. phase space reconstruction and time series prediction.

## **2.4 Phase Space Reconstruction**

Reconstruction of the phase space requires transforming the time series  $x_i$  into the phase space vectors  $\mathbf{x}_i$ , which involves the determination of the time delay  $\tau$  and the embedding dimension  $m$ .

#### 2.4.1 Time Delay $\tau$

To produce good phase space reconstruction, the time delay  $\tau$  must be large enough so that  $x_i$  and  $x_{i-\tau}$  are rather independent of each other, but not be so large that  $x_i$  and  $x_{i-\tau}$  are completely independent in a statistical sense. Fraser and Swinney (1986) suggested using the average mutual information (AMI) analysis to determine the optimal time delay  $\tau$ .

Defined by Gallager (1968), the mutual information between measurement  $a_i$  drawn from a set  $A = \{a_i\}$  and measurement  $b_j$  drawn from a set  $B = \{b_j\}$  is the amount learned by the measurement of  $a_i$  about the measurement of  $b_j$ . The mutual information can be expressed in bits as

$$\log_2 \left[ \frac{P_{AB}(a_i, b_j)}{P_A(a_i)P_B(b_j)} \right], \quad (2.7)$$

where  $P_{AB}(a, b)$  is the joint probability density for measurements  $A$  and  $B$ ,  $P_A(a)$  and  $P_B(b)$  are the individual probability densities for measurements  $A$  and  $B$ . If  $a_i$  is completely independent of  $b_j$ , then  $P_{AB}(a_i, b_j) = P_A(a_i)P_B(b_j)$ , the mutual information is zero as it should be. The average over all measurements of this information statistic,

known as the average mutual information between  $A$  measurements and  $B$  measurements, is

$$I_{AB} = \sum_{a_i, b_j} P_{AB}(a_i, b_j) \log_2 \left[ \frac{P_{AB}(a_i, b_j)}{P_A(a_i)P_B(b_j)} \right]. \quad (2.8)$$

Taking  $x_i$  as  $a_i$  in measurements  $A$  and  $x_{i-\tau}$  as  $b_j$  in measurements  $B$ , the average mutual information between  $x_i$  and  $x_{i-\tau}$ , i.e. the amount learned by  $x_i$  about  $x_{i-\tau}$ , is

$$I(\tau) = \sum_{x_i, x_{i-\tau}} P(x_i, x_{i-\tau}) \log_2 \left[ \frac{P(x_i, x_{i-\tau})}{P(x_i)P(x_{i-\tau})} \right]. \quad (2.9)$$

As shown in Equation (2.9), when the time delay  $\tau$  becomes large, the chaotic behavior of the signal makes  $x_i$  and  $x_{i-\tau}$  independent, i.e.  $P(x_i, x_{i-\tau}) \rightarrow P(x_i)P(x_{i-\tau})$ , the average mutual information  $I(\tau)$  will tend to zero. The  $\tau$  value at which the first minimum of the average mutual information  $I(\tau)$  occurs is suggested to be selected as the optimal time delay (Abarbanel, 1996).

Figure 2.4 shows the average mutual information of the Lorenz time series. The first minimum of  $I(\tau)$  is at  $\tau = 10$ , which is selected as the optimal time delay to reconstruct the phase space for Lorenz model.

### 2.4.2 Embedding Dimension $m$

In the reconstructed phase space, the nearest neighbors of  $\mathbf{x}_i$  may come to the neighborhood of  $\mathbf{x}_i$  either through dynamical origins or by projection from a lower dimension. In the latter case, the nearest neighbors are referred to as the false nearest

neighbors (FNN). Embedding dimension  $m$  is the lowest dimension which unfolds the state vectors so that none of the false nearest neighbors remains. Kennel et al. (1992) developed the false nearest neighbors method to determine the optimal embedding dimension  $m$ .

Denote one nearest neighbor of  $\mathbf{x}_i$  in the  $m$  dimensional phase space as  $\mathbf{x}_i^{NN} = (x_i^{NN}, x_{i-\tau}^{NN}, \dots, x_{i-(m-1)\tau}^{NN})$ , it can be easily established whether  $\mathbf{x}_i^{NN}$  is false or not by comparing the distance between  $\mathbf{x}_i$  and  $\mathbf{x}_i^{NN}$  in dimension  $m$  with the distance in dimension  $m+1$ . The square of the Euclidian distance between  $\mathbf{x}_i$  and  $\mathbf{x}_i^{NN}$  in dimension  $m$  is

$$R_i(m)^2 = \sum_{j=1}^m [x_{i-(j-1)\tau} - x_{i-(j-1)\tau}^{NN}]^2, \quad (2.10)$$

while in dimension  $m+1$  it is

$$\begin{aligned} R_i(m+1)^2 &= \sum_{j=1}^{m+1} [x_{i-(j-1)\tau} - x_{i-(j-1)\tau}^{NN}]^2 \\ &= R_i(m)^2 + |x_{i-m\tau} - x_{i-m\tau}^{NN}|^2 \end{aligned} \quad (2.11)$$

The distance between  $\mathbf{x}_i$  and  $\mathbf{x}_i^{NN}$  in dimension  $m+1$  relative to the distance in dimension  $m$  is

$$\sqrt{\frac{R_i(m+1)^2 - R_i(m)^2}{R_i(m)^2}} = \frac{|x_{i-m\tau} - x_{i-m\tau}^{NN}|}{R_i(m)}. \quad (2.12)$$

In practice, when this quantity is larger than some threshold, approximately 15, the nearest neighbor  $\mathbf{x}_i^{NN}$  will be declared false. When the phase space dimension increases,

the percentage of false nearest neighbors will decrease. The optimal embedding dimension  $m$  can be determined by selecting the one where the percentage of false nearest neighbors drops to zero.

As shown in Figure 2.5, for the Lorenz time series with time delay  $\tau=10$ , the percentage of false nearest neighbors drops to zero at  $m=3$ . This indicates that the Lorenz time series  $x_i$  can be reconstructed in a phase space of  $\mathbf{x}_i = (x_i, x_{i-10}, x_{i-2 \times 10})$ . This reconstructed phase space for Lorenz model is plotted in Figure 2.6, where a clear pattern is revealed compared to its random evolution as shown in Figure 2.1.

## 2.5 Time Series Prediction

Having reconstructed the phase space and revealed the underlying determinism, the time series prediction can be achieved using phase space vectors as surrogates. The basic idea is to set a functional relationship between the current state  $\mathbf{x}_i$  and the future state  $\mathbf{x}_{i+T}$  in the form

$$\mathbf{x}_{i+T} = f_T(\mathbf{x}_i), \quad (2.13)$$

where  $T$  is the lead time or prediction horizon, and  $f_T(\cdot)$  is the mapping function.

### 2.5.1 Local Model

Local model (LM), often referred to as local linear model, is an effective method of finding the optimal expression for the mapping function  $f_T(\cdot)$ , in which only the most similar trajectories from the past are used to make predictions for the future. Local model

was first brought up by Farmer and Sidorowich in 1987 (Farmer and Sidorowich, 1987), and was gradually improved with applications in different disciplines, e.g. Babovic and Keijzer 1999, Babovic et al. 2005, Sannasiraj et al. 2005.

A conceptual sketch of the local model approach is depicted in Figure 2.7. The steps involved can be described as follows,

- Embedding the time series  $x_i$  into a phase space  $\mathbf{x}_i$

As discussed in Section 2.4, this step requires the determination of time delay  $\tau$  and embedding dimension  $m$ .

- Finding  $k$  nearest neighbors in the phase space

To predict a future state  $\mathbf{x}_{t+T}$ , a Euclidean metric is imposed on the phase space to find the  $k$  nearest neighbors of the current state  $\mathbf{x}_t$ , denoted by  $\mathbf{x}_n$  ( $n = 1, 2, \dots, k$ ).

- Calculating the ‘expected’ future state

Having reconstructed the phase space and pooled the  $k$  nearest neighbors of the current state  $\mathbf{x}_t$ , the ‘expected’ vector of the future state  $\mathbf{x}_{t+T}$ , denoted as  $\hat{\mathbf{x}}_{t+T}$ , can be estimated through averaging as

$$\hat{\mathbf{x}}_{t+T} = \left( \sum_{n=1}^k \mathbf{x}_{n+T} \right) / k. \quad (2.14)$$

- Deriving the forecasted scalar value

In the phase space, the ‘expected’ future state  $\hat{\mathbf{x}}_{t+T}$  can be expressed in the form of Equation (2.3) as

$$\hat{\mathbf{x}}_{t+T} = \left( \hat{x}_{t+T}, \hat{x}_{t+T-\tau}, \dots, \hat{x}_{t+T-(m-1)\tau} \right). \quad (2.15)$$

The predicted scalar values  $\hat{x}_{t+T}$ ,  $\hat{x}_{t+T-\tau}$ , ... in the time series  $x_i$  can be retrieved according to the structure.

Local model shares fundamental ideas with Takens' time-delay embedding theorem, making it well suited for the prediction of chaotic time series. Although local model makes use of a linear approximation for each prediction, the resulting overall model can be highly nonlinear, as each of these linear approximations is made for each separate neighborhood.

### 2.5.2 Standard Approach

Besides the embedding parameters  $m$  and  $\tau$ , the number of nearest neighbors  $k$  also needs to be determined in local model. There are two approaches to determine these parameters, i.e. the standard approach and the inverse approach. The standard approach uses the theoretical sights as criteria, whereas the inverse approach solves a global optimization problem to determine these parameters.

In the standard approach, the embedding parameters  $m$  and  $\tau$  are respectively optimized through the false nearest neighbors and average mutual information analyses, as described in Section 2.4.1 and Section 2.4.2. The number of nearest neighbors  $k$  is calculated by following empirical formulae (Farmer and Sidorwich, 1987)

$$k = m + 1, \tag{2.16}$$

or

$$k = 2m + 1. \tag{2.17}$$

### 2.5.3 Inverse Approach

The standard approach has been shown to provide suboptimal choices of the local model parameters (Babovic et al., 2000; Liong et al., 2005). Babovic et al. (2000) proposed an alternative inverse approach, in which genetic algorithm (GA) acts as a search engine to simultaneously optimize these parameters. As defined by Holland (1975) and Goldberg (1989), genetic algorithm is a search technique based upon the mechanics of natural genetics, which combines Darwin's theory of evolution with a structured information exchange among chromosomes.

Figure 2.8 displays the flow diagram of inverse approach, with the evolving process in genetic algorithm illustrated in Figure 2.9. The steps in the inverse approach can be summarized as follows,

1. Initializing population

An initial population of chromosomes  $P_i = \{m, \tau, k\}$ , where  $m$ ,  $\tau$  and  $k$  are represented by binary bits, is randomly generated within the specified ranges of parameters.

2. Evaluating the fitness of each chromosome in the initial population

Fitness of each initial chromosome is evaluated in terms of the root mean square error (RMSE) produced by local model designed with corresponding parameters.

3. Evolving chromosomes through the following process until an entirely new population is generated



- Selection: The roulette wheel selection scheme is adopted to select the chromosomes to reproduce offspring according to their respective fitness. The chromosome with higher fitness has a better chance of being selected.
  - Crossover: Some portion of a pair of chromosomes selected from the population is exchanged according to some constraints in order to generate two new sets of parameters.
  - Mutation: One individual chromosome selected from the population is transformed to a new individual by inverting some of its binary values.
4. Evaluating the fitness of each chromosome in the new population and repeating 3 until stop criterion

Fitness of each new chromosome is evaluated, and the process of selection, crossover and mutation is repeated. With the hope that the fitter parents will create a better generation of children, successive generations are created until the user-defined threshold for the fitness or number of maximum generation is reached.

Neither Equation (2.9) nor Equation (2.12) is a function of prediction horizon  $T$ , the average mutual information and false nearest neighbors analyses will therefore result in the same set of  $m$  and  $\tau$  irrespective of  $T$ . In contrast, the inverse approach targets to achieve the best prediction accuracy, different  $m$ ,  $\tau$  and  $k$  will therefore be resulted from different  $T$ .

#### **2.5.4 Lorenz Time Series Prediction**

This section applies local model to predict the Lorenz time series, in which the parameters are determined by both the standard approach and the inverse approach. The data set is divided into two subsets, i.e. training data and testing data. Training data are used to determine the parameters, whereas the testing set is used to test the performance of local model.

As described in preceding sections, the set of parameters  $(m, \tau, k)$  for the Lorenz time series has been identified by the standard approach to be  $(3, 10, 4)$ . Figure 2.10 shows the predicted Lorenz time series when  $T = 2$ , together with the desired values and the residuals. The Lorenz time series are well predicted, reducing root mean square (RMS, quadratic mean) from 12.7033 to 0.4366.

The typical parameters in the inverse approach to predict the Lorenz time series when  $T = 2$  is summarized in Table 2.1. The search result for  $(m, \tau, k)$  is  $(4, 4, 2)$ , which differs from  $(3, 10, 4)$  determined by the standard approach. Figure 2.11 shows the predicted Lorenz time series when  $T = 2$ , together with the desired values and the residuals. Local model with parameters determined by the inverse approach successfully predicts the Lorenz time series, further reducing the root mean square of residuals to 0.2714.

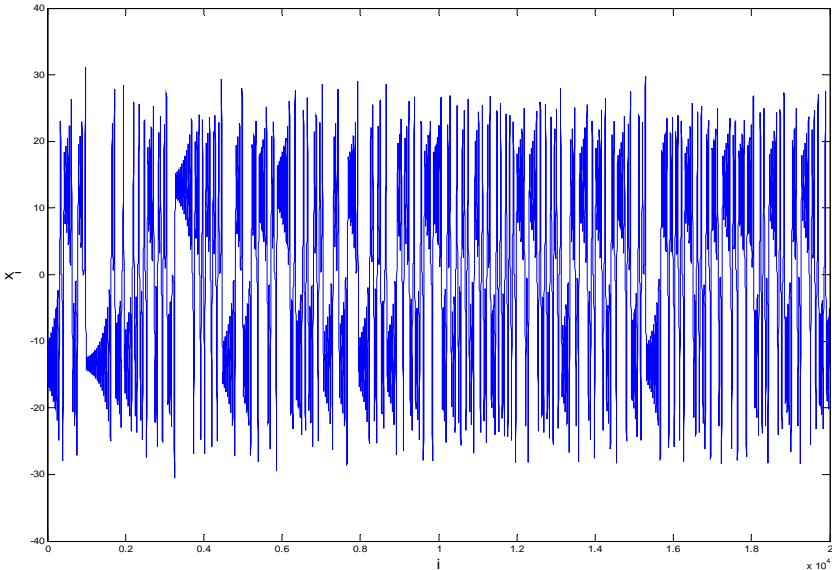


Figure 2.1 Lorenz time series.

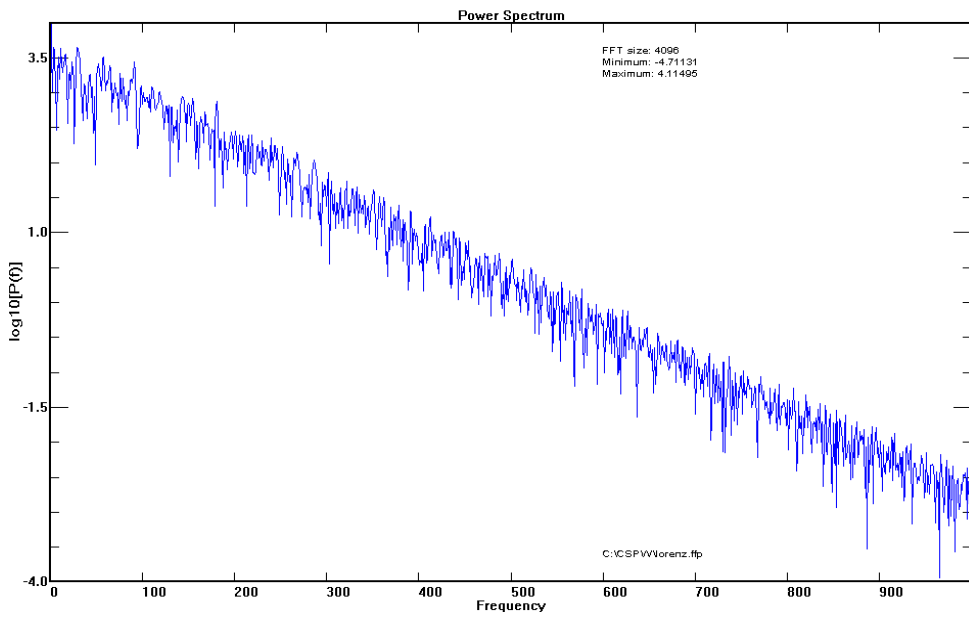


Figure 2.2 Fourier power spectrum of Lorenz time series.

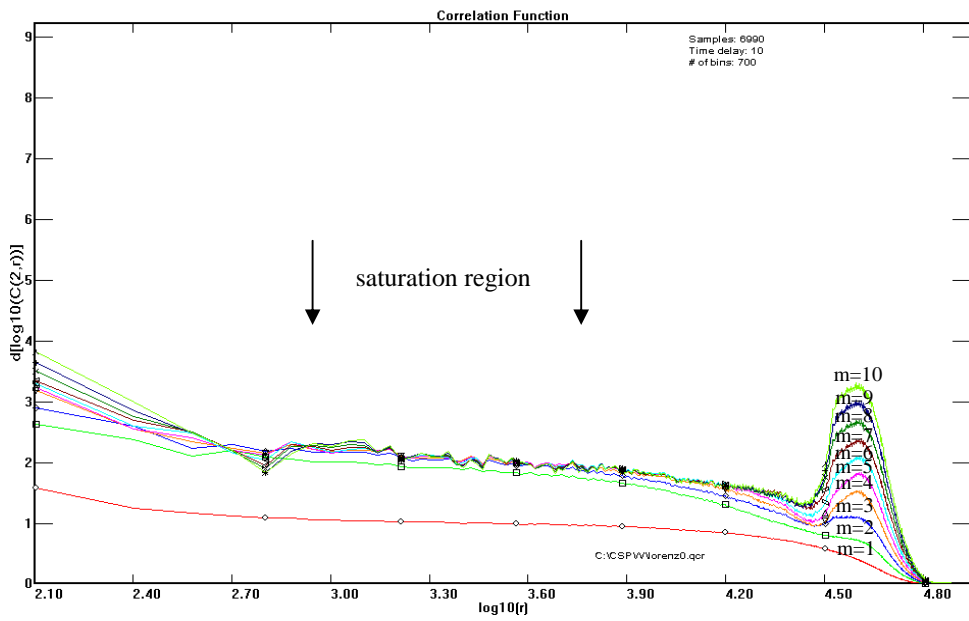


Figure 2.3 Correlation integral analysis for Lorenz time series (different colors represent different embedding dimensions).

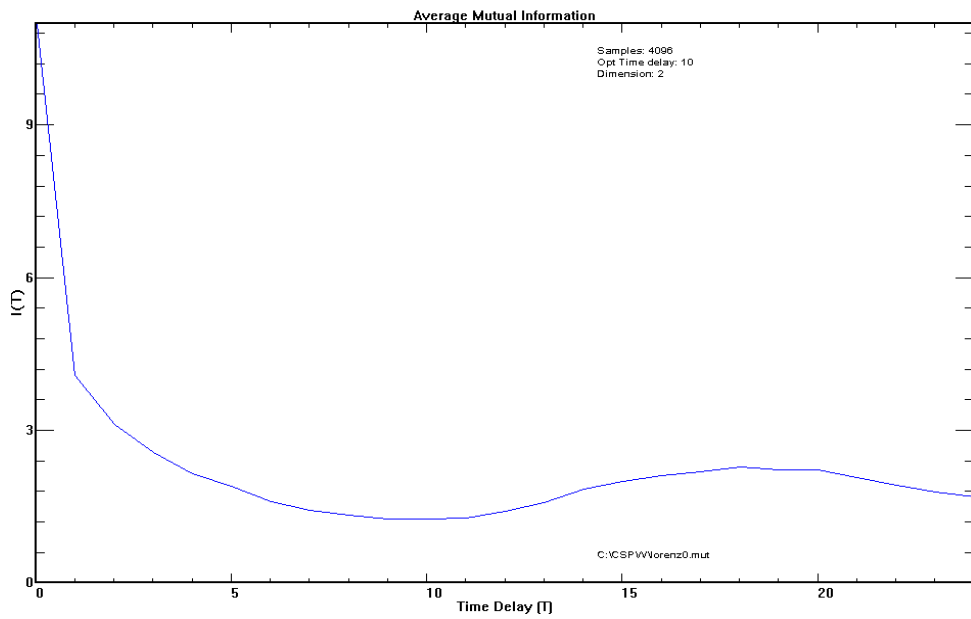


Figure 2.4 Average mutual information of Lorenz time series.

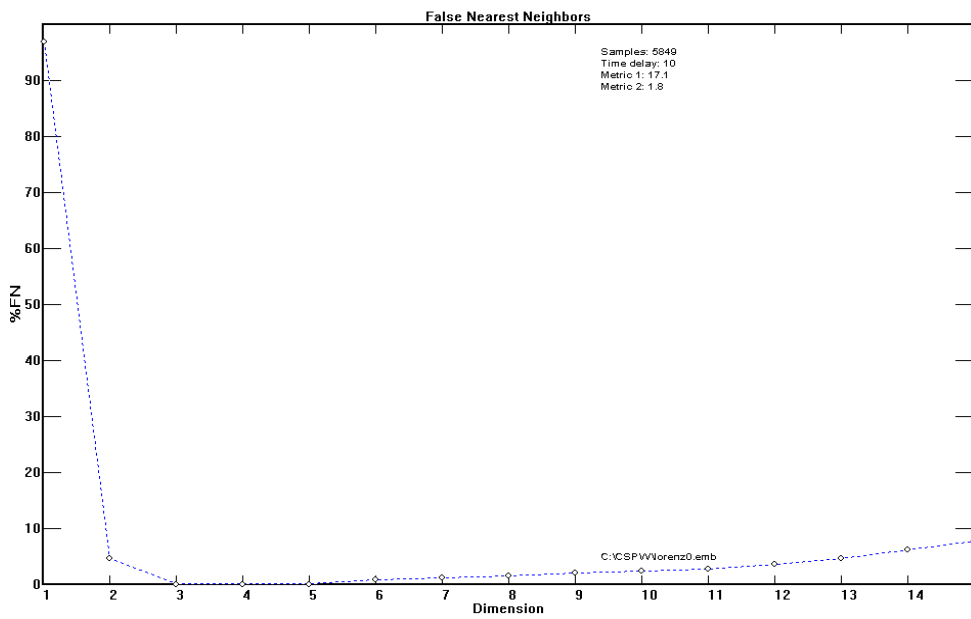


Figure 2.5 False nearest neighbors analysis for Lorenz time series.

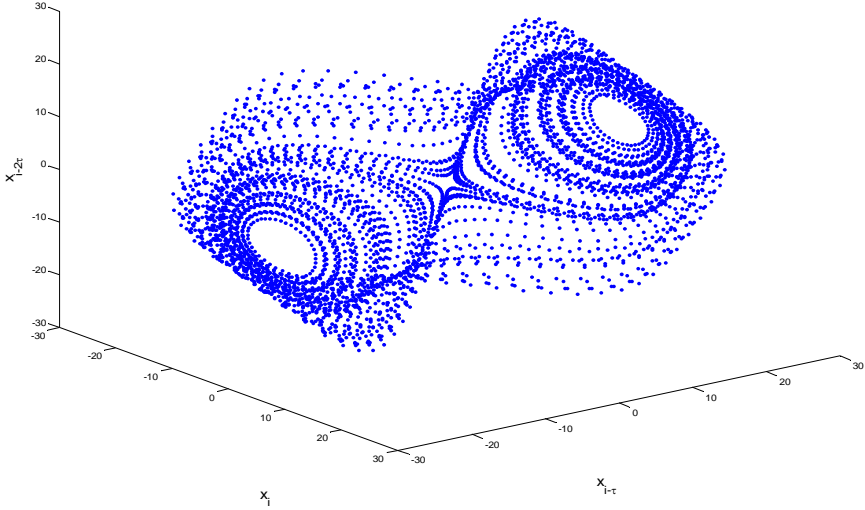


Figure 2.6 Reconstructed phase space for Lorenz model.

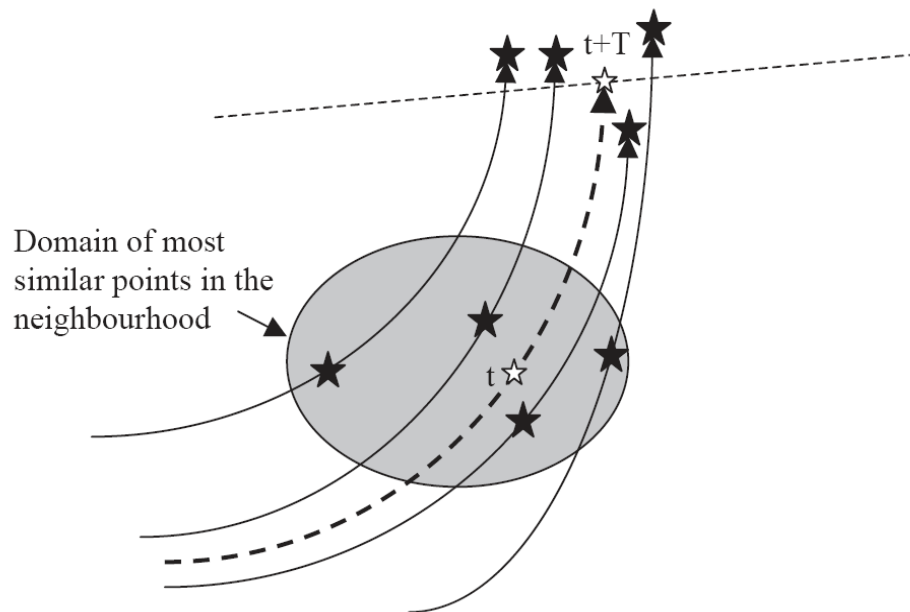


Figure 2.7 Conceptual sketch of the local model approach (the black stars are the nearest neighbors to the white star representing the predicted value).

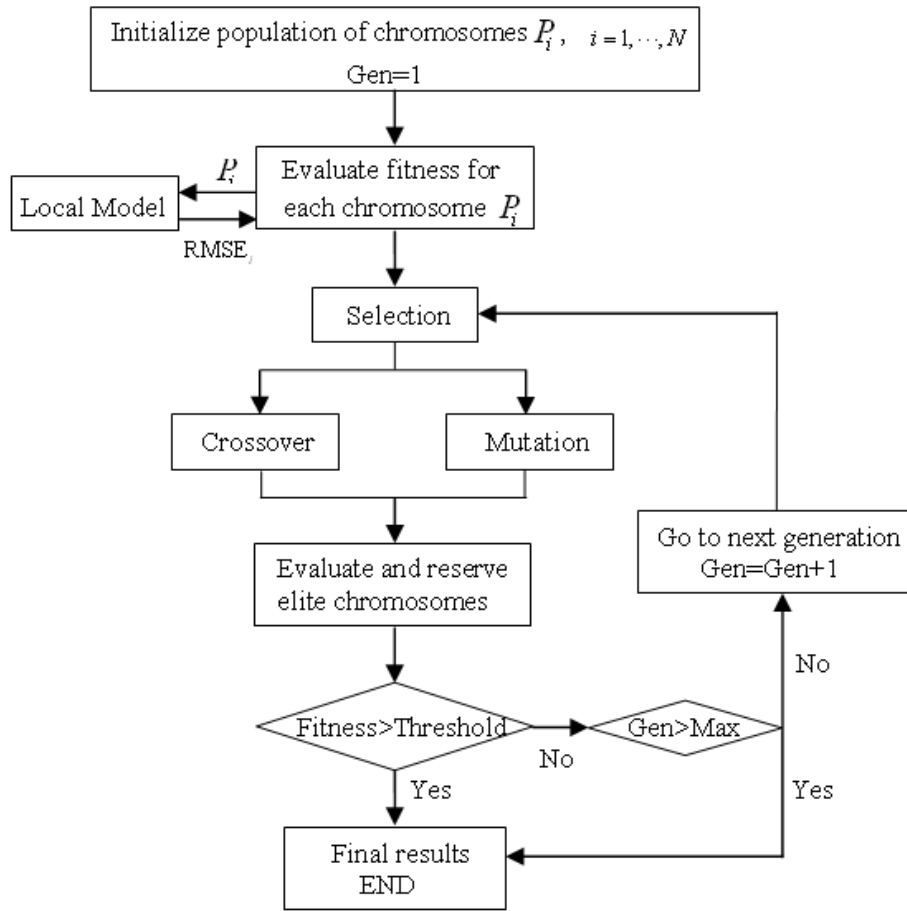


Figure 2.8 Flow diagram of genetic algorithm.

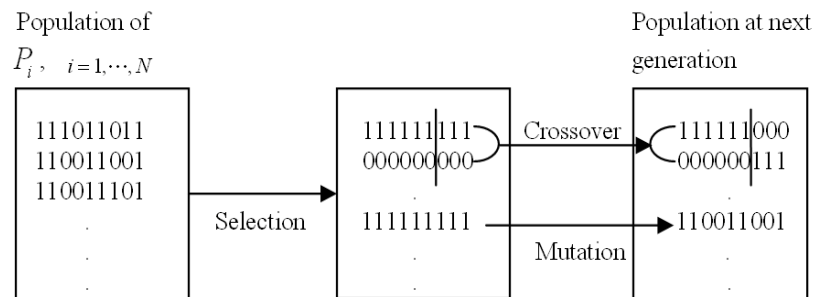


Figure 2.9 Schematic illustration of evolving process in genetic algorithm.



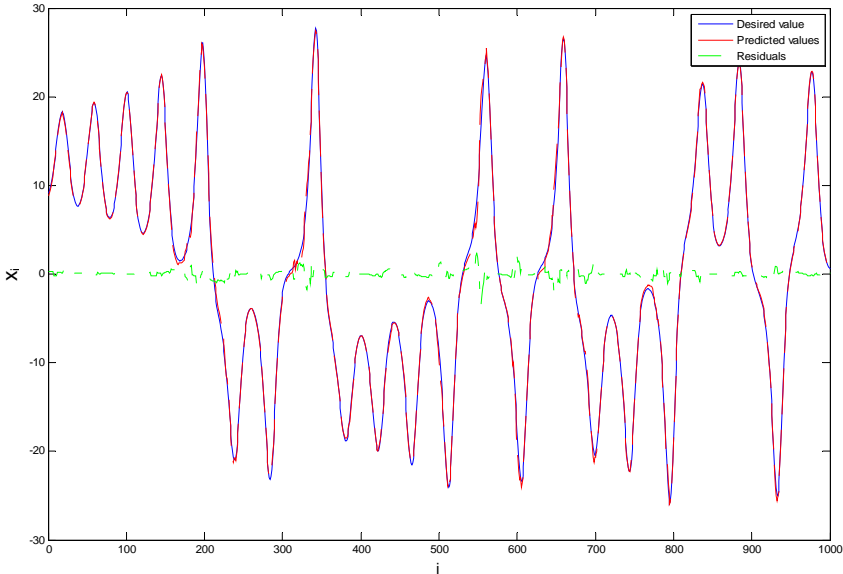


Figure 2.10 Lorenz time series prediction using local model (standard approach;  $T=2$ ).

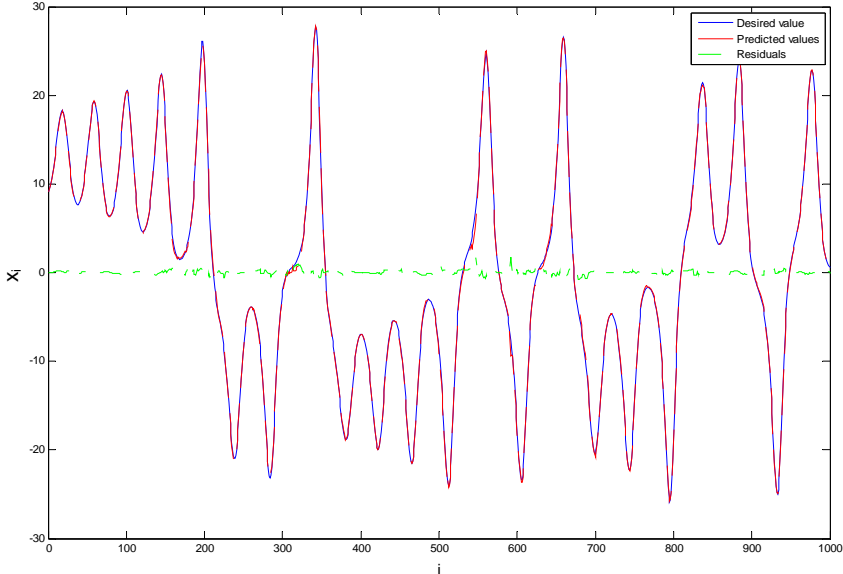


Figure 2.11 Lorenz time series prediction using local model (inverse approach;  $T=2$ ).

Table 2.1 Parameters in the inverse approach for Lorenz model.

Parameters	Value	Remark
T	2	prediction horizon
parmin	1, 1, 1	minimum values of $m$ , $\tau$ and $k$
parmax	6, 20, 50	maximum values of $m$ , $\tau$ and $k$
npopsiz	10	population size
maxgen	200	maximum number of generation
pcross	50	probability of crossover (%)
pmutate	2	probability of mutation (%)
rmsestop	0	stop RMSE

## **Chapter 3**

### **Artificial Neural Networks**

#### **3.1 Introduction**

The human brain is a highly complex, nonlinear, and parallel information processing system. It has the capability to perform certain computations many times faster than the fastest digital computer in existence. The recognition that the human brain computes in an entirely different way from the conventional digital computer has motivated the emergence and rapid development of artificial neural networks, commonly referred to as neural networks (McCulloch and Pitts, 1943; Rosenblatt, 1958; Minsky and Papert, 1969; Hopfield, 1982; Kohonen, 1982; Powell, 1985; Vapnik, 1995).

Haykin (1999) offered a deliberate definition as, artificial neural networks are massively parallel distributed processors made up of simple processing units, known as neurons, which have a natural propensity for storing experiential knowledge and making it available for use. Artificial neural networks resemble the human brain in two respects:

- Knowledge is acquired by the network from its environment through a learning process.

- Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

The purpose of this chapter is to provide basic concepts of artificial neural networks and variants pertaining to this study. For a complete discussion about artificial neural networks, please refer to Haykin (1999).

### 3.2 Neuron

Neurons are information processing units that are fundamental to the operation of artificial neural networks. As shown in the block diagram of Figure 3.1, a neuron is composed of 3 basic elements:

1. A set of synapses, each of which is characterized by a weight of its own.
2. An adder for summing the input signals, weighted by the respective synapses of the neuron.
3. An activation function for limiting the amplitude of the output of a neuron.

In mathematical terms, a neuron  $k$  can be described by following pair of equations,

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (3.1)$$

and

$$y_k = \varphi(u_k + b_k) \quad (3.2)$$

where  $x_1, x_2, \dots, x_m$  are the input signals,  $w_{k1}, w_{k2}, \dots, w_{km}$  are the synaptic weights of neuron  $k$ ,  $u_k$  is the linear combiner output due to the input signals,  $b_k$  is the bias,  $\varphi(\cdot)$  is the activation function, and  $y_k$  is the output signal of the neuron.

The bias  $b_k$  has the effect of applying an affine transformation to the output  $u_k$  of the linear combiner. Adding a new synapse with its input signal  $x_0 = +1$  and its synaptic weight  $w_{k0} = b_k$ , the model of neuron  $k$  can be reformulated as

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (3.3)$$

and

$$y_k = \varphi(v_k) \quad (3.4)$$

where  $v_k = u_k + b_k$  is the net input of the activation function, often referred to as the induced local field or activation potential.

### 3.3 Activation Function

The activation function, denoted by  $\varphi(\cdot)$ , defines the output of a neuron in terms of the activation potential  $v$ . Two basic types of activation functions can be identified as follows,

- Threshold Function (Hard Limiter)

$$\varphi(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \quad (3.5)$$

In engineering literature, this form of an activation function is commonly referred to as a Heaviside step function. Such a neuron is called the McCulloch-Pitts model, in recognition of the pioneering work done by McCulloch and Pitts.

- Sigmoid Function

The sigmoid function, whose graph is s-shaped, is by far the most common form of activation function used in the construction of artificial neural networks. The sigmoid function is defined as a strictly increasing function that exhibits a graceful balance between linear and nonlinear behavior. An example of the sigmoid function is the logistic function, defined by

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (3.6)$$

where  $a$  is the slope parameter of the sigmoid function. Compared to the threshold function, the sigmoid function is differential everywhere with a continuous range of values from 0 to 1.

### 3.4 Multilayer Perceptron

Single-layer perceptron is the simplest form of an artificial neural network developed as a pattern classifier by Rosenblatt in 1958 (Rosenblatt, 1958). As shown in Figure 3.2, single-layer perceptron is basically built around a single neuron, i.e. the McCulloch-Pitts model. Single-layer perceptron is capable of performing pattern classification with linearly separable classes. However, if the classes are not linearly separable, there cannot

be any single-layer perceptron that achieves the classification task (Minsky and Papert, 1969).

By introducing extra layers, multilayer perceptron successfully resolves the limitation of single-layer perceptron. Figure 3.3 shows the architecture of a multilayer perceptron with two hidden layers. Multilayer perceptron has three distinctive characteristics,

1. Multilayer perceptron is a feedforward artificial neural network, where the input signals propagate through the network in a forward direction on a layer-by-layer basis.
2. The network contains one or more layer of hidden neurons, which enable the network to learn complex tasks by extracting progressively more meaningful features from the input signals.
3. As opposed to the hard limiter used in Rosenblatt's perceptron, multilayer perceptron typically adopts a continuously differentiable nonlinear activation function for each neuron.

### 3.5 Back-propagation Algorithm

The back-propagation algorithm is a popular algorithm to train multilayer perceptron in a supervised manner (Werbos, 1974; Rumelhart and McClelland, 1986; Rumelhart et al., 1986). In the back-propagation algorithm, the synaptic weights of the network is adjusted according to the generalized delta rule

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n), \quad (3.7)$$

where

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n). \quad (3.8)$$

In Equation (3.8),  $\alpha$  is called the momentum constant,  $\eta$  is the learning rate parameter, and  $\delta_j(n)$  is the local gradient. The local gradient  $\delta_j(n)$  can be recursively calculated by

$$\delta_j(n) = e_j(n) \phi_j'(v_j(n)), \quad \text{for output neuron } j, \quad (3.9)$$

and

$$\delta_j(n) = \sum_{k \in C} \delta_k(n) w_{kj}(n) \phi_j'(v_j(n)), \quad \text{for hidden neuron } j. \quad (3.10)$$

The detailed description and derivation of the back-propagation algorithm are presented in Appendix A.

### 3.6 Application of Multilayer Perceptron

Multilayer perceptron is normally applied to two tasks, i.e. pattern classification and function approximation. Both error prediction and error distribution in this study fall within the latter category – function approximation. To predict the model errors, the mapping function that explicitly reveals the relationship in the time series is approximated, whereas error distribution is achieved by approximating the mapping function between the model outputs at different stations. This section will discuss the method implemented in error prediction, and error distribution will be discussed in Chapter 7.

#### 3.6.1 Network Architecture

As stated in Chapter 2, the general form of time series prediction can be formulated by



$$x_{t+T} = f(x_{t-a}, x_{t-b}, x_{t-c} \dots). \quad (3.11)$$

By feeding the scalar values  $x_{t-a}, x_{t-b}, x_{t-c} \dots$  as input to the network with the future value  $x_{t+T}$  being the desired output, multilayer perceptron can be applied to approximate the mapping function  $f(\cdot)$ .

An issue of time series prediction using multilayer perceptron lies in the selection of the input scalars. Experience, trial and error are two common methods, such as in Karunasinghe and Liong (2006), Chang et al. (2007) and Wang et al. (2007). Although these networks show acceptable results, the selection strategies are insufficiently supported from theoretical point of view. Inspired by the fact that the underlying structure of the time series can be better viewed in the reconstructed phase space, Equation (3.11) can be reformulated as

$$x_{t+T} = g_T(\mathbf{x}_t), \quad (3.12)$$

where  $\mathbf{x}_t = (x_t, x_{t-\tau}, \dots, x_{t-(m-1)\tau})$  is the current state vector, and  $g_T(\cdot)$  is an alternative mapping function. The components of the state vectors  $\mathbf{x}_t$  can be determined by the average mutual information and false nearest neighbors analyses in chaos theory.

Supported by the universal approximation theorem (Cybenko, 1989), essentially stating that a single hidden layer is sufficient for multilayer perceptron to compute a uniform approximation of any continuous function, multilayer perceptron used in study has only one hidden layer. The structures of input state vectors and output scalar in Equation (3.12) require  $m$  input neurons in the network, and one neuron in the output layer.

### 3.6.2 Lorenz Time Series Prediction

A network is said to generalize well when the input-output mapping computed by the network is correct for data never used in training the network. To reach the point of best generalization, cross validation data are required for multilayer perceptron to avoid overfitting. When the errors between the predicted values and the desired values in the cross validation data begin to increase, the training stops and this is considered to be the point of best generalization. Different from in local model, the data set is divided into three subsets when applying multilayer perceptron, i.e. training data, cross validation data and testing data.

Parameters for the networks and in the training algorithm are highly dependent on the nature of the problem, the complexity of the issue and the computational power. To predict the Lorenz time series, parameters are set as follows,

- No. of hidden neurons: 20
- No. of epochs: 100
- learning rate  $\eta = 0.1$
- momentum constant  $\alpha = 0.7$

Figure 3.4 shows the predicted Lorenz time series when  $T = 2$ , together with the desired values and the residuals. Multilayer perceptron almost perfectly predicts the Lorenz time series, with root mean square reduced to 0.2612, same order as in the GA-based local model.

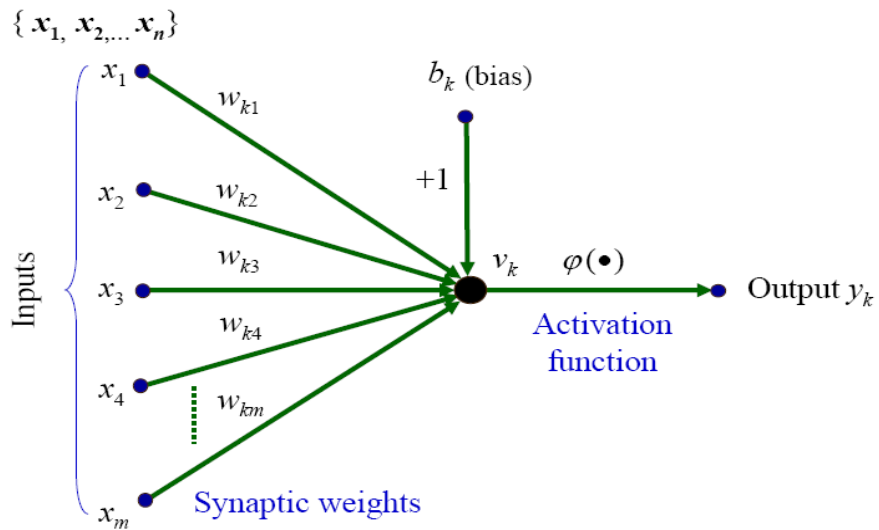


Figure 3.1 Nonlinear model of a neuron.

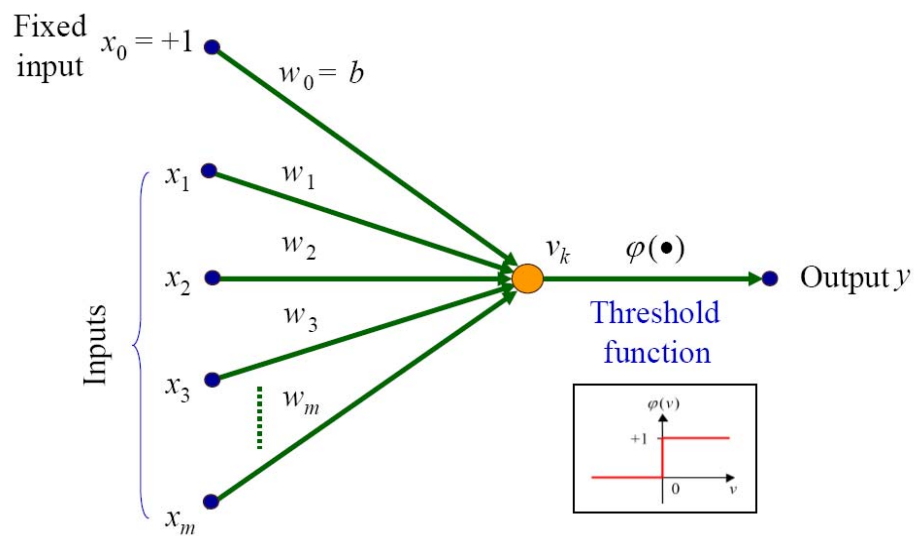


Figure 3.2 Model of a single-layer perceptron.

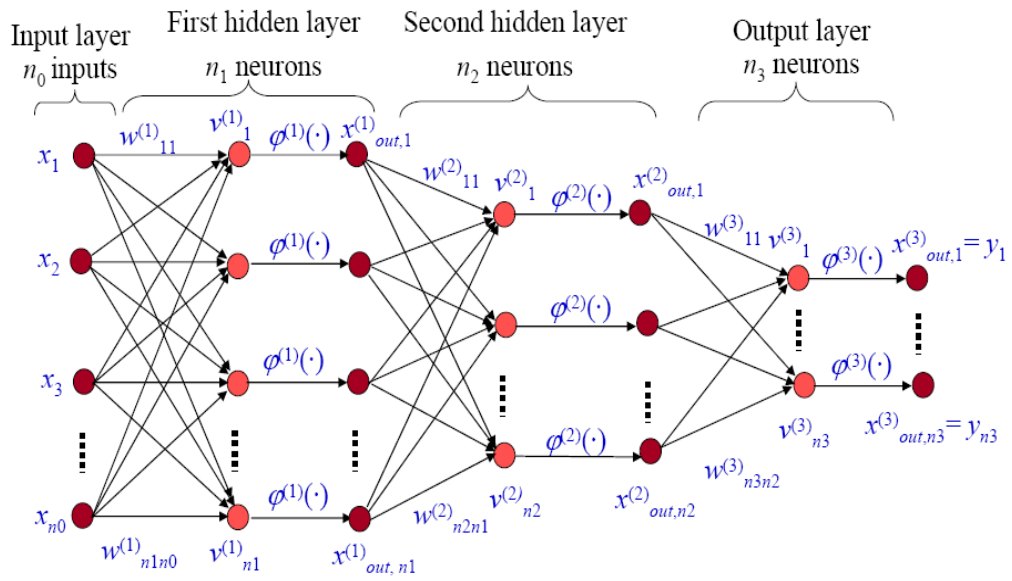


Figure 3.3 Architectural graph of a multilayer perceptron with two hidden layers.

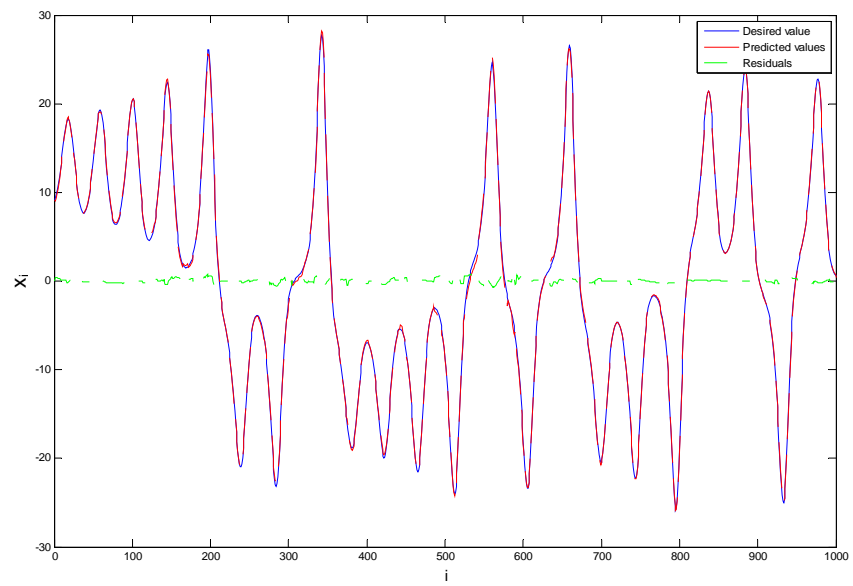


Figure 3.4 Lorenz time series prediction using multilayer perceptron (T=2).

## Chapter 4

### Kalman Filter

Kalman filter is essentially a recursive algorithm for estimating the state of a dynamic system. With the focus to address how Kalman filter is implemented in error distribution, this chapter will briefly describe three variants related to this study, i.e. the linear Kalman filter, the extended Kalman filter and the steady-state Kalman filter. For a complete description about the Kalman filter theory, please refer to Maybeck (1979).

#### 4.1 Linear Kalman Filter

Kalman filter is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the linear filtering problem (Kalman 1960). The originally proposed Kalman filter, referred to as the linear Kalman filter, deals with the linear dynamic system controlled by the coupled equations in the state-space form (Welch and Bishop, 2001)

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_{k-1}, \quad (4.1)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \quad (4.2)$$

Equation (4.1) describes the numerical scheme in a deterministic model, where  $\mathbf{x}_k \in R^n$  is the state vector,  $\mathbf{u}_k \in R^l$  is the forcing term,  $\mathbf{w}_{k-1} \in R^n$  represents the model noise, the  $n \times n$  matrix  $\mathbf{A}_k$  relates the state at the previous time step  $k-1$  to the state at the current step  $k$ , while the  $n \times l$  matrix  $\mathbf{B}_k$  relates the forcing term to the state. Equation (4.2) formulates the measurements that are available in the model domain, where  $\mathbf{z}_k \in R^m$  is the measurement vector,  $\mathbf{v}_k \in R^m$  represents the measurement noise, and the  $m \times n$  matrix  $\mathbf{H}_k$  relates the state to the measurement.

In the linear Kalman filter, the model noise  $\mathbf{w}_k$  and the measurement noise  $\mathbf{v}_k$  are assumed to be independent, and of normal probability distributions,

$$p(\mathbf{w}_k) \sim N(0, \mathbf{Q}_k), \quad (4.3)$$

$$p(\mathbf{v}_k) \sim N(0, \mathbf{R}_k), \quad (4.4)$$

where  $\mathbf{Q}_k = E[\mathbf{w}_k \cdot \mathbf{w}_k^T]$  and  $\mathbf{R}_k = E[\mathbf{v}_k \cdot \mathbf{v}_k^T]$  are respectively the model error covariance and the measurement error covariance.

Denote the forecast state estimate as  $\mathbf{x}_k^f \in R^n$  issued from the model, and the analysis state estimate as  $\mathbf{x}_k^a \in R^n$  in consideration of the measurement  $\mathbf{z}_k$ , the error covariances for the forecast estimate and the analysis estimate can be calculated by

$$\mathbf{P}_k^f = E[\mathbf{e}_k^f \cdot \mathbf{e}_k^{fT}], \quad (4.5)$$

$$\mathbf{P}_k^a = E[\mathbf{e}_k^a \cdot \mathbf{e}_k^{aT}], \quad (4.6)$$

where  $\mathbf{e}_k^f = \mathbf{x}_k - \mathbf{x}_k^f$ ,  $\mathbf{e}_k^a = \mathbf{x}_k - \mathbf{x}_k^a$  represent the forecast and analysis errors respectively.

The objective of linear Kalman filter is to find the analysis state estimate  $\mathbf{x}_k^a$  that yields the minimum analysis error covariance  $\mathbf{P}_k^a$ . The qualified analysis state estimate  $\mathbf{x}_k^a$  can be formulated as (Hayking, 2001)

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^f). \quad (4.7)$$

In Equation (4.7),  $\mathbf{H}_k \mathbf{x}_k^f$  indicates the forecast measurement, the difference  $\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^f$  is called the measurement innovation, and the  $n \times m$  matrix  $\mathbf{K}_k$ , referred to as the Kalman gain, serves as a weighting function between the model forecast  $\mathbf{x}_k^f$  and the measurement innovation  $\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^f$ .

Appendix B describes in detail the derivation of the linear Kalman filter algorithm. As shown in Figure 4.1, the linear Kalman filter algorithm can be summarized as follows

- At time step  $k = 0$ , initialize  $\mathbf{x}_k^a$  and  $\mathbf{P}_k^a$  as

$$\mathbf{x}_0^a = E[\mathbf{x}_0], \quad (4.8)$$

$$\mathbf{P}_0^a = E\left[(\mathbf{x}_0 - \mathbf{x}_0^a) \cdot (\mathbf{x}_0 - \mathbf{x}_0^a)^T\right]. \quad (4.9)$$

- At time step  $k = 1, 2, 3, \dots$ ,
  - Forecast step ('Predict'): Project forward the forecast state estimate and the forecast error covariance from time step  $k-1$  to  $k$ .

$$\mathbf{x}_k^f = \mathbf{A}_k \mathbf{x}_{k-1}^a + \mathbf{B}_k \mathbf{u}_k \quad (4.10)$$

$$\mathbf{P}_k^f = \mathbf{A}_k \mathbf{P}_{k-1}^a \mathbf{A}_k^T + \mathbf{Q}_k \quad (4.11)$$



- Analysis step ('Correct'): Compute the Kalman gain, incorporate a measurement into the model forecast to obtain an improved analysis state estimate, calculate the analysis error covariance.

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^T \left[ \mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{R}_k \right]^{-1} \quad (4.12)$$

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k \left( \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^f \right) \quad (4.13)$$

$$\mathbf{P}_k^a = \left( \mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^f \quad (4.14)$$

- $k = k + 1$ : Repeat the forecast step and the analysis step until the desired time step  $k$  is reached.

## 4.2 Extended Kalman Filter

The extended Kalman filter evolves from the linear Kalman filter through a statistical linearization procedure, dealing with the nonlinear system controlled by the coupled equations in the state-space form

$$\mathbf{x}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) + \mathbf{w}_{k-1}, \quad (4.15)$$

$$\mathbf{z}_k = h\left(\mathbf{x}_k\right) + \mathbf{v}_k, \quad (4.16)$$

where, in contrast with the linear system,  $f(\cdot)$  and  $h(\cdot)$  respectively denote a nonlinear model operator and a nonlinear measurement operator.

Analogous to the linear Kalman filter, the state vector and the measurement vector can be forecasted irrespective of the model and measurement noise

$$\mathbf{x}_k^f = f\left(\mathbf{x}_{k-1}^a, \mathbf{u}_k\right), \quad (4.17)$$

$$\mathbf{z}_k^f = h(\mathbf{x}_k^f), \quad (4.18)$$

where  $\mathbf{x}_k^f$ ,  $\mathbf{z}_k^f$  represent the forecast estimate of the state and measurement, while  $\mathbf{x}_{k-1}^a$  is the analysis state estimate in consideration of the measurement  $\mathbf{z}_{k-1}$ .

Based on Equations (4.17) and (4.18), Equations (4.15) and (4.16) can be linearized in the neighborhood of  $(\mathbf{x}_{k-1}^a, \mathbf{x}_k^f)$  and  $(\mathbf{x}_k^f, \mathbf{z}_k^f)$  using the first-order Taylor series expansion

$$\mathbf{x}_k \doteq \mathbf{x}_k^f + \mathbf{A}_k (\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^a) + \mathbf{w}_{k-1}, \quad (4.19)$$

$$\mathbf{z}_k \doteq \mathbf{z}_k^f + \mathbf{H}_k (\mathbf{x}_k - \mathbf{x}_k^f) + \mathbf{v}_k, \quad (4.20)$$

where  $\mathbf{A}_k$ ,  $\mathbf{H}_k$  are the Jacobian matrices in the forms of

$$\mathbf{A}_k = \left. \frac{\partial f(\cdot)}{\partial \mathbf{x}_{k-1}} \right|_{\mathbf{x}_{k-1} = \mathbf{x}_{k-1}^a}, \quad (4.21)$$

$$\mathbf{H}_k = \left. \frac{\partial h(\cdot)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \mathbf{x}_k^f}. \quad (4.22)$$

The  $ij$  th entry of  $\mathbf{A}_k$  is equal to the partial derivative of the  $i$  th component of  $f(\cdot)$  with respect to the  $j$  th component of  $\mathbf{x}_{k-1}$ .  $\mathbf{H}_k$  is constructed in like manner.

Notice Equations (4.21) and (4.22) closely resemble Equations (4.1) and (4.2), the linear Kalman filter algorithm can be applied to derive the extended Kalman filter algorithm. Figure 4.2 shows the extended Kalman filter algorithm, which compared to the linear Kalman filter algorithm is only different in the way to forecast the state vector and the measurement vector.

### 4.3 Steady-state Kalman Filter

In the classical Kalman filter algorithm, the propagation of the error covariance matrix normally requires huge computational power, which hampers the application of Kalman filter in high dimensional systems. Some sub-optimal Kalman filter approximations have been developed to solve this problem, such as the ensemble Kalman filter (Evensen, 1994; Madsen and Cañizares, 1999; Haugen and Evensen, 2002; Evensen, 2003), the reduced rank square root filter (Verlann and Heemink, 1997; Cañizares et al., 1998; Madsen and Cañizares, 1999), and the steady-state Kalman filter (Cañizares et al., 2001; Sørensen et al., 2001; Oke et al., 2002; Verlaan et al., 2005; Serafy and Mynett, 2008).

The ensemble Kalman filter is based on a Monte Carlo simulation approach for the propagation of error covariance. The reduced rank square root filter uses square root factorization to approximate the error covariance matrix by a reduced rank matrix, in which only the leading eigenvectors of the error covariance matrix are kept. A constant Kalman gain calculated off-line is used in the steady-state Kalman filter to circumvent the error covariance propagation, making the steady-state Kalman filter the most cost-efficient among the sub-optimal routines.

Cañizares et al. (2001) proposed the steady-state Kalman filter, realizing the fact that the Kalman gain  $\mathbf{K}_k$  shows little time variation. In the steady-state Kalman filter, the error covariance is propagated off-line, i.e.

$$\mathbf{P}_k^f = \mathbf{A}_k \mathbf{P}_{k-1}^a \mathbf{A}_k^T + \mathbf{Q}_k, \quad (4.23)$$

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^f, \quad (4.24)$$

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^T \left[ \mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{R}_k \right]^{-1}. \quad (4.25)$$

When the solution for Equations (4.23), (4.24) and (4.25) reaches a steady state, the constant Kalman gain  $\mathbf{K}$  is determined as the average of the Kalman gains  $\mathbf{K}_k$ . Once the Kalman gain  $\mathbf{K}$  is available, only the equations for the state forecast and analysis have to be solved on-line, i.e.

$$\mathbf{x}_k^f = \mathbf{A}_k \mathbf{x}_{k-1}^a + \mathbf{B}_k \mathbf{u}_k, \quad (4.26)$$

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K} (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^f). \quad (4.27)$$

On-line error propagation is not part of the steady-state Kalman filter implying that the computational costs are only slightly more expensive than a normal model run.

#### 4.4 Application of Kalman Filter in Error Distribution

Although Kalman filter computes the error covariance internally, it is a valid depiction of the true errors committed by the filter only to the extent that the filter's own model adequately portrays the true system behavior (Maybeck, 1979). In other words, Kalman filter will give an optimal estimate only if the specified error statistics exactly represent the true measurement and model errors. However, it is impossible to access the true error statistics in practice. As a prevailing way to solve this issue, the structure of the error

covariance can be explicitly modelled under the assumption of isotropy and homogeneity (Verlaan, 1998; Mitchell and Houtekamer 2000; Sørensen et al., 2001).

In the error distribution application of this study, the model error covariance  $\mathbf{Q}$  and the measurement error covariance  $\mathbf{R}$  are respectively formulated as

$$\mathbf{Q} = [Q_{i,j}] = \rho^{d_{ij}} \sigma_{mo}^2, \quad (4.28)$$

$$\mathbf{R} = \text{diag}[\sigma_{me1}^2, \dots, \sigma_{mep}^2], \quad (4.29)$$

where  $\rho$  is the spatial correlation for the model errors,  $d_{ij} = |i - j|/\sqrt{2}$  is the distance between element  $ij$  and diagonal in matrix  $\mathbf{Q}$ ,  $\sigma_{mo}$  is the standard deviation of the model errors,  $\sigma_{mei}$  ( $i = 1, 2, \dots, p$ ) is the standard deviation of the  $i$ th measurement error, and  $p$  is the total number of measurements. Parameters in Equations (4.28) and (4.29) are determined through an exhaustive search within a pre-defined range.

Once the model errors are predicted at the measurement stations, the measurements can be updated as a combination of the predicted errors and the model outputs. The updated measurements and the model outputs are then optimally merged weighted by the constant Kalman gain computed off-line. As a consequence, the predicted errors are distributed to the stations of interest, where the forecasting accuracy can also be improved.

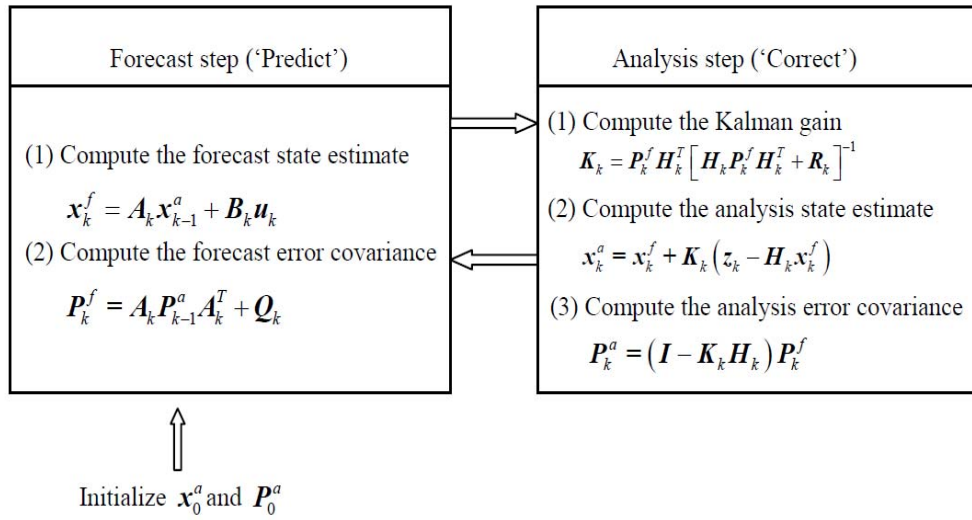


Figure 4.1 Linear Kalman filter algorithm.

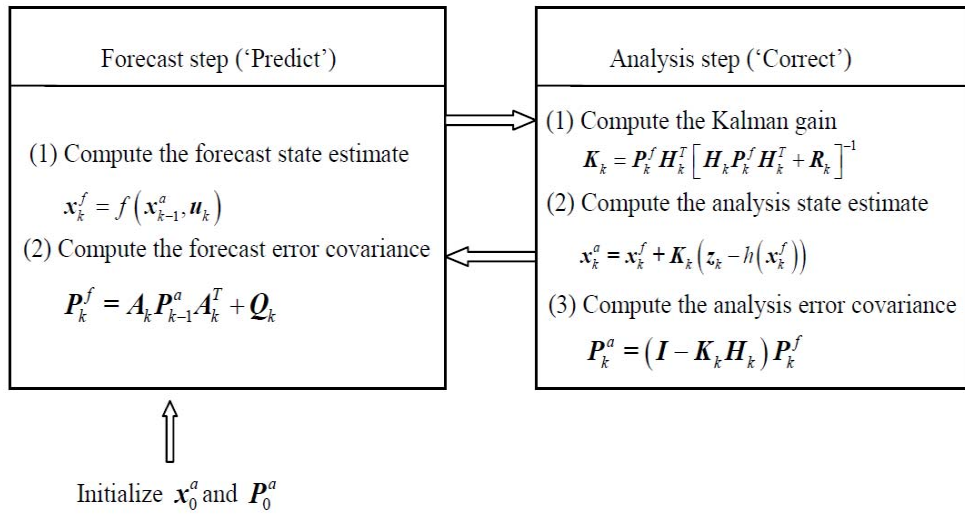


Figure 4.2 Extended Kalman filter algorithm.

## **Chapter 5**

### **Singapore Regional Model**

Singapore Strait is one of the busiest shipping routes in the world. Since the 1960s, the coastal area has been heavily utilized as ports or related industrial facilities with rapid economic development. With the intention to provide hydrodynamic information of the water surrounding Singapore for accurate scheduling of harbor facilities, docking and sailing times, the dedicated Singapore Regional Model (SRM) was developed within the Delft3D-FLOW modelling system in 2004 by WL | Delft Hydraulics, the Netherlands (Kernkamp and Zijl, 2004).

This chapter presents a general description of the modelling environment - Delft3D-FLOW, including governing equations and numerical approximations. The model set-up of the Singapore Regional Model is then discussed, followed by examples of the model output.

#### **5.1 Delft3D-FLOW**

##### **5.1.1 Introduction**

Delft3D is a fully integrated computer software suite for numerical computations in the fluvial, estuarine and coastal environments, developed by WL | Delft Hydraulics, the Netherlands (Deltares, 2009). Delft3D is composed of several modules, which can respectively carry out simulations of flows, waves, water quality, particle tracking, ecology, sediment transports and morphological development. These modules are grouped around a mutual interface and capable of interacting with one another.

Delft3D-FLOW is the core of Delft3D, providing hydrodynamic basis for the other modules. Delft3D-FLOW has been validated for modelling a wide range of flow conditions, such as turbulent flows in laboratory flumes, rapidly varying flows in rivers, wind driven flows in lakes and tidal flows in estuaries. The validation approach is based on the Guidelines for Validation Documents of the International Association for Hydraulic Research (IAHR Bulletin, 1994).

### 5.1.2 Governing Equations

Delft3D-FLOW solves the shallow water equations, which are derived from the principles of mass and momentum conservations under the shallow water and the Boussinesq assumptions. Formulated in the orthogonal curvilinear co-ordinates  $(\xi, \eta)$  in the horizontal direction and in the  $\sigma$  co-ordinate in the vertical direction, these governing equations can be formulated as

$$\frac{\partial \zeta}{\partial t} + \frac{1}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}} \frac{\partial \left[ (d + \zeta) U \sqrt{G_{\eta\eta}} \right]}{\partial \xi} + \frac{1}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}} \frac{\partial \left[ (d + \zeta) V \sqrt{G_{\eta\eta}} \right]}{\partial \eta} = Q, \quad (5.1)$$



$$\begin{aligned}
 & \frac{\partial u}{\partial t} + \frac{u}{\sqrt{G_{\xi\xi}}} \frac{\partial u}{\partial \xi} + \frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial u}{\partial \eta} + \frac{\omega}{d+\zeta} \frac{\partial u}{\partial \sigma} \\
 & + \frac{uv}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\xi\xi}}}{\partial \eta} - \frac{v^2}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\eta\eta}}}{\partial \xi} - fv \\
 & = -\frac{1}{\rho_0\sqrt{G_{\xi\xi}}} P_\xi + F_\xi + \frac{1}{(d+\zeta)^2} \frac{\partial}{\partial \sigma} \left( \nu_{mol} + \max(\nu_{3D}, \nu_V^{back}) \frac{\partial u}{\partial \sigma} \right) + M_\xi
 \end{aligned} \tag{5.2}$$

$$\begin{aligned}
 & \frac{\partial v}{\partial t} + \frac{u}{\sqrt{G_{\xi\xi}}} \frac{\partial v}{\partial \xi} + \frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial v}{\partial \eta} + \frac{\omega}{d+\zeta} \frac{\partial v}{\partial \sigma} \\
 & + \frac{uv}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\eta\eta}}}{\partial \xi} - \frac{u^2}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\xi\xi}}}{\partial \eta} + fu \\
 & = -\frac{1}{\rho_0\sqrt{G_{\eta\eta}}} P_\eta + F_\eta + \frac{1}{(d+\zeta)^2} \frac{\partial}{\partial \sigma} \left( \nu_{mol} + \max(\nu_{3D}, \nu_V^{back}) \frac{\partial v}{\partial \sigma} \right) + M_\eta
 \end{aligned} \tag{5.3}$$

where,

- $\zeta$  - free surface elevation
- $t$  - time
- $\sqrt{G_{\xi\xi}}, \sqrt{G_{\eta\eta}}$  - co-ordinates transforming coefficients
- $d$  - water depth
- $U, V$  - depth-averaged velocities in  $\xi$  and  $\eta$  directions
- $Q$  - global source/sink per unit area
- $u, v, \omega$  - flow velocities in  $x, y$ , and  $\sigma$  directions
- $f$  - Coriolis coefficient
- $\rho_0$  - reference water density
- $P_\xi, P_\eta$  - hydrostatic pressure gradients

$F_\xi, F_\eta$  - turbulent momentum fluxes

$U_{mol}, U_{3D}, U_V^{back}$  - kinematic viscosity, eddy viscosity and background vertical viscosity

$M_\xi, M_\eta$  - sources/sinks of momentum

Noting that  $\omega$  is the vertical velocity relative to the moving  $\sigma$  plane, the vertical flow velocity  $w$  in the Cartesian  $z$  co-ordinate system can be calculated by

$$w = \omega + \frac{1}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}} \left[ u\sqrt{G_{\eta\eta}} \left( \sigma \frac{\partial H}{\partial \xi} + \frac{\partial \zeta}{\partial \xi} \right) + v\sqrt{G_{\xi\xi}} \left( \sigma \frac{\partial H}{\partial \eta} + \frac{\partial \zeta}{\partial \eta} \right) \right] + \left( \sigma \frac{\partial H}{\partial t} + \frac{\partial \zeta}{\partial t} \right), \quad (5.4)$$

where  $H = d + \zeta$  is the total water depth.

To make the mathematical problem well-posed, the governing equations are supplemented by appropriate boundary conditions. At the closed boundaries, such as river banks and coast lines, the boundary condition is specified as

$$v_\perp = 0, \quad (5.5)$$

which means no inflow or outflow can pass through the closed boundaries. At the open boundaries, following types of boundary conditions can be prescribed

- Water level:  $\zeta = F_\zeta(t)$ ,
- Velocity:  $U = F_U(t)$ ,
- Discharge:  $Q = F_Q(t)$ ,
- Riemann invariant:  $U \pm \zeta \sqrt{\frac{g}{d}} = F_R(t)$ .

The data needed for the open boundary conditions can be obtained from measurements, tide tables, or a larger model which encloses the present model.

### 5.1.3 Numerical Aspects

The numerical method of Delft3D-FLOW is based on finite differences. As shown in Figure 5.1, the variables describing the flows are arranged in a staggered grid system. The water level points are defined in the centre of a cell, the depth points are defined at the corners of a cell, while the velocity components are defined to be perpendicular to the grid cell faces. Staggered grid has several advantages, such as (Stelling, 1983)

- Boundary conditions can be implemented in a rather simple way;
- Staggered grid can achieve better accuracy compared to non-staggered grid;
- Staggered grid prevents spatial oscillations in the water levels.

Delft3D-FLOW adopts the Alternating Direction Implicit (ADI) method for temporal integration. The Alternating Direction Implicit method was introduced by Leendertse (1967, 1971, 1973) and extended by Stelling (1983). As a computationally efficient finite difference method, the Alternating Direction Implicit method splits one time step into two stages, which can be formulated in vector form as

Stage 1:

$$\frac{\vec{U}^{\ell+\frac{1}{2}} - \vec{U}^{\ell}}{\frac{1}{2}\Delta t} + \frac{1}{2}A_x\vec{U}^{\ell+\frac{1}{2}} + \frac{1}{2}A_y\vec{U}^{\ell} + B\vec{U}^{\ell+\frac{1}{2}} = \vec{d}, \quad (5.6)$$

Stage 2:

$$\frac{\vec{U}^{\ell+1} - \vec{U}^{\ell+\frac{1}{2}}}{\frac{1}{2}\Delta t} + \frac{1}{2}A_x\vec{U}^{\ell+1} + \frac{1}{2}A_y\vec{U}^{\ell+\frac{1}{2}} + B\vec{U}^{\ell+1} = \vec{d}, \quad (5.7)$$

With:

$$A_x = \begin{bmatrix} 0 & -f & g\frac{\partial}{\partial x} \\ 0 & u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} & 0 \\ H\frac{\partial}{\partial x} & 0 & u\frac{\partial}{\partial x} \end{bmatrix}, \quad (5.8)$$

$$A_y = \begin{bmatrix} u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} & 0 & 0 \\ f & 0 & g\frac{\partial}{\partial y} \\ 0 & H\frac{\partial}{\partial x} & v\frac{\partial}{\partial y} \end{bmatrix}, \quad (5.9)$$

and

$$B = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}, \quad (5.10)$$

where  $\lambda$  is the linearized bottom friction coefficient, and  $\vec{d}$  denotes the external forces like wind and atmospheric pressure.

In stage 1, the  $v$  momentum equation, Equation (5.3), is solved first explicitly, thus the  $v$  velocity components are available for the cross terms in the  $u$  momentum equation, Equation (5.2). The  $u$  momentum equation is then coupled with the continuity equation, Equation (5.1), and solved implicitly. Similar procedure is performed in stage 2, but first

for the  $u$  momentum equation explicitly, followed by the  $v$  momentum equation and the continuity equation implicitly. For a complete time step, each separate term of the equations is still a second-order consistent approximation to the differential equations.

As the Alternating Direction Implicit method is essentially an implicit scheme, stability is not an essential issue in most cases for Delft3D-FLOW. However, the Alternating Direction Implicit method may lead to inaccurately predicted flow patterns due to the ADI-effect that is introduced by splitting the spatial operator in two directions. The accuracy is dependent on the Courant number defined by

$$Cr = \frac{\Delta t \cdot \sqrt{gH}}{\{\Delta x, \Delta y\}}, \quad (5.11)$$

where  $\Delta t$  is the time step,  $g$  is the acceleration of gravity,  $H$  is the water depth, and  $\{\Delta x, \Delta y\}$  is the minimal value of the grid spacing in either direction. In practical situations, the Courant number should not exceed a value of 10.

Further details about Delft3D-FLOW can be found in Deltares (2009).

## 5.2 Singapore Regional Model

### 5.2.1 Model Set-up

Figure 5.2 shows the extent, grid and bathymetry of the Singapore Regional Model. In order to compute the residual currents in Singapore Strait, large parts of the seas around Singapore are included in the model domain. Open boundaries are located in the Andaman Sea, in the South China Sea, and in the Java Sea. The Singapore Regional Model grid consists of around 38,500 curvilinear grid cells in the horizontal plane. Grid

sizes vary from about 200 m to 300 m around Singapore up to over 15 km at the open boundaries. The bathymetry in the Singapore Regional Model is based on Admiralty charts. The maximum depth of the model is about 2000 m in the Andaman Sea, whereas the maximum depth in Singapore Strait is over 160 m.

Singapore is located between two large water bodies: the South China Sea on the east and the Andaman Sea on the west. The water motion in Singapore Strait is driven by tides coming from both sides, by mean sea level differences between seas and by the wind. Therefore, the hydrodynamics of Singapore water is complex.

The Singapore Regional Model has been intensively calibrated with following parameters determined (Kernkamp and Zijl, 2004)

- Time step: 4 min
- Initial water level: 0 m
- Gravity:  $9.81 \text{ m/s}^2$
- Water density:  $1023 \text{ kg/m}^3$
- Bottom roughness: Manning  $0.022 \text{ s/m}^{1/3}$
- Wall roughness: Free slip
- Horizontal eddy viscosity:  $1.00 \text{ m}^2/\text{s}$

The values for these parameters were initially set based on the specialist knowledge of the modellers, and then fine-tuned through a sensitivity analysis process.

### **5.2.2 Numerical Simulation**

Figure 5.3 illustrates the 6 measurement stations studied, which are located at Jurong (1.31 N, 103.72 E), Tanjong Pagar (1.26 N, 103.85 E), Bukom (1.23 N, 103.78 E), Raffles (1.16 N, 103.74 E), Sembawang (1.47 N and 103.84 E) and Horsburgh (1.33 N, 104.41 E). In order to take account of the seasonal variation in the model outputs, the numerical simulation covers a period of one and a half years from 1<sup>st</sup> January 2000 00:00 to 31<sup>st</sup> June 2001 23:00, producing time series of 13,128 hourly data for all grid points. The corresponding observed tidal levels at the measurement stations are provided by the Singapore Marine and Port Authority (MPA).

As summarized in Table 5.1, the Singapore Regional Model outputs are compared with the observations at the 6 measurement stations, in terms of root mean square error (RMSE) and correlation coefficient ( $r$ ) defined as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\eta_i - \eta_i')^2} \quad (5.12)$$

$$r = \frac{\sum (\eta_i - \bar{\eta}_i)(\eta_i' - \bar{\eta}_i')}{\sqrt{\sum (\eta_i - \bar{\eta}_i)^2 \sum (\eta_i' - \bar{\eta}_i')^2}} \quad (5.13)$$

$$\bar{\eta}_i = \frac{\sum \eta_i}{N}; \bar{\eta}_i' = \frac{\sum \eta_i'}{N} \quad (5.14)$$

where  $N$  is the length of the time series,  $\eta_i$  represent the observed values, and  $\eta_i'$  are the Singapore Regional Model outputs. When tides propagate from the deep ocean to the shallow water, both spatial and temporal characteristics tend to transform due to the bottom friction from the sea bed. Attributed to the shoaling effect, the Singapore

Regional Model is less competent to capture the complex tidal movement in the coastal area, and hence produces larger model errors at the near shore measurement stations.

Examples of the model outputs at Jurong and Horsburgh are plotted in Figures 5.4 and 5.5, accompanied by the observations and the model errors. The discrepancies between the model outputs and the observations can be noticed especially at the tidal level extrema. To get a direct perception of the model errors, Figures 5.6 and 5.7 demonstrate independently the model errors at Jurong and Horsburgh. The model errors oscillate drastically with obvious random behaviors.



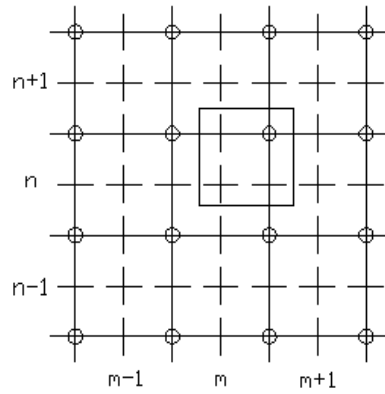




Figure 5.1 Staggered grid of Delft3D-FLOW.

Legend:

- Full lines      the numerical grid
- +              water level point ( $\zeta$ )
- o              depth point ( $d$ )
- —              horizontal velocity component
- |              vertical velocity component
-       variables with identical index ( $m, n$ )
-       a continuity cell / a computational control volume

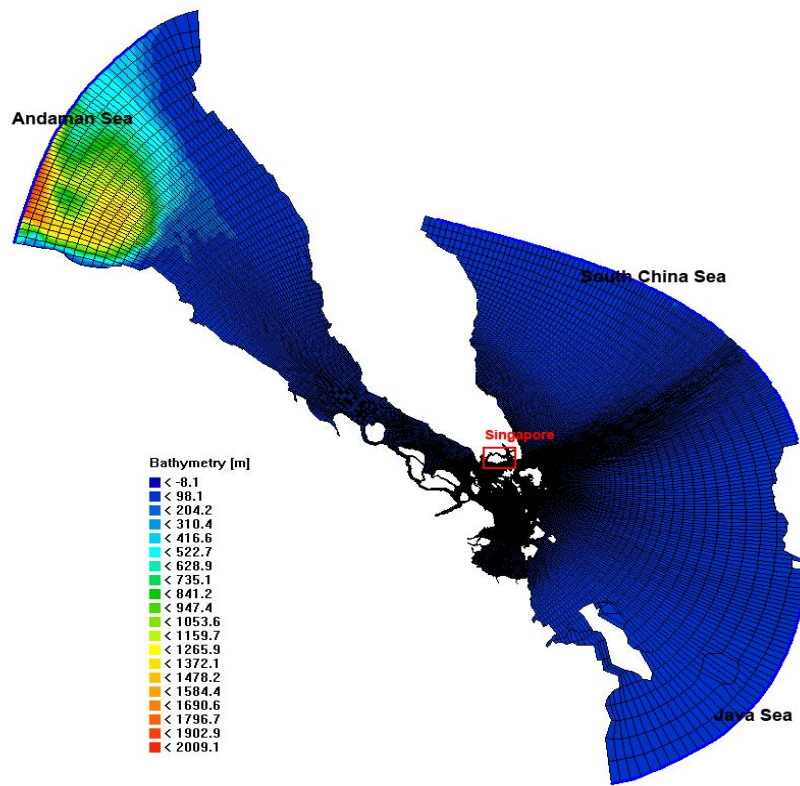


Figure 5.2 Extent, grid and bathymetry of Singapore Regional Model.

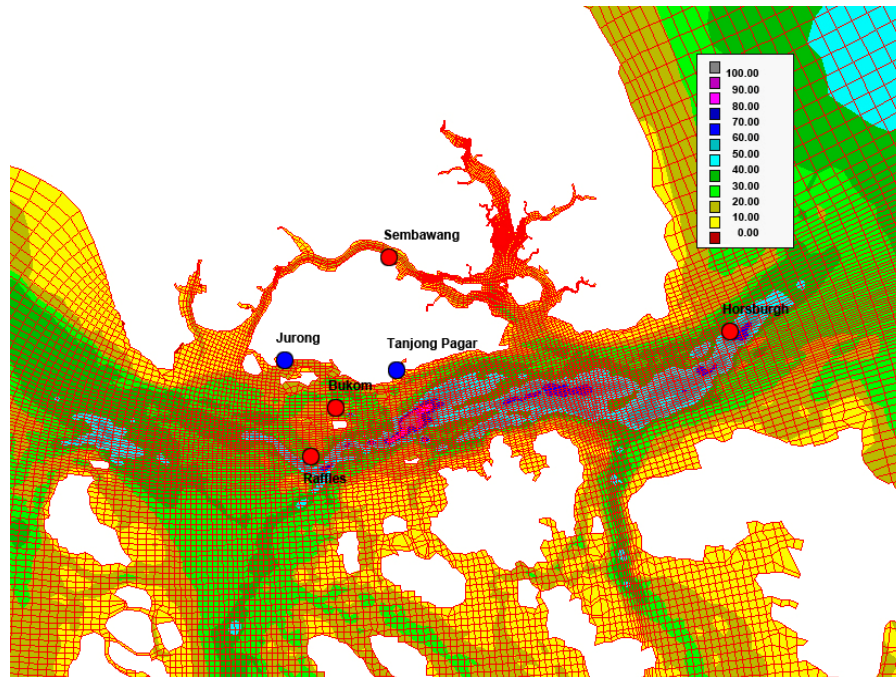


Figure 5.3 Measurement stations around Singapore.

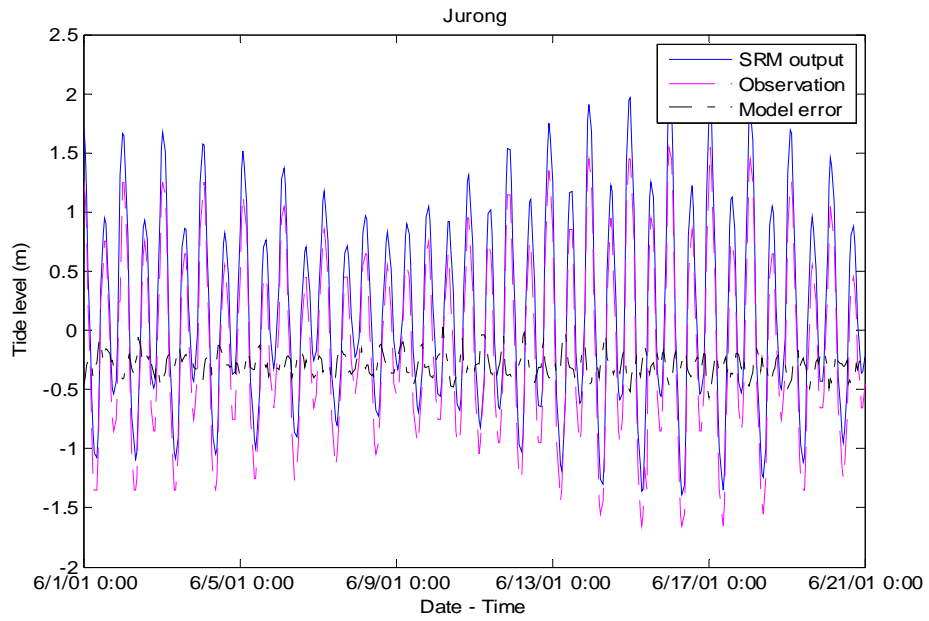


Figure 5.4 SRM outputs, observations and model errors at Jurong.

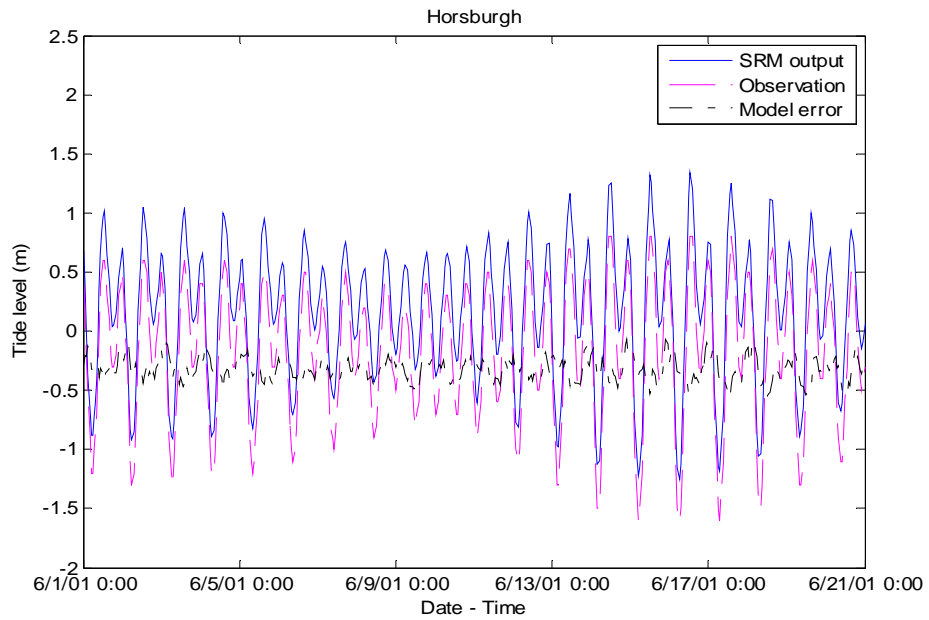


Figure 5.5 SRM outputs, observations and model errors at Horsburgh.

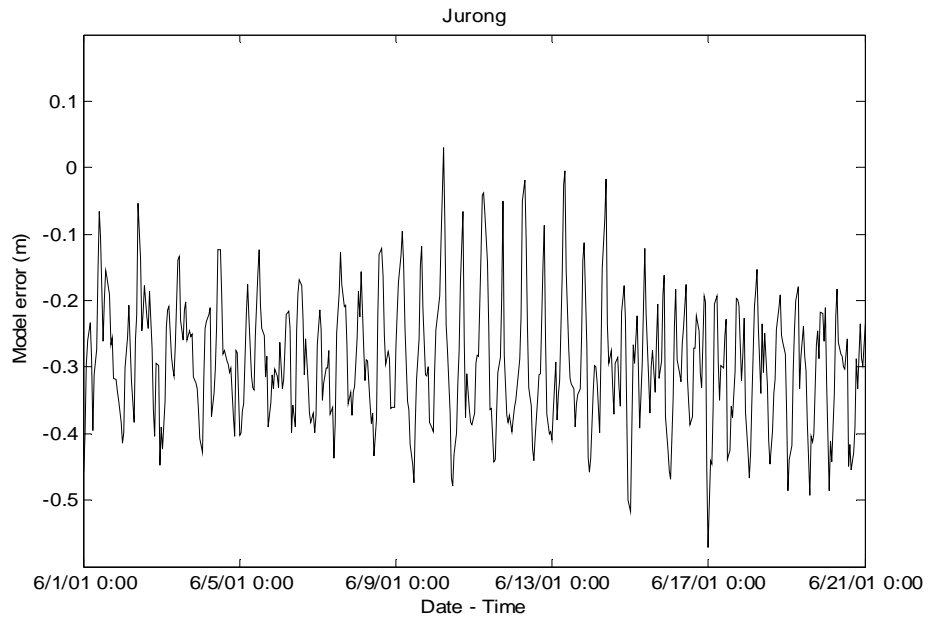


Figure 5.6 Model errors at Jurong.

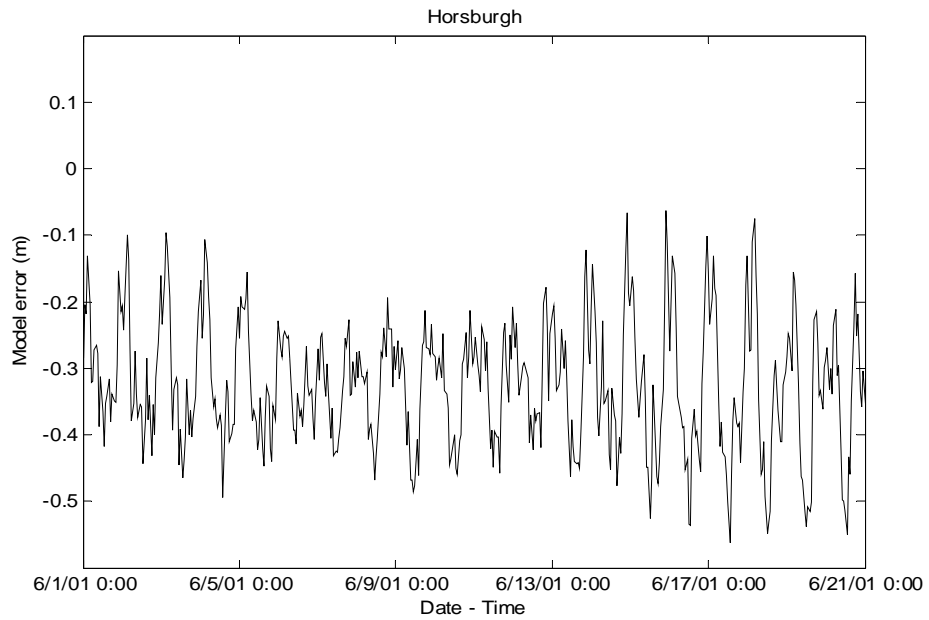


Figure 5.7 Model errors at Horsburgh.

Table 5.1 Statistics of model errors at the measurement stations.

	Jurong	Tanjong Pagar	Bukom	Raffles	Sembawang	Horsburgh
RMSE (cm)	18.80	18.57	16.54	16.16	19.40	13.91
r	0.91	0.91	0.91	0.91	0.89	0.93

## **Chapter 6**

### **Error Prediction with Local Model and Multilayer Perceptron**

#### **6.1 Introduction**

As discussed in previous chapters, Singapore Regional Model produces model errors with significant magnitudes. These model errors come from several sources,

- Insufficient tidal constituents prescribed at the open boundaries,
- Neglect of wind,
- Inaccuracy in the numerical discretization scheme,
- Uncertainties in the model parameters,
- Nonlinearities caused by the shoaling effect.

In order to compensate for these limitations and hence improve the forecast accuracy of the numerical model, this chapter applies both local model and multilayer perceptron to predict the model errors at the measurement stations. The model errors are forecasted to 5 prediction horizons ranging from 2 hours to 96 hours. The methodologies and results are discussed in detail, followed by a performance comparison between these two methods.

## 6.2 Application of Local Model in Error Prediction

### 6.2.1 Chaos Identification

Although the model errors exhibit distinct random behaviors, whether these behaviors arise from chaotic or stochastic dynamics remains undetermined. Correlation integral analysis is then conducted to identify chaos.

Using the model errors at Jurong as an example, Figure 6.1 illustrates the results of correlation integral analysis. The lines with different colors represent the analysis results in different embedding dimensions  $m$  within a predefined range  $[1, 10]$ . In the saturation region, the correlation dimension  $d$  falls within 1 to 4. This verifies the model error time series is governed by low dimensional chaos, and validates the applicability of predicting the model errors in the chaotic dynamic system. Same conclusions can be drawn for other measurement stations.

### 6.2.2 Parameter Determination

The first 120 data points (1<sup>st</sup> January 0:00 – 5<sup>th</sup> January 23:00) are discarded in order to eliminate the effect of initial conditions. The model error data sets are divided into two subsets, i.e. training data (from 121 to 8784; 6<sup>th</sup> January 0:00 – 31<sup>st</sup> December 23:00) and testing data (from 8785 to 13128; 1<sup>st</sup> January 0:00 – 30<sup>st</sup> June 23:00). Training data are used to determine the optimal  $m$ ,  $\tau$  and  $k$ , whereas the testing set is used to test the performance of local model.

The inverse approach is applied on the training data to simultaneously optimize  $m$ ,  $\tau$  and  $k$ . Table 6.1 summarizes the parameter settings in genetic algorithm, verified to be



sufficient for the algorithm to converge. The optimized parameters  $m$ ,  $\tau$  and  $k$  are summarized in Table 6.2. Because the higher complexity of the chaotic systems progressing along with the prediction horizons requires a higher dimension to unfold the underlying structures, the embedding dimension  $m$  increases when the prediction horizon  $T$  increases. Attributed to the extreme sensitivity of the chaotic systems to the initial conditions, the optimized parameters are unique for each prediction horizon at different measurement stations.

The reconstructed phase spaces are only visualizable when the embedding dimension  $m \leq 3$ . An example of the reconstructed phase spaces at Jurong is plotted in Figure 6.2. Compared to the original chaotic model error time series, clear patterns are revealed in the phase spaces. Trends in the trajectories of the state vectors establish the basis to predict the model errors.

### 6.2.3 Results

Figures 6.3 – 6.6 illustrate examples at Jurong and Horsburgh of error prediction with local model when prediction horizon  $T = 2$  hours and  $T = 96$  hours. The 2-hour local model forecast agrees well with the model errors, whereas the discrepancies become larger when  $T = 96$  hours. When the prediction horizon increases, the local model approach becomes less competent to capture the trajectories of the state vectors in the higher-dimensional phase spaces. However, even when  $T = 96$  hours, the local model forecast still successfully resolves the rising and falling tendencies of the model errors, resulting in residual errors with reduced magnitudes.

By integrating the predicted model errors into the model outputs, the forecasting accuracy of the Singapore Regional Model can be improved. Figures 6.7 – 6.8 plot the scatter diagrams at Jurong of the model outputs and the local model corrected outputs when  $T = 2$  hours. The Singapore Regional Model tends to over-predict the reality, while the scatter is significantly reduced in the local model corrected outputs.

Table 6.3 summarizes the error prediction efficiency of local model through evaluating the respective residual root mean square error and the correlation coefficient after error correction. For the prediction horizon  $T = 2$  hours, almost 80% of the root mean square errors have been removed from the Singapore Regional Model outputs. When  $T = 96$  hours, local model forecast removes about 50% of the root mean square errors. Averaged over 5 prediction horizons, the error reduction is about 60%, and the correlation coefficient between the corrected model outputs and the observations is enhanced to 0.98.

## **6.3 Application of Multilayer Perceptron in Error Prediction**

### **6.3.1 Methodology**

Cross validation data are required for multilayer perceptron to avoid overfitting. Therefore, the model error data sets are divided into three subsets, i.e. training data (from 121 to 8784; 6<sup>th</sup> January 0:00 – 31<sup>st</sup> December 23:00), cross validation data (from 8785 to 10944; 1<sup>st</sup> January 0:00 – 31<sup>st</sup> March 23:00) and testing data (from 10945 to 13128; 1<sup>st</sup> April 0:00 – 30<sup>st</sup> June 23:00).

The average mutual information and the false nearest neighbors analyses are then applied on the training data for the optimization of the embedding parameters  $m$  and  $\tau$ . Analyses examples at Jurong are plotted in Figures 6.9 and 6.10. The optimal time delay  $\tau$  is selected at the first minimum of the average mutual information, whereas the optimal embedding dimension  $m$  is determined when the percentage of false nearest neighbors drops to zero. The optimized parameters  $m$  and  $\tau$  are summarized in Table 6.4. Compared to the parameters in local model, identical set of embedding parameters is utilized in multilayer perceptron to reconstruct the phase spaces for different prediction horizons.

To approximate the mapping function  $g_T(\cdot)$  in Equation (3.12), i.e.  $x_{t+T} = g_T(\mathbf{x}_t)$ , the components of  $\mathbf{x}_t$  are fed as input to the multilayer perceptron with  $x_{t+T}$  being the desired response. Therefore, as shown in Figure 6.11, multilayer perceptron is composed of an input layer with  $m$  input neurons, a hidden layer, and an output layer with one output neuron. The logistic function is used as the activation function in the hidden layer, and the linear function is used in the output layer.

Fine-tuned by trial and error and in consideration of both learning rate and stability, parameters for the networks and in the training algorithm are set as follows,

- No. of hidden neurons: 50
- No. of epochs: 200
- learning rate  $\eta = 0.1$
- momentum constant  $\alpha = 0.7$

The prediction accuracy improves when increasing no. of hidden neurons and epochs, but deteriorates at certain critical values due to the generalization effect.

Multilayer perceptron tends to converge to local optima if the synaptic weights are not properly initialized. A set of 5 multilayer perceptrons with different initial synaptic weights is trained. The multilayer perceptron with the best prediction accuracy in the cross validation data set is selected as the optimal network, which is applied to predict the testing data set.

### 6.3.2 Results

Figures 6.12 and 6.13 illustrate examples of error prediction with multilayer perceptron at Jurong when prediction horizon  $T = 2$  hours and  $T = 96$  hours. Both the 2-hour and 96-hour multilayer perceptron forecast successfully resolves the oscillating trend of the model error time series, while the magnitudes of the residual errors increase slightly when  $T = 96$  hours.

The error prediction efficiency of multilayer perceptron is summarized in Table 6.5 through evaluating the respective residual root mean square error and the correlation coefficient after error correction. For the prediction horizon  $T = 2$  hours, more than 70% of the root mean square errors have been removed from the Singapore Regional Model outputs, while multilayer perceptron still removes over 50% of the root mean square errors when  $T = 96$  hours. Averaged over 5 prediction horizons, the error reduction is about 65%, and the correlation coefficient between the corrected model outputs and the observations is enhanced to 0.98.

## 6.4 Comparison between Local Model and Multilayer Perceptron

Figures 6.14 and 6.15 plot the residual root mean square error resulted from both local model and multilayer perceptron against the prediction horizon at Jurong and Horsburgh, together with the root mean square error before error correction. When the prediction horizons are small, the chaotic system exhibits low dimensional dynamical behavior, and local model is capable of capturing the dynamics in a better way than multilayer perceptron. When the prediction horizon increases, the local model approach becomes less competent to capture the trajectories of the state vectors in the higher-dimensional phase spaces, making the prediction accuracy of the local model approach deteriorate relatively drastically when  $T$  increases. Comparatively, multilayer perceptron is less sensitive to the prediction horizon with a more consistent performance. Averaged over 5 different prediction horizons, both methods are able to remove about 60% of the root mean square errors in the model error time series, while the overall performance of multilayer perceptron is slightly better.

In terms of computational cost, local model is more efficient than multilayer perceptron in model error prediction. Coded in FORTRAN and running in an Intel Core Due T2400 1.83 GHz machine with 2.5 GB RAM, it only takes less than 2 hours for local model to analyze one model error time series, including parameter determination and time series prediction. Multilayer perceptron is implemented using MATLAB Neural Networks Toolbox and running in the same machine. Without considering the time to optimize the embedding parameters, it takes about 20 hours to train a network. The

computational time increases significantly along with the data set size, the number of hidden neurons and epochs. However, as it is once-and-for-all to train a network, multilayer perceptron is still practical in real applications.

In the present study, the parameters in local model are determined by the inverse approach from searching in a sufficient scope, while the parameters in multilayer perceptron are determined by average mutual information and false nearest neighbors analyses. The fact that multilayer perceptron with sub-optimal parameters outperforms local model implies that multilayer perceptron is preferable to predict the model errors, especially in case of large prediction horizons.

CHAPTER 6. ERROR PREDICTION WITH LOCAL MODEL  
AND MULTILAYER PERCEPTRON

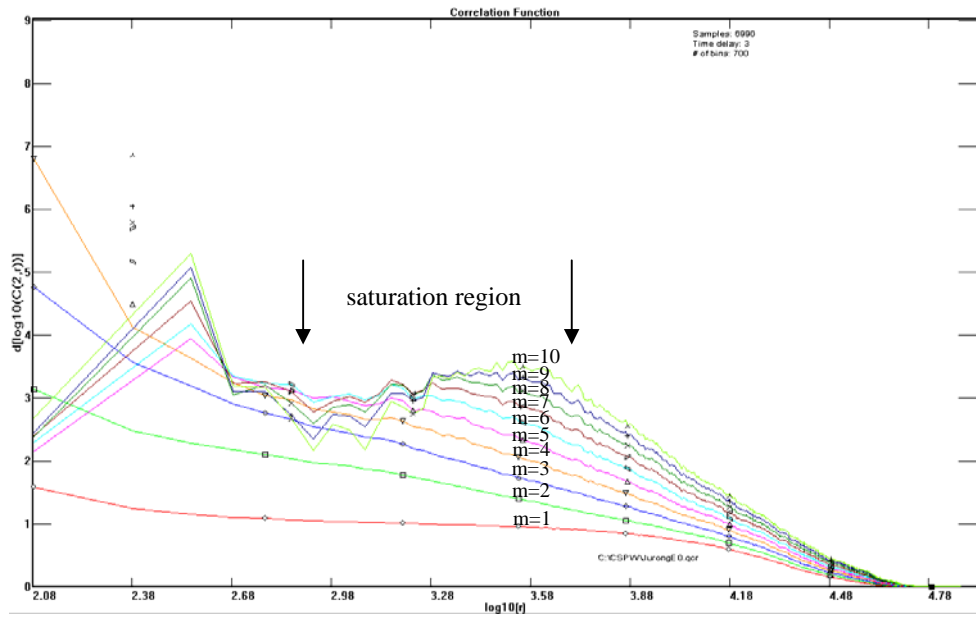


Figure 6.1 Correlation integral analysis for the model error time series at Jurong (different colors represent different embedding dimensions).

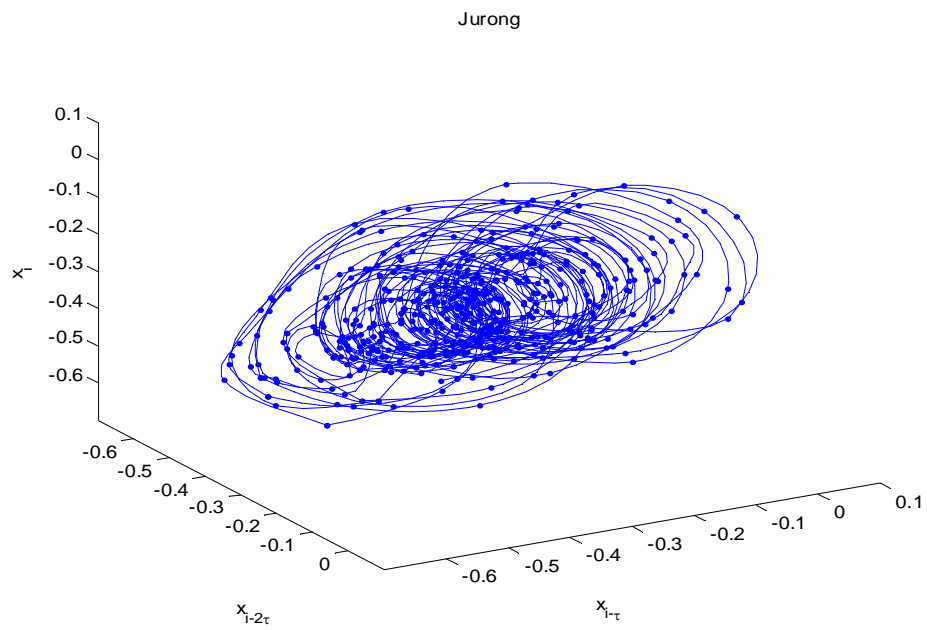


Figure 6.2 Reconstructed phase space for the model errors at Jurong ( $T=2$  hours).



CHAPTER 6. ERROR PREDICTION WITH LOCAL MODEL  
AND MULTILAYER PERCEPTRON

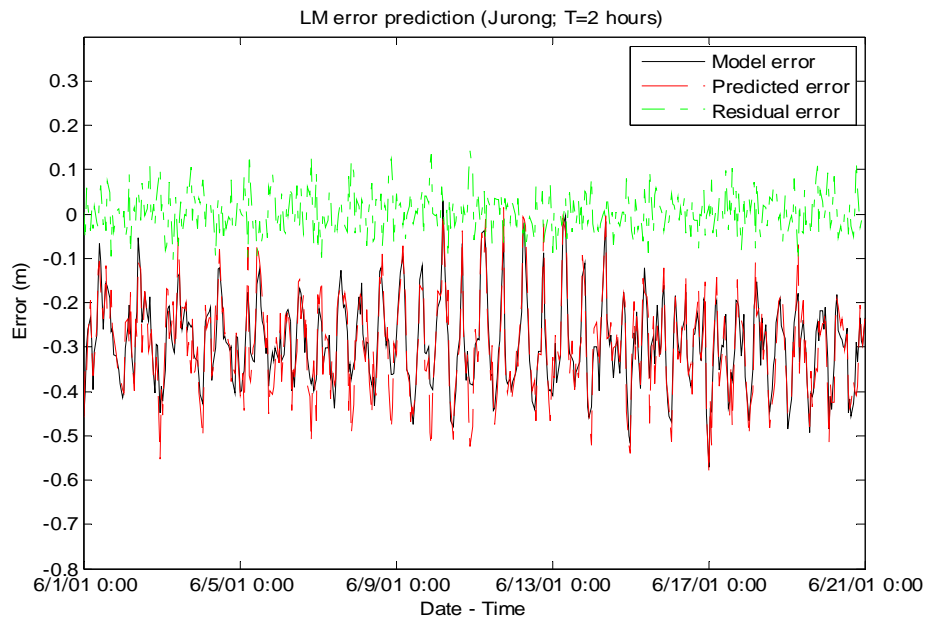


Figure 6.3 Error prediction with local model at Jurong (T=2 hours).

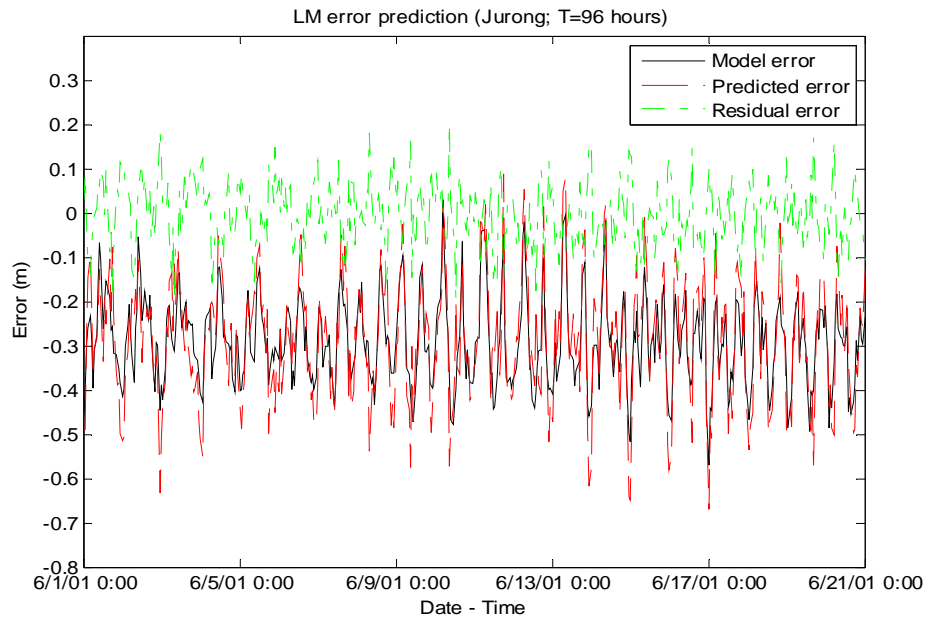


Figure 6.4 Error prediction with local model at Jurong (T=96 hours).

CHAPTER 6. ERROR PREDICTION WITH LOCAL MODEL  
AND MULTILAYER PERCEPTRON

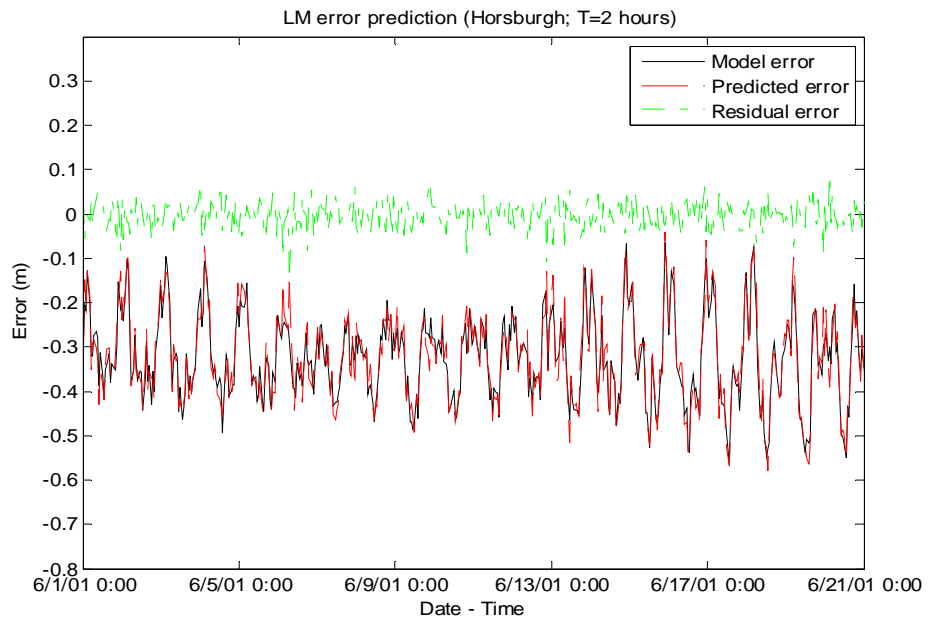


Figure 6.5 Error prediction with local model at Horsburgh (T=2 hours).

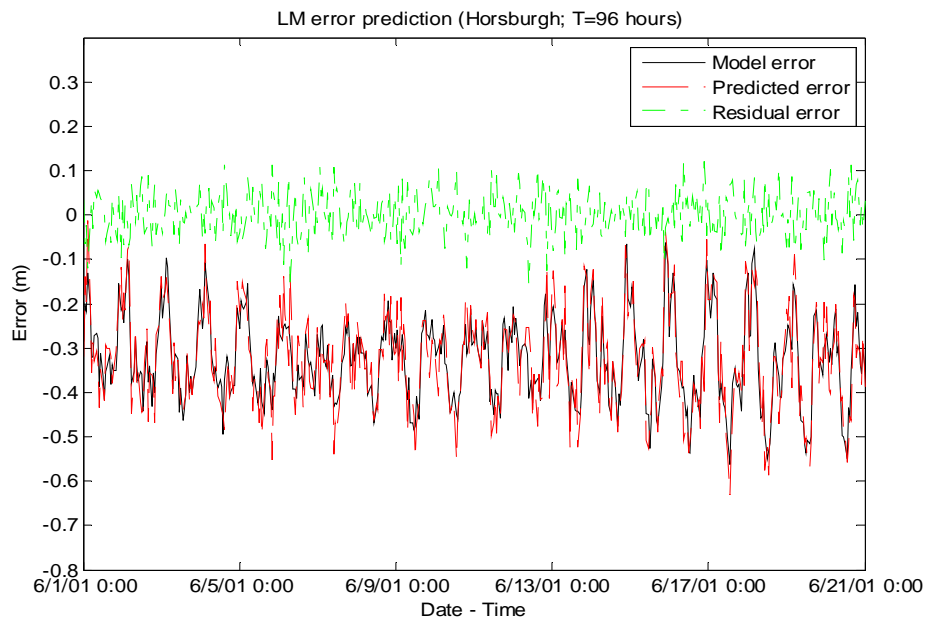


Figure 6.6 Error prediction with local model at Horsburgh (T=96 hours).

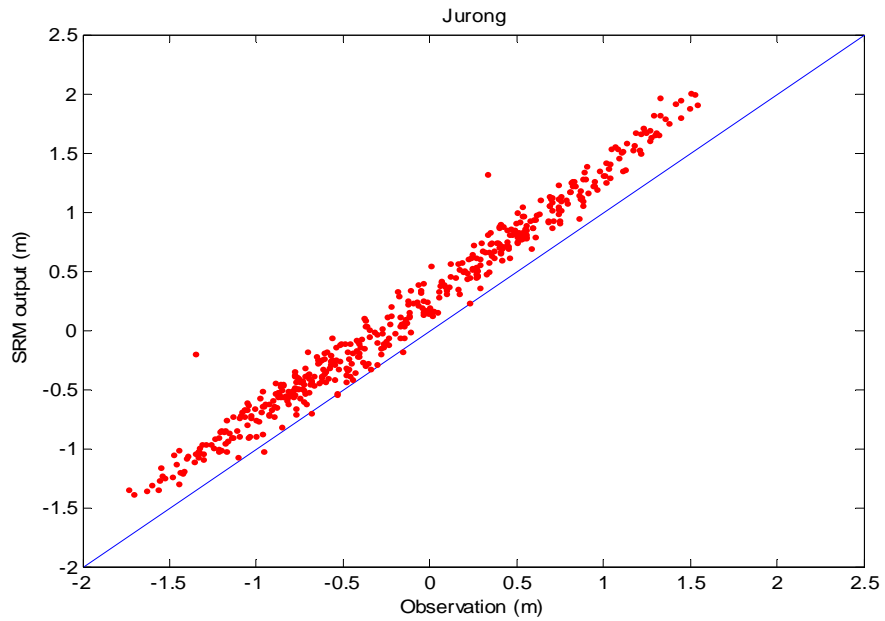


Figure 6.7 Scatter diagrams of SRM outputs at Jurong.

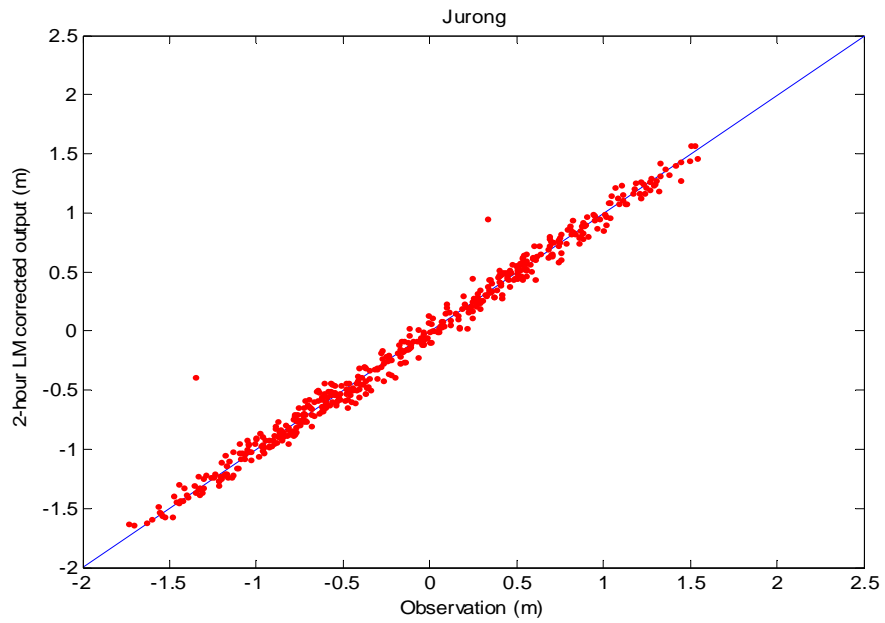


Figure 6.8 Scatter diagrams of LM corrected outputs at Jurong (T=2 hours).

CHAPTER 6. ERROR PREDICTION WITH LOCAL MODEL  
AND MULTILAYER PERCEPTRON

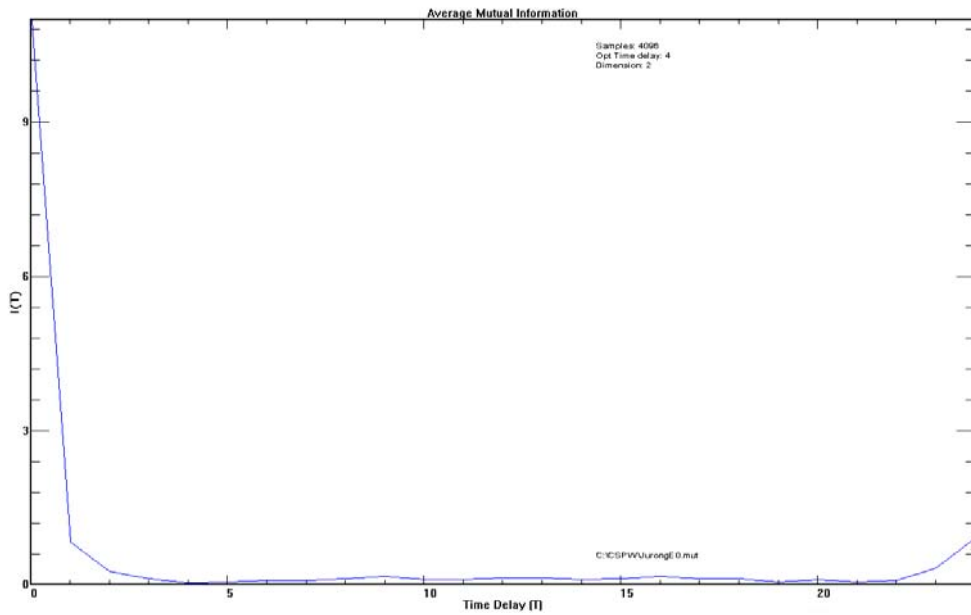


Figure 6.9 Average mutual information of the model errors at Jurong.

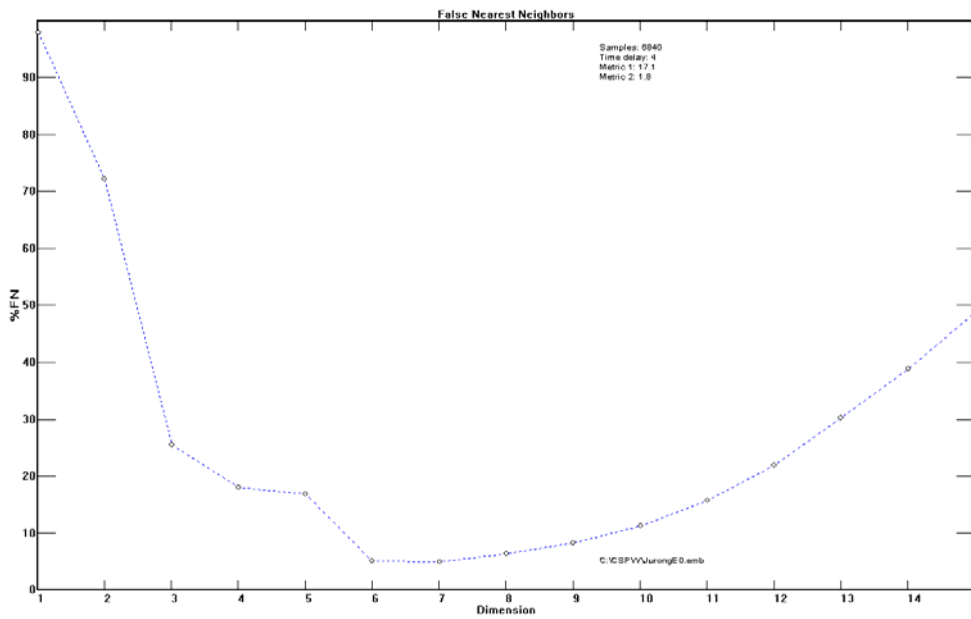


Figure 6.10 False nearest neighbors analysis for the model errors at Jurong.

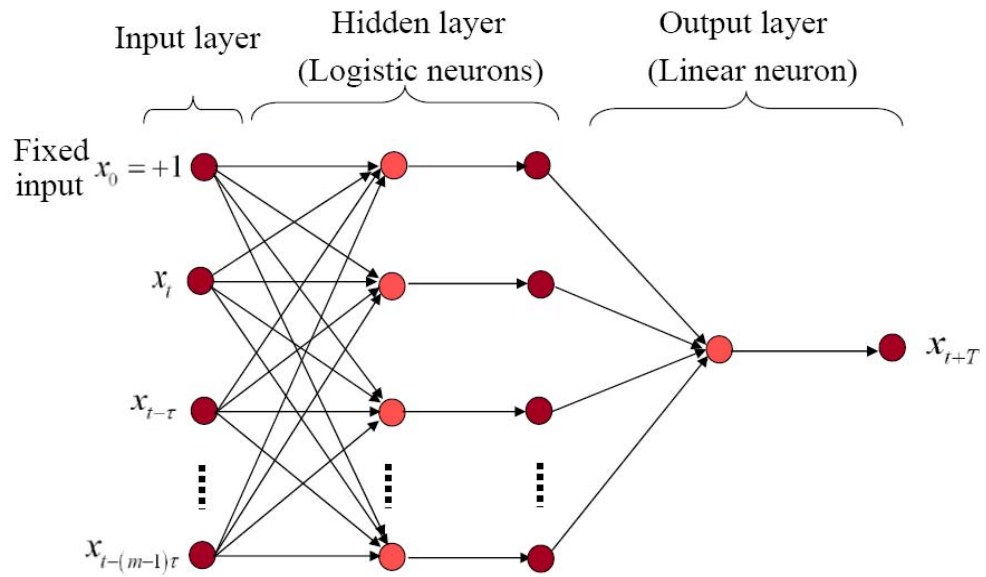


Figure 6.11 Architecture of multilayer perceptron in error prediction.

CHAPTER 6. ERROR PREDICTION WITH LOCAL MODEL  
AND MULTILAYER PERCEPTRON

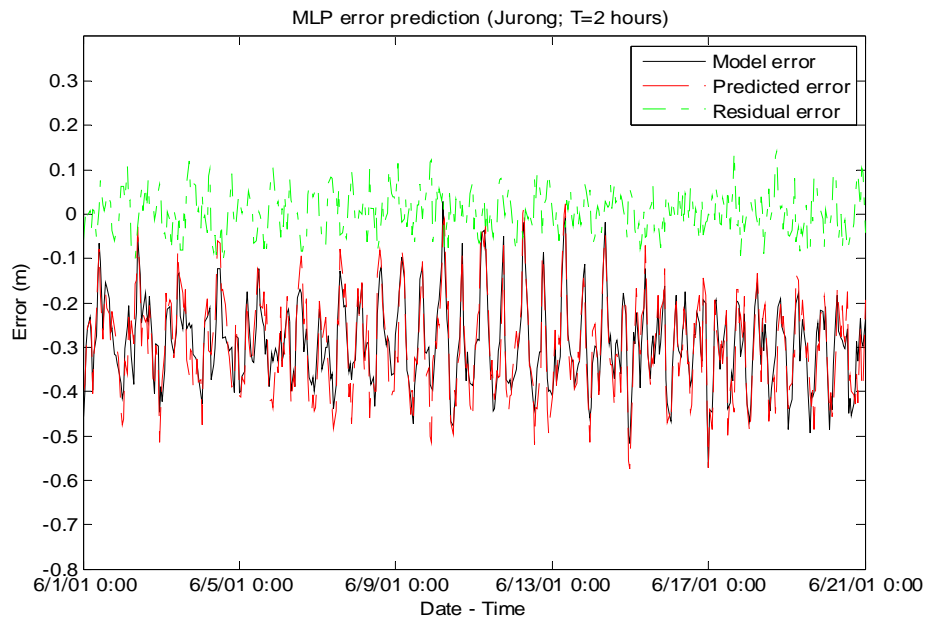


Figure 6.12 Error prediction with multilayer perceptron at Jurong (T=2 hours).

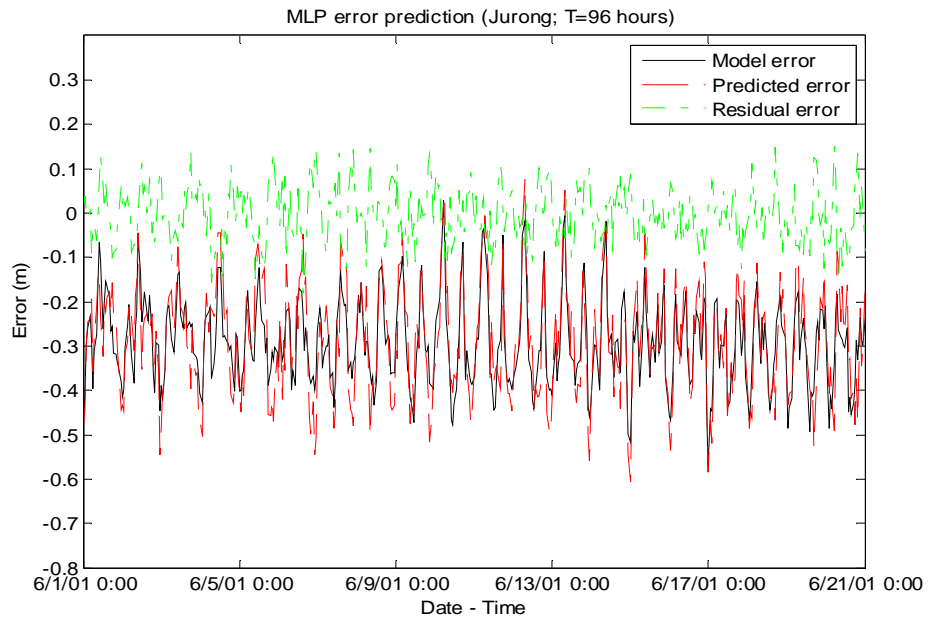


Figure 6.13 Error prediction with multilayer perceptron at Jurong (T=96 hours).

CHAPTER 6. ERROR PREDICTION WITH LOCAL MODEL  
AND MULTILAYER PERCEPTRON

---

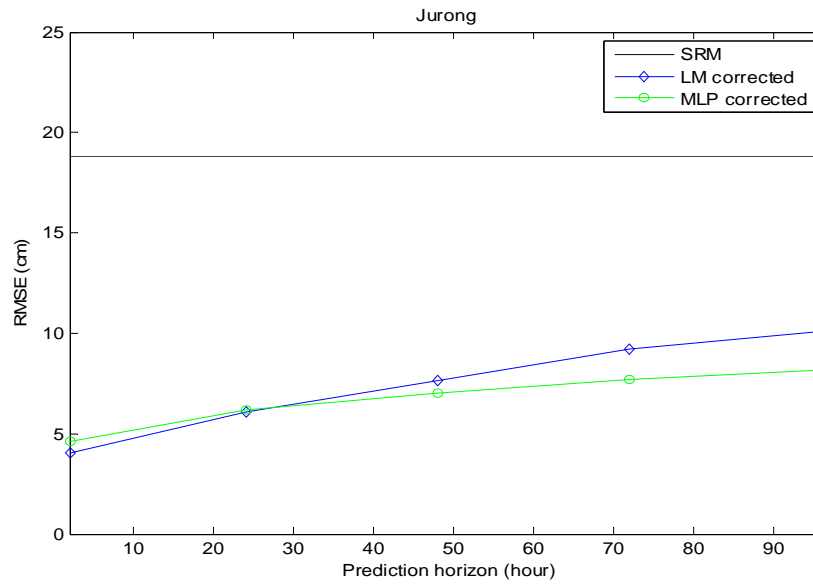


Figure 6.14 RMSE vs. prediction horizon at Jurong.

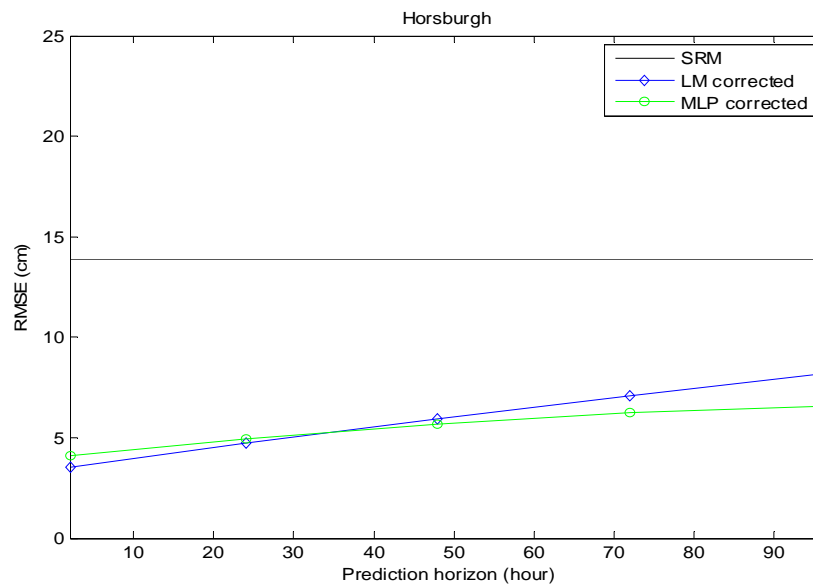


Figure 6.15 RMSE vs. prediction horizon at Horsburgh.

Table 6.1 Parameter settings in genetic algorithm.

Parameters	Value	Remark
T	2, 24, 48, 72, 96	prediction horizon
parmin	1, 1, 1	minimum values of $m$ , $\tau$ and $k$
parmax	20, 50, 100	maximum values of $m$ , $\tau$ and $k$
npopsiz	10	population size
maxgen	200	maximum number of generation
pcross	50	probability of crossover (%)
pmutate	2	probability of mutation (%)
rmsestop	0	stop RMSE



CHAPTER 6. ERROR PREDICTION WITH LOCAL MODEL  
AND MULTILAYER PERCEPTRON

---

Table 6.2 Embedding parameters ( $m$ ,  $\tau$ ,  $k$ ) in local model.

	Jurong			Tanjong Pagar			Bukom			Raffles			Sembawang			Horsburgh		
	$m$	$\tau$	$k$	$m$	$\tau$	$k$	$m$	$\tau$	$k$	$m$	$\tau$	$k$	$m$	$\tau$	$k$	$m$	$\tau$	$k$
T=2 Hr	3	1	35	3	1	13	3	1	24	4	1	36	2	1	28	4	1	23
T=24 Hr	4	12	53	4	21	44	5	17	50	5	15	49	3	12	48	6	21	37
T=48 Hr	5	25	41	6	24	40	6	16	51	7	11	76	4	21	43	7	31	34
T=72 Hr	7	29	41	8	32	51	7	44	12	8	44	41	5	44	46	9	44	28
T=96 Hr	9	36	44	12	34	60	10	40	17	11	36	34	6	34	39	14	43	53

CHAPTER 6. ERROR PREDICTION WITH LOCAL MODEL  
AND MULTILAYER PERCEPTRON

---

Table 6.3 Statistics of residual errors at the measurement stations (local model).

	Jurong		Tanjong Pagar		Bukom		Raffles		Sembawang		Horsburgh	
	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r
SRM	18.80	0.91	18.57	0.91	16.54	0.91	16.16	0.91	19.40	0.89	13.91	0.93
T=2 Hr	4.05	0.99	4.09	0.99	3.67	0.99	3.58	0.99	4.28	0.98	3.51	0.99
T=24 Hr	6.08	0.99	6.11	0.99	5.74	0.99	5.82	0.99	5.93	0.98	4.73	0.99
T=48 Hr	7.64	0.98	7.81	0.98	7.04	0.98	7.23	0.98	7.30	0.98	5.96	0.98
T=72 Hr	9.21	0.98	8.94	0.98	8.27	0.98	8.35	0.98	8.49	0.97	7.11	0.98
T=96 Hr	10.12	0.97	9.76	0.97	9.07	0.97	9.11	0.97	9.60	0.97	8.19	0.98
Average	7.42	0.98	7.34	0.98	6.76	0.98	6.82	0.98	7.12	0.98	5.90	0.98

CHAPTER 6. ERROR PREDICTION WITH LOCAL MODEL  
AND MULTILAYER PERCEPTRON

---

Table 6.4 Embedding parameters ( $m$ ,  $\tau$ ) in multilayer perceptron.

	Jurong	Tanjong Pagar	Bukom	Raffles	Sembawang	Horsburgh
$m$	6	5	4	4	5	4
$\tau$	4	4	6	6	4	5

CHAPTER 6. ERROR PREDICTION WITH LOCAL MODEL  
AND MULTILAYER PERCEPTRON

---

Table 6.5 Statistics of residual errors at the measurement stations (multilayer perceptron).

	Jurong		Tanjong Pagar		Bukom		Raffles		Sembawang		Horsburgh	
	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r
SRM	18.80	0.91	18.57	0.91	16.54	0.91	16.16	0.91	19.40	0.89	13.91	0.93
T=2 Hr	4.64	0.99	4.71	0.99	4.33	0.99	4.43	0.99	5.34	0.98	4.12	0.99
T=24 Hr	6.21	0.99	5.94	0.99	5.59	0.99	5.49	0.99	6.14	0.98	4.93	0.99
T=48 Hr	7.03	0.98	6.91	0.98	6.43	0.98	6.24	0.98	6.82	0.98	5.67	0.98
T=72 Hr	7.69	0.98	7.56	0.98	6.85	0.98	6.66	0.98	7.46	0.97	6.23	0.98
T=96 Hr	8.19	0.97	8.03	0.97	7.03	0.97	6.99	0.97	7.99	0.97	6.58	0.98
Average	6.75	0.98	6.63	0.98	6.05	0.98	5.96	0.98	6.75	0.98	5.51	0.98

## **Chapter 7**

# **Error Distribution with Kalman Filter and Multilayer Perceptron**

### **7.1 Introduction**

Having predicted the model errors at the measurement stations within the chaotic dynamic system, the forecasting interest can be extended to the remainder of the computational domain by distributing the predicted model errors using optimal interpolation schemes. Two approaches of error distribution are explored in the present study, i.e. Kalman filter and multilayer perceptron.

Among the 6 stations in the Singapore Regional Model domain, 2 near shore stations, i.e. Jurong and Tanjong Pagar, are assumed to be the measurement stations, where the model errors are directly forecasted using multilayer perceptron based on the past recordings. The remaining 4 stations, i.e. Bukom, Raffles, Sembawang and Horsburgh, are assumed to be the non-measurement stations, where the model outputs are corrected by spatially distributing the predicted errors from the measurement stations. The corrected model outputs at the non-measurement stations are compared against the observations, which are assumed to be unavailable in the process of error correction.

In order to test the sensitivity of the error distribution approaches to the information contained in the predicted errors at the 2 measurement stations, 3 different combinations are utilized to distribute the errors,

- Case 1: Jurong and Tanjong Pagar;
- Case 2: Jurong only;
- Case 3: Tanjong Pagar only.

The correlation coefficients between the Singapore Regional Model outputs at the measurement stations and the non-measurement stations are summarized in Table 7.1. As indicated, the model outputs are highly correlated at the stations located near to each other, while the correlation becomes smaller when the distance increases.

## 7.2 Application of Kalman Filter in Error Distribution

### 7.2.1 Error Statistics Approximation

As discussed in Chapter 4, applying Kalman filter requires the knowledge of the error statistics of the numerical models and the measurements. Under the assumption of isotropy and homogeneity, in this study the model error covariance  $\mathbf{Q}$  and the measurement error covariance  $\mathbf{R}$  are explicitly formulated as

$$\mathbf{Q} = [Q_{i,j}] = \rho^{d_{ij}} \sigma_{mo}^2, \quad (7.1)$$

$$\mathbf{R} = \text{diag}[\sigma_{me1}^2, \dots, \sigma_{mep}^2], \quad (7.2)$$

where  $\rho$  is the spatial correlation for the model errors,  $d_{ij} = |i - j|/\sqrt{2}$  is the distance between element  $ij$  and diagonal in matrix  $\mathbf{Q}$ ,  $\sigma_{mo}$  is the standard deviation of the model

errors,  $\sigma_{mei}$  ( $i = 1, 2, \dots, p$ ) is the standard deviation of the  $i$ th measurement error, and  $p$  is the total number of measurements.

To determine the parameters in Equations (7.1) and (7.2), an exhaustive search is executed in a predefined range. The possible spatial correlation values considered are 0.6, 0.7, ..., 1.0, while the standard deviations for model errors and measurement errors are set between 0.01 to 0.20 with a step of 0.01. For each combination of parameters, Kalman filter is run and the updating process is conducted. The selection criterion is based on the residual errors between the measurements and the corrected model forecasts. Following parameters are finally specified,

- Spatial correlation for the model errors:  $\rho = 0.9$ ;
- Standard deviation for the model errors:  $\sigma_{mo} = 0.15$  m;
- Standard deviation for the measurement errors:  $\sigma_{mei} = 0.01$  m ( $i = 1, 2, \dots, p$ ).

In the searching process, the Kalman filter is quite robust to the error statistics. Small changes in the parameters will not affect the results. This finding agrees with Madsen and Cañizares (1999), Cañizares et al. (2001) and Sørensen et al. (2001).

The error covariance is propagated off-line after determining  $\mathbf{Q}$  and  $\mathbf{R}$ . A constant Kalman gain  $\mathbf{K}$  is then calculated as the average of the Kalman gains  $\mathbf{K}_k$  in the off-line error covariance propagation. Once the Kalman gain  $\mathbf{K}$  is available, it will serve as a weighting function between the model forecast and the measurement, with only the equations for the state forecast and analysis solved on-line.

### **7.2.2 Results**

Overviews of error distribution efficiency with Kalman filter at Bukom, Raffles, Sembawang and Horsburgh are summarized in Tables 7.2 – 7.5 respectively, in terms of residual root mean square error as well as the correlation coefficient after error correction. The best cases of error distribution are highlighted (\*) in Tables 7.2 – 7.5. The distributed errors at Horsburgh from the best cases are plotted in Figures 7.1 and 7.2, together with the model errors and the residual errors.

At Bukom and Raffles, where the outputs of the Singapore Regional Model at the non-measurement stations and the measurement stations are highly correlated, Kalman filter performs well in distributing the predicted errors to the non-measurement stations. For the best cases, more than 50% of the root mean square errors in the Singapore Regional Model outputs have been removed on average. The correlation coefficient also increases from 0.91 to 0.97 after error distribution. However, when the correlation decreases, the performance of Kalman filter in error distribution deteriorates at a rapid pace. As shown in Figures 7.1 and 7.2, although the distributed errors are still able to follow the up and down trends of the model errors, the magnitudes of the residual errors increase significantly at Horsburgh. Averaged over 5 prediction horizons, Kalman filter removes only about 40% of the root mean square errors at Sembawang and 20% of the root mean errors at Horsburgh.

## **7.3 Application of Multilayer Perceptron in Error Distribution**

### **7.3.1 Methodology**



The Singapore regional model, due to the shoaling effect, produces larger model errors at the near shore measurement stations, where the water levels are relatively larger. In other words, the magnitude of the model errors is directly proportional to the magnitude of the water levels. In addition, as shown in Figures 5.4 to 5.7, the model errors reach maximum absolute values at the tidal level extrema, whereas the model errors reach minimum absolute values at zero tidal levels. The high correlation between the model errors and the model outputs inspires an error distribution scheme based on multilayer perceptron.

Assuming that the model errors will be distributed in a similar manner as the distribution of the model outputs, the error distribution scheme based on multilayer perceptron can be accomplished in 3 steps,

- Step 1. Approximate with multilayer perceptron a non-linear mapping function  $\phi(\cdot)$  between the numerical model outputs at the measurement stations  $M$  and the non-measurement stations  $N$

$$N = \phi(M) \tag{7.3}$$

- Step 2. Given the model errors forecasted at the measurement stations,  $\hat{E}_M$ , the distributed errors at the non-measurement stations  $\hat{E}_N$  can be calculated by

$$\hat{E}_N = \phi(\hat{E}_M) \tag{7.4}$$

- Step 3. Correct the numerical model outputs at the non-measurement stations

$$\hat{N} = N + \hat{E}_N \tag{7.5}$$

To approximate the mapping function  $\phi(\cdot)$ , the model outputs at the measurement stations are fed as input to the multilayer perceptron with the model outputs at the non-measurement stations being the desired response. Therefore, as shown in Figure 7.3, multilayer perceptron is composed of an input layer with two input neurons or one input neuron, corresponding to case 1 and cases 2 or 3, a hidden layer, and an output layer with 4 output neurons. Logistic function and linear function are respectively used as the activation function in the hidden layer and in the output layer. Parameters for the networks and in the training algorithm are set as follows

- No. of hidden neurons: 50
- No. of epochs: 200
- learning rate:  $\eta = 0.1$
- momentum constant:  $\alpha = 0.7$

To avoid converging to local optima, a set of 5 multilayer perceptrons with different initial synaptic weights is trained. The multilayer perceptron with the best accuracy is selected as the optimal network to distribute the model errors.

### 7.3.2 Results

Overviews of error distribution efficiency with multilayer perceptron at Bukom, Raffles, Sembawang and Horsburgh are summarized in Tables 7.6 – 7.9 respectively, in terms of residual root mean square error as well as the correlation coefficient after error correction. The best cases of error distribution are highlighted (\*) in Tables 7.6 – 7.9. Compared to Kalman filter, multilayer perceptron is less sensitive to the correlations. For the best cases,

multilayer perceptron removes more than 50% of the root mean square errors at Bukom and Raffles on average, while the error reduction keeps larger than 40% at Sembawang and Horsburgh.

Figures 7.4 and 7.5 illustrate examples of error distribution with multilayer perceptron at Horsburgh when prediction horizon  $T = 2$  hours and  $T = 96$  hours. Both the 2-hour and 96-hour distributed errors successfully approach the model errors, generating residual errors with reduced magnitudes.

#### **7.4 Comparison between Kalman Filter and Multilayer Perceptron**

When the model outputs at the measurement stations and the non-measurement stations are highly correlated, i.e. at Bukom and Raffles, both Kalman filter and multilayer perceptron successfully remove over 50% of the root mean square errors. When the correlation decreases, i.e. at Sembawang and Horsburgh, the performance of both Kalman filter and multilayer perceptron deteriorates. Kalman filter can only remove about 40% of the root mean square errors at Sembawang and 20% at Horsburgh. In comparison, multilayer perceptron has a more consistent performance in error distribution, with an error reduction of more than 40% at both Sembawang and Horsburgh. Figures 7.6 plots the residual root mean square error at Horsburgh resulted from both Kalman filter and multilayer perceptron against the prediction horizon, together with the root mean square error before error correction, from which the better error distribution performance of multilayer perceptron in case of smaller correlations can be noticed.

Multilayer perceptron is also more computationally efficient than Kalman filter in model error distribution. Kalman filter is coded in MATLAB and run in an Intel Core Duo T2400 1.83 GHz machine with 2.5 GB RAM. Although assumptions about the error covariance structures and the constant Kalman gain accelerate the operation of Kalman filter, it is required to run Kalman filter dozens of times in the parameter searching procedure, each of which takes a computational time of 2 hours. Multilayer perceptron is implemented using MATLAB Neural Networks Toolbox and running in the same machine. It takes about 20 hours to train a network, and one multilayer perceptron is capable of distributing the model errors to all the non-measurement stations. Considering both performance efficiency and computational cost, multilayer perceptron is the preferable approach to distribute the model errors.

As indicated in Tables 7.2 – 7.9, distributing the predicted errors from both measurement stations gives the best results at Bukom and Raffles, while at Sembawang and Horsburgh the cases with only one measurement station show the superiority. Due to the misleading information from less correlated stations, more measurement stations will not necessarily guarantee more accurate error distribution. Apart from the relative geographical locations, the correlation coefficients of the model outputs are valuable references in selecting the correlated measurement stations. As suggested by this finding, for locating future measurement stations which are also intended for model correction purposes, a prior correlation analysis among possible sites is favorable.

CHAPTER 7. ERROR DISTRIBUTION WITH KALMAN FILTER  
AND MULTILAYER PERCEPTRON

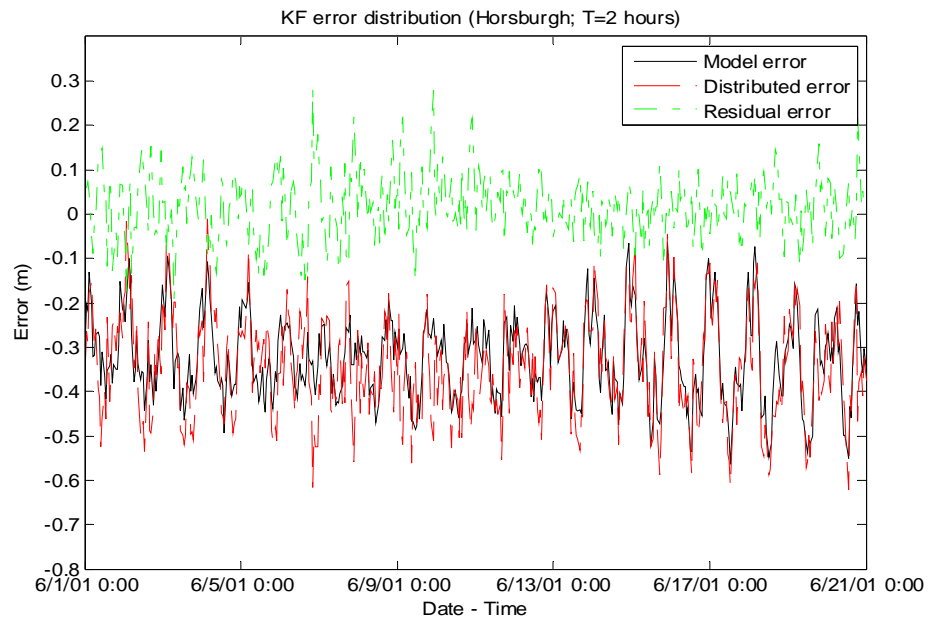


Figure 7.1 Error distribution with Kalman filter at Horsburgh (T=2 hours; Case 3).

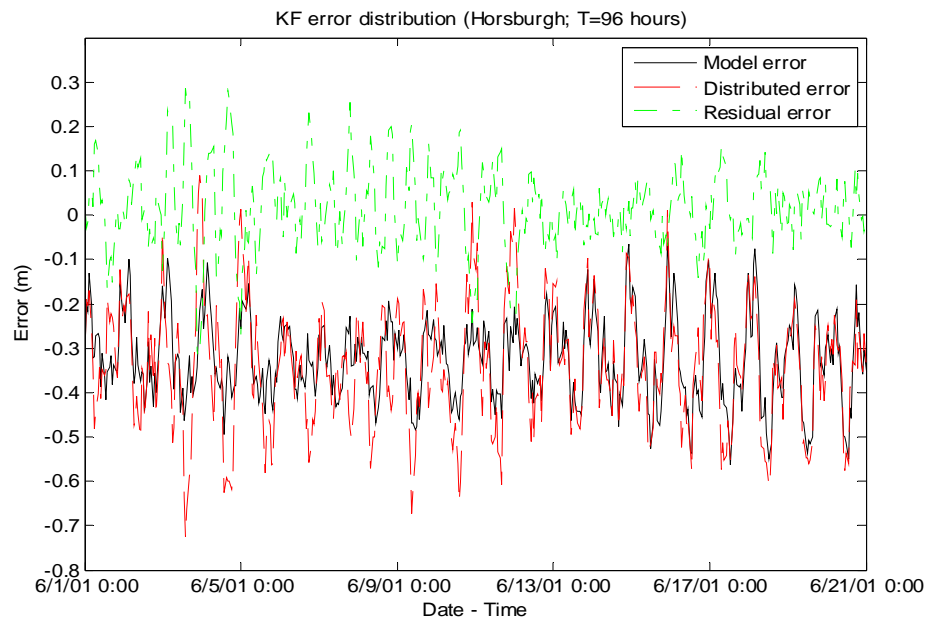


Figure 7.2 Error distribution with Kalman filter at Horsburgh (T=96 hours; Case 3).

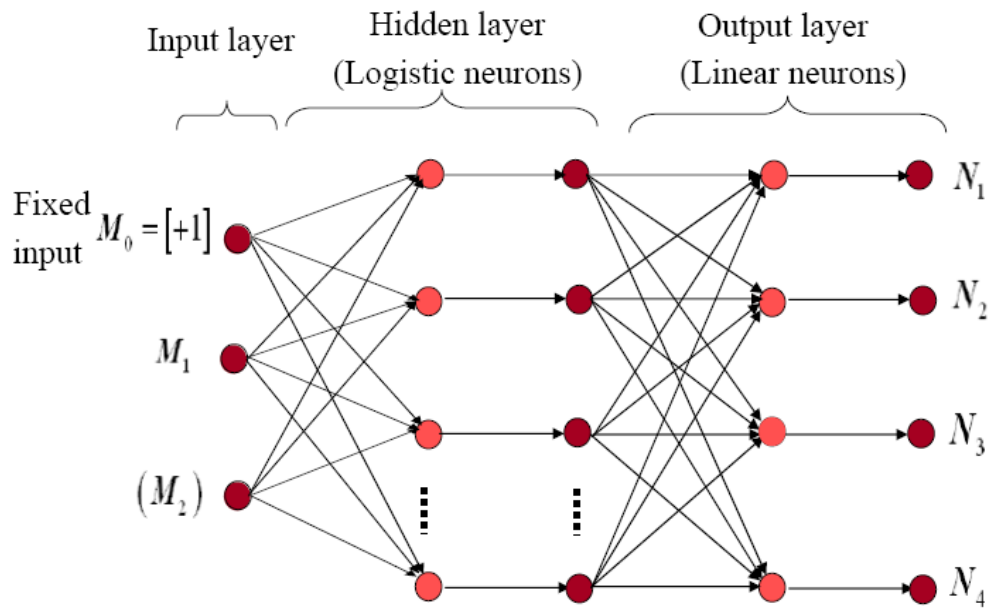


Figure 7.3 Architecture of multilayer perceptron in error distribution.

## CHAPTER 7. ERROR DISTRIBUTION WITH KALMAN FILTER AND MULTILAYER PERCEPTRON

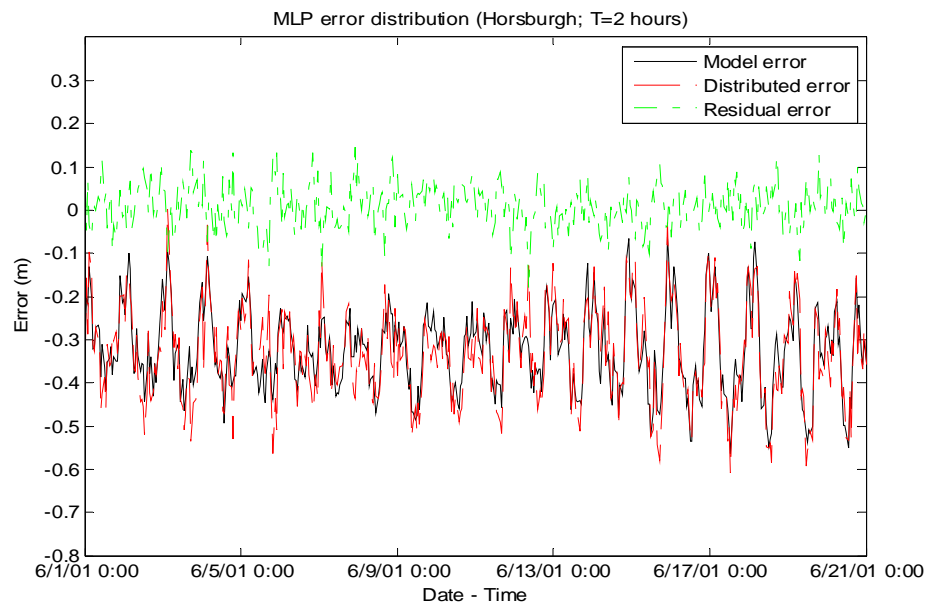


Figure 7.4 Error distribution with multilayer perceptron at Horsburgh (T=2 hours; Case 3).

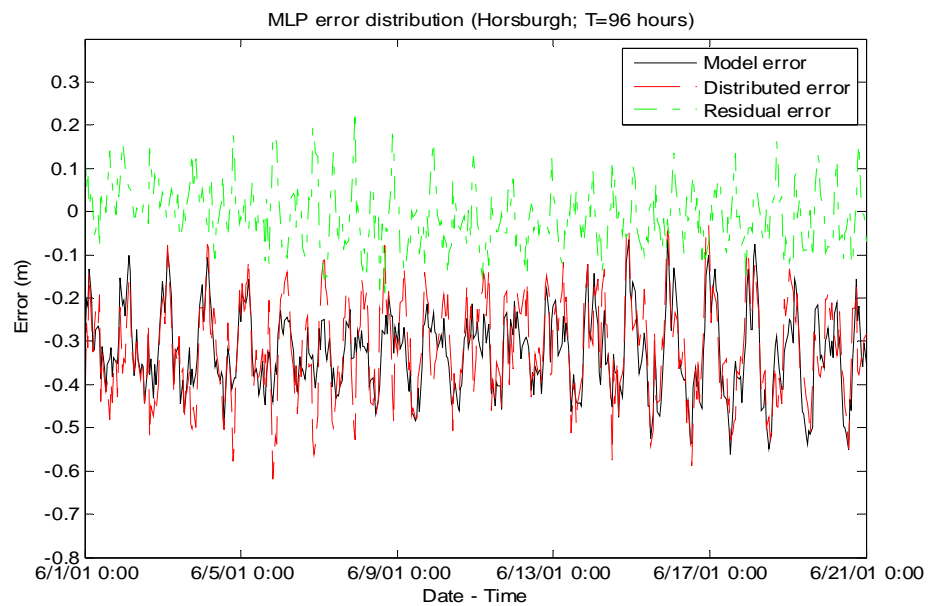


Figure 7.5 Error distribution with multilayer perceptron at Horsburgh (T=96 hours; Case 3).

CHAPTER 7. ERROR DISTRIBUTION WITH KALMAN FILTER  
AND MULTILAYER PERCEPTRON

---

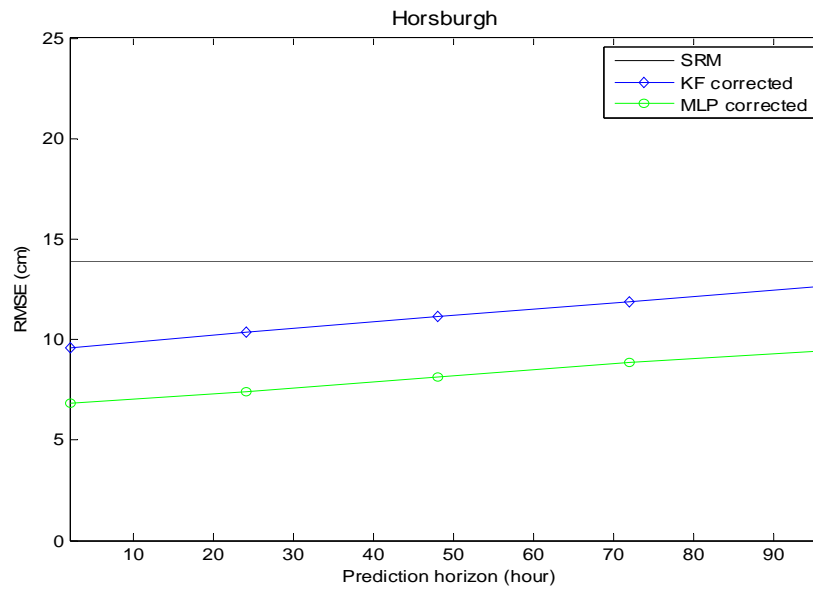


Figure 7.6 RMSE vs. prediction horizon at Horsburgh



CHAPTER 7. ERROR DISTRIBUTION WITH KALMAN FILTER  
AND MULTILAYER PERCEPTRON

---

Table 7.1 Correlation coefficient between the SRM outputs at the measurement stations and the non-measurement stations.

	Bukom	Raffles	Sembawang	Horsburgh
Jurong	0.9574	0.9482	0.8028	0.6805
Tanjong Pagar	0.9520	0.9475	0.9016	0.8040
Average	0.9547	0.9479	0.8522	0.7423

CHAPTER 7. ERROR DISTRIBUTION WITH KALMAN FILTER  
AND MULTILAYER PERCEPTRON

---

Table 7.2 Statistics of residual errors at Bukom (Kalman filter; \*: best case).

	Bukom					
	Case 1 *		Case 2		Case 3	
	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r
SRM	16.54	0.91	16.54	0.91	16.54	0.91
T=2 Hr	5.19	0.98	5.30	0.98	5.39	0.98
T=24 Hr	6.46	0.97	6.75	0.97	6.77	0.97
T=48 Hr	7.65	0.97	7.92	0.97	7.98	0.97
T=72 Hr	8.80	0.97	8.98	0.97	9.06	0.97
T=96 Hr	9.82	0.97	10.07	0.97	10.45	0.97
Average	7.58	0.97	7.80	0.97	7.92	0.97

Table 7.3 Statistics of residual errors at Raffles (Kalman filter; \*: best case).

	Raffles					
	Case 1 *		Case 2		Case 3	
	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r
SRM	16.16	0.91	16.16	0.91	16.16	0.91
T=2 Hr	5.33	0.98	5.43	0.98	5.44	0.98
T=24 Hr	6.58	0.97	6.72	0.97	6.75	0.97
T=48 Hr	7.75	0.97	7.98	0.97	8.14	0.97
T=72 Hr	8.84	0.97	9.02	0.97	9.35	0.97
T=96 Hr	9.89	0.97	10.17	0.97	10.49	0.97
Average	7.69	0.97	7.86	0.97	8.03	0.97

CHAPTER 7. ERROR DISTRIBUTION WITH KALMAN FILTER  
AND MULTILAYER PERCEPTRON

---

Table 7.4 Statistics of residual errors at Sembawang (Kalman filter; \*: best case).

	Sembawang					
	Case 1		Case 2		Case 3 *	
	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r
SRM	19.40	0.89	19.40	0.89	19.40	0.89
T=2 Hr	10.54	0.96	12.09	0.95	9.35	0.97
T=24 Hr	11.37	0.95	12.86	0.95	10.46	0.96
T=48 Hr	12.14	0.95	13.72	0.94	11.48	0.95
T=72 Hr	12.97	0.94	14.56	0.93	12.51	0.95
T=96 Hr	13.66	0.94	15.23	0.93	13.87	0.94
Average	12.14	0.95	13.69	0.94	11.53	0.95

Table 7.5 Statistics of residual errors at Horsburgh (Kalman filter; \*: best case).

	Horsburgh					
	Case 1		Case 2		Case 3 *	
	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r
SRM	13.91	0.93	13.91	0.93	13.91	0.93
T=2 Hr	9.73	0.97	10.37	0.96	9.57	0.97
T=24 Hr	10.60	0.96	11.04	0.95	10.35	0.96
T=48 Hr	11.48	0.95	11.81	0.95	11.13	0.96
T=72 Hr	12.27	0.94	12.63	0.94	11.87	0.95
T=96 Hr	13.05	0.94	13.58	0.93	12.657	0.94
Average	11.43	0.95	11.89	0.95	11.11	0.96

CHAPTER 7. ERROR DISTRIBUTION WITH KALMAN FILTER  
AND MULTILAYER PERCEPTRON

---

Table 7.6 Statistics of residual errors at Bukom (multilayer perceptron; \*: best case).

	Bukom					
	Case 1 *		Case 2		Case 3	
	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r
SRM	16.54	0.91	16.54	0.91	16.54	0.91
T=2 Hr	5.28	0.98	5.46	0.98	5.63	0.98
T=24 Hr	6.14	0.97	6.57	0.97	6.78	0.97
T=48 Hr	7.06	0.97	7.59	0.97	7.85	0.97
T=72 Hr	7.98	0.97	8.67	0.97	8.99	0.97
T=96 Hr	8.92	0.97	9.70	0.97	10.14	0.97
Average	7.08	0.97	7.60	0.97	7.88	0.97

Table 7.7 Statistics of residual errors at Raffles (multilayer perceptron; \*: best case).

	Raffles					
	Case 1 *		Case 2		Case 3	
	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r
SRM	16.16	0.91	16.16	0.91	16.16	0.91
T=2 Hr	5.17	0.98	5.34	0.98	5.68	0.98
T=24 Hr	6.05	0.97	6.37	0.97	6.87	0.97
T=48 Hr	6.97	0.97	7.49	0.97	8.04	0.97
T=72 Hr	7.94	0.97	8.51	0.97	9.15	0.97
T=96 Hr	8.87	0.97	9.67	0.97	10.34	0.97
Average	7.00	0.97	7.86	0.97	8.02	0.97

CHAPTER 7. ERROR DISTRIBUTION WITH KALMAN FILTER  
AND MULTILAYER PERCEPTRON

---

Table 7.8 Statistics of residual errors at Sembawang (multilayer perceptron; \*: best case).

	Sembawang					
	Case 1		Case 2		Case 3 *	
	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r
SRM	19.40	0.89	19.40	0.89	19.40	0.89
T=2 Hr	8.46	0.97	9.20	0.97	7.93	0.97
T=24 Hr	9.47	0.97	10.21	0.96	8.84	0.97
T=48 Hr	10.51	0.96	11.37	0.95	9.78	0.96
T=72 Hr	11.59	0.95	12.46	0.94	10.65	0.96
T=96 Hr	12.66	0.94	13.73	0.94	11.73	0.95
Average	10.54	0.96	11.39	0.95	9.79	0.96

Table 7.9 Statistics of residual errors at Horsburgh (multilayer perceptron; \*: best case).

	Horsburgh					
	Case 1		Case 2		Case 3 *	
	RMSE (cm)	r	RMSE (cm)	r	RMSE (cm)	r
SRM	13.91	0.93	13.91	0.93	13.91	0.93
T=2 Hr	7.47	0.97	8.03	0.97	6.82	0.97
T=24 Hr	8.30	0.97	9.01	0.97	7.41	0.97
T=48 Hr	9.24	0.97	9.92	0.96	8.15	0.97
T=72 Hr	10.17	0.96	10.80	0.96	8.87	0.97
T=96 Hr	11.05	0.95	11.78	0.95	9.43	0.97
Average	9.25	0.96	9.91	0.96	8.14	0.97

## **Chapter 8**

# **Use of Data Assimilation in Understanding Sea Level Anomalies**

### **8.1 Introduction**

Sea level anomalies (SLA), defined as non-tidal water levels, are known to be important phenomena in narrow straits separating large water bodies, such as the Singapore and Malacca Straits. Many locations in this region experience sea level anomalies as high as 60 cm to 130 cm, indicating the presence of a strong non-tidal driving force. Effects of sea level anomalies can at times dominate the tidal flow conditions; reports from pilot's logs indicate that in monsoon periods current velocities may be much stronger and more persistent than suggested by the tidal tables; on occasions flow reversal actually disappears. This provides a motivation for studying of sea level anomalies and associated current anomalies (CA) in the Singapore and Malacca Straits, targeting on distinguishing the driving forces behind these processes and assessing whether they can be modelled in real time to facilitate navigation and port operation support systems.

The origin of sea level anomalies is not fully understood, but some research reveals that persistent basin-scale monsoon winds and their short scale variations over the South

China Sea and Andaman Sea are major contributing factors, creating regional water level variations that drive residual currents through the Straits (Babovic, 2007; Gerritsen et al., 2009). As discussed in Chapters 1 and 5, the Singapore Regional Model fails to consider the influence from the wind, and hence is unable to model the sea level anomalies. Even if wind was included in the Singapore Regional Model as an external force, the inaccuracy in the wind data, the uncertainty in meteorological prediction, the constraint from the domain size, will hamper the real time modelling of sea level anomalies.

This chapter tends to apply the data assimilation techniques in the analysis and prediction of sea level anomalies. At the open boundaries of the Singapore Regional Model, hourly sea level anomalies can be predicted using multilayer perceptron with prediction horizon  $T = 24$  hours. Imposing sea level anomalies predicted at the open boundaries as driving force to the numerical model will facilitate numerically modelling the sea level anomalies inside the model domain. The extraction of the anomaly data, the procedure of data assimilation, and the detailed results will be discussed in this chapter.

## **8.2 Overview of Sea Level Anomalies**

### **8.2.1 Sources of Marine Data**

The water level data used to extract sea level anomalies are primarily from archived radar altimeter data, measured by satellites that routinely track the open ocean with repeated cycles of 10 days (T/P and Jason-1) or 35 days (ERS 1/2; Envisat). Figure 8.1 shows the Jason-1 and Envisat ground tracks in the vicinity of Malacca Strait, with Table 8.1 summarizing some general properties of these two satellites. Jason-1 has a higher

temporal resolution and lower spatial resolution, while Envisat has a lower temporal resolution and higher spatial resolution.

The altimeter sea level anomalies can be accessed from the Radar Altimeter Database System (RADS, <http://rads.tudelft.nl>) and the gridded AVISO (DUACS, [www.aviso.oceanobs.com](http://www.aviso.oceanobs.com)) data sets. RADS provides level 2 altimeter data, implying these are geo-located along-track measurements at 10 days or 35 days intervals with calibrations and corrections applied to convert the raw measurement into a geophysical quantity of interest. Comparatively, DUACS provides level 3 altimeter data, implying that the along-track sea level anomalies have been processed into composite gridded maps ( $1/3^{\circ} \times 1/3^{\circ}$ ) at weekly intervals based on optimal interpolation algorithms.

As a complementary source of sea level anomalies, the in-situ water level measurements are obtained from the database at the University of Hawaii Sea Level Center (UHSLC, <http://ilikai.soest.hawaii.edu/uhs/c/datai.html>). These research quality data sets are based on the local tidal gauge measurements with good temporal coverage. Figure 8.2 depicts the geographical locations for various measurement stations in the region, with station codes inherited from the UHSLC website. The water level data are available at 17 stations for periods between 1996 and 2005.

### 8.2.2 Extraction of Sea Level Anomalies

Sea level anomalies are the residuals when the astronomic tide is subtracted from sea level measurements, mathematically stated,

$$\text{SLA}(t) = \text{Measured water level}(t) - \text{Tidal water level}(t) \quad (8.1)$$



Tidal water level in Equation (8.1) can be described in terms of a series of simple harmonic constituent motions. The general formula for the astronomic tide is,

$$H(t) = A_o + \sum_{i=1}^k A_i \cdot F_i \cdot \cos(\omega_i \cdot t + (V_o + u)_i - G_i), \quad (8.2)$$

where

- $H(t)$  - astronomic tidal level
- $A_o$  - mean water level
- $i$  - index of a constituent
- $k$  - number of relevant constituents
- $A_i$  - amplitude of a constituent
- $F_i$  - amplitude factor of a constituent
- $\omega_i$  - angular velocity of a constituent
- $(V_o + u)_i$  - astronomical argument of a constituent
- $G_i$  - phase lag of a constituent

In Equation (8.2),  $F_i$  and  $(V_o + u)_i$  are frequency dependent, whose values are pre-determined for each constituent with own characteristic angular velocity  $\omega_i$ ;  $A_o$ ,  $A_i$  and  $G_i$  are on the other hand position dependent variables representing the local characteristics of the tide, whose values can be determined through tidal analysis based on the least-square fitting technique.

Figure 8.3 gives an example of the tidal analysis of the RADS altimeter data and the in-situ measurements. The amplitudes and phases of M2 along the satellite ground tracks

are plotted, while the full-dot symbols show the same information derived from the in-situ measurements. The results, analyzed from these two different data sources, show a good agreement.

In order to optimally de-tide the measured water levels, the number of relevant constituents  $k$  is determined through an iteration of tidal analysis and Fourier analysis. The steps of iteration are summarized as follows,

1. Initialize the constituents.

Initial set of constituents consists of annual constituent SA, semi-annual constituent SSA, the 8 major diurnal and semi-diurnal constituents, i.e. Q1, O1, P1, K1, N2, M2, S2 and K2, and other related higher-order harmonic constituents.

2. Conduct tidal analysis on the water levels.

The constituents with amplitudes smaller than  $A_{\min} = 0.002$  m are considered to be noise, and will be excluded in the next run.

3. Conduct Fourier analysis on the residuals.

Residuals' frequencies with amplitude larger than  $A_{\max} = 0.03$  m are picked up as constituents overlooked, and will be supplemented in the next run.

4. Repeat steps 2 and 3 with the updated constituents.

Steps 2 and 3 are repeated until all the constituents' amplitudes in tidal analysis are larger than  $A_{\min} = 0.002$  m, and no residuals' frequencies in Fourier analysis have amplitude larger than  $A_{\max} = 0.03$  m.

### 8.2.3 Statistical Analysis of Sea Level Anomalies

The in-situ sea level anomaly time series in 2004 are compiled and statistically analyzed for each UHSLC station. Table 8.2 summarizes the analysis results of the sea level anomalies at the 17 stations, in terms of mean, max, min, and root mean square (RMS, quadratic mean) defined as

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{t=1}^N \text{SLA}(t)^2}, \quad (8.3)$$

where  $N$  is the length of the sea level anomaly time series.

The mean sea level anomaly values at all the 17 stations are observed to be 0, indicating no major long term shifts or long term trends in sea level anomalies in this region. The sea level anomalies vary in a wide range between -78 cm to 79 cm. Stations 322 and 140 appear to be high sea level anomaly activity locations with largest RMS values, whereas station 387 shows smallest sea level anomaly fluctuations. Due to the high complexity in the bathymetry and topography of this region, there exists no obvious regularity for sea level anomaly distribution, but in general anomalous ocean dynamics is more pronounced in shallower water area.

#### **8.2.4 RADS SLA vs. DUACS SLA**

Figures 8.4 and 8.5 respectively show the along track RADS sea level anomalies and gridded DUACS sea level anomalies for a period centered on 20<sup>th</sup> November 2005. As shown in Figure 8.4, a build-up of high sea level anomalies to the east of Thailand and Malaysia can be observed in the along track RADS data, which reaches peak values of above 0.5 m in the period between 20<sup>th</sup> and 23<sup>rd</sup> November 2005. However, these high

sea level anomalies are less pronounced from the gridded DUACS data as shown in Figure 8.5. The maximum sea level anomalies to the east of Malaysia on 23<sup>rd</sup> November 2005 are of the order 20 to 30 cm, which are considerably below values from along track RADS data.

As discussed in Section 8.2.1, the along track RADS data represent the actual altimeter records, whereas for the gridded DUACS data the records have been interpolated in both space and time, leading to smoothing of the sea level anomaly signal. In consequence, along track RADS data are more representative for the actual along track state at a specific time, while the gridded DUACS data provide a more synoptic overview.

### **8.2.5 Altimeter SLA vs. In-situ SLA**

To compare the time series of RADS and DUACS altimeter sea level anomalies with UHSLC in-situ sea level anomalies, the along track RADS data from Jason-1 and Envisat are selected at cross-over locations in the vicinity of the UHSLC station, the gridded DUACS data are interpolated to the location of interest, and the UHSLC in-situ data are smoothed using a 24 hours moving average filter to remove the higher frequency noise.

Examples of comparison results are shown in Figures 8.6 and 8.7. A reasonable fit between the different sea level anomalies can be observed. The RMS differences between the altimeter data and in-situ data are 0.07 m and 0.1 m at station Cendering/320, and increase to 0.1 m and 0.13 m at station Kelang/140. Attributing to the different data selection and processing procedure applied, the overall RMS differences for DUACS sea level anomalies are about 3 cm below the RMS differences for RADS data.

### **8.3 Assimilation of Sea Level Anomalies into Singapore Regional Model**

As shown in Figure 8.8, the Singapore Regional Model has 3 open boundaries located in 3 sea basins with a total of 17 boundary support points. At the boundary support points along the open boundaries, the correct model forcing of actual water levels (AWL) should be prescribed as,

$$AWL(t) = MDT + TWL(t) + SLA(t) \quad (8.4)$$

where the mean dynamic topography (MDT) signifies the stationary part of the sea surface topography, related to long term ocean circulation, meteorological forcing and geostrophy, the tidal water levels (TWL) are represented by a set of tidal constituents, the sea level anomaly (SLA) is the non-tidal, largely monsoon driven sea level variations.

In the original set-up of the Singapore Regional Model, the forcing terms of mean dynamic topography and tidal water levels are included. To assimilate the sea level anomalies into the Singapore Regional Model and assess the effect on the water dynamics from the sea level anomalies, this section will apply multilayer perceptron to predict the sea level anomaly time series at the open boundaries with prediction horizon  $T = 24$  hours, and then impose the sea level anomalies predicted at the open boundaries as driving force to the numerical model with the hope to numerically model the sea level anomalies inside the model domain.

#### **8.3.1 Prediction of SLA at Open Boundaries**

##### **8.3.1.1 Preprocess of SLA Time Series**

Due to the sparse spatial distribution of the in-situ measurement stations, the prediction of sea level anomalies will be based on the gridded DUACS sea level anomaly data sets. Although the prescription of actual sea level anomalies at the open boundaries is required for simulating the full dynamics of barotropic motion, prescription of DUACS sea level anomalies at the open boundaries can serve as a first proxy to demonstrate the process and assess the efficiency of data assimilation.

Altimeter DUACS sea level anomaly data in 2005 are used to obtain the hourly sea level anomaly time series at the 17 boundary support points of the Singapore Regional Model. The DUACS data are processed in the following way,

- Linear interpolation of the gridded DUACS sea level anomalies to the boundary support points.
- Spline interpolation of the weekly DUACS sea level anomalies into hourly sea level anomaly time series.

Resulting sea level anomalies at one support point of each Singapore Regional Model boundary are illustrated in Figure 8.9. The asterisks indicate the weekly DUACS sea level anomalies, whereas the continuous lines indicate the hourly sea level anomaly time series. Figure 8.9 clearly reveals the different cycles of sea level anomalies in representative locations along the three boundaries.

### **8.3.1.2 Methodology**

As concluded in Chapter 6, multilayer perceptron is preferable over local model to predict the chaotic time series. This section will therefore apply multilayer perceptron to

the prediction of sea level anomalies. The hourly sea level anomaly time series are divided into three subsets, i.e. training data (from 1 to 3000), cross validation data (from 3001 to 4000), and testing data (from 4001 to 8760). Training data are used to train the network in adjusting the synaptic weights, cross validation data are used for avoiding overfitting, whereas the performance of multilayer perceptron is tested on the remaining testing data.

Training multilayer perceptron to approximate the mapping function  $g_T(\cdot)$  in  $x_{t+T} = g_T(\mathbf{x}_t)$  requires feeding the components of  $\mathbf{x}_t = (x_t, x_{t-\tau}, \dots, x_{t-(m-1)\tau})$  as input to the network with  $x_{t+T}$  being the desired response. Therefore, as shown in Figure 8.10, multilayer perceptron is composed of an input layer with  $m$  input neurons, a hidden layer, and an output layer with one output neuron. The optimal embedding parameters  $m$  and  $\tau$  are determined by applying the average mutual information and the false nearest neighbors analyses on the training data and cross validation data. For the support points at the SCS boundary (ID 9), the Andaman Sea boundary (ID 4), and the Java Sea boundary (ID 15), the optimized  $(m, \tau)$  are respectively (3, 6), (2, 15) and (2, 6). Parameters for the networks and in the training algorithm are set as follows

- number of hidden neurons: 50
- number of epochs: 200
- learning rate  $\eta = 0.1$
- momentum constant  $\alpha = 0.7$

To avoid converging to local optima, a set of 5 multilayer perceptrons with different initial synaptic weights is trained. The network with the best prediction accuracy in the cross validation data set is considered to be optimal, which is then applied to predict the testing data set.

### 8.3.1.3 Results

Figures 8.11 to 8.13 illustrate examples of sea level anomaly prediction with multilayer perceptron at the SCS, the Andaman Sea, and the Java Sea boundary support points with prediction horizon  $T = 24$  hours. The 24-hour multilayer perceptron forecast successfully captures the up and down trends of the sea level anomaly time series, producing residues with significantly reduced magnitudes. Compared to the root mean squares of the sea level anomaly time series, the root mean squares of the residues at the 3 boundary support points are respectively reduced from 5.74 cm, 5.14 cm, and 3.44 cm to 0.38 cm, 0.42 cm, and 0.26 cm.

### 8.3.2 Numerical Simulation of Internal SLA

Having predicted the sea level anomalies at the open boundaries, the effect of the sea level anomalies on the water dynamics inside the model domain can be assessed by applying the predicted sea level anomalies at the open boundaries as forcing terms to the numerical model.

Figure 8.14 plots the numerically simulated sea level anomalies (red line) at Tanjong Pagar, together with the DUACS sea level anomalies (blue asterisks). Although the



simulated sea level anomalies fluctuate with high frequencies, the trend of the DUACS sea level anomalies is successfully captured in the numerical simulation Figure 8.15 shows sea level anomaly maps from the numerical simulation (left panels) in selected months compared to sea level anomaly maps obtained from the DUACS data sets (right panels). A reasonable correspondence can be observed between the modelled sea level anomalies and the DUACS sea level anomalies, verifying that the internal sea level anomalies can be decently modelled through numerical simulation provided that the sea level anomalies are properly prescribed at the open boundaries.

#### **8.4 Research in Progress and Future**

Chapter 8 proposed an effective method to predict the sea level anomalies by means of data assimilation. This data assimilation procedure is validated to be operational in real time. At the open boundaries of the Singapore Regional Model, the sea level anomaly time series are predicted using multilayer perceptron with prediction horizon  $T = 24$  hours. Imposing the sea level anomalies predicted at the open boundaries as driving force to the Singapore Regional Model numerically models the sea level anomalies inside the model domain.

Altimetry provides global data with good spatial resolution, but sparse in time. In-situ measurements are spatially sparse, but with good temporal resolution, hourly or better. As a demonstration of the proposed data assimilation scheme, the sea level anomaly time series herein analyzed are the spatially and temporally interpolated DUACS sea level anomalies. To predict the real sea level anomalies at the open boundaries and to force the

Singapore Regional Model with real sea level anomalies using the complementary features of the altimeter data and in-situ measurements will be a direct extension of present work.

In a hydrodynamic model, the governing equations correlate variations of currents with water level gradients and vice versa. This inspires a hypothesis for future study: a major contribution to the current anomalies results from sea level anomaly gradients. Although some research indicates the current anomalies and the sea level anomaly gradients are fairly correlated based on limited in-situ measurements (Calkoen et al., 2009, Gerritsen et al., 2009), to further validate this hypothesis based on numerical modelling is an important subject requiring additional investigation.

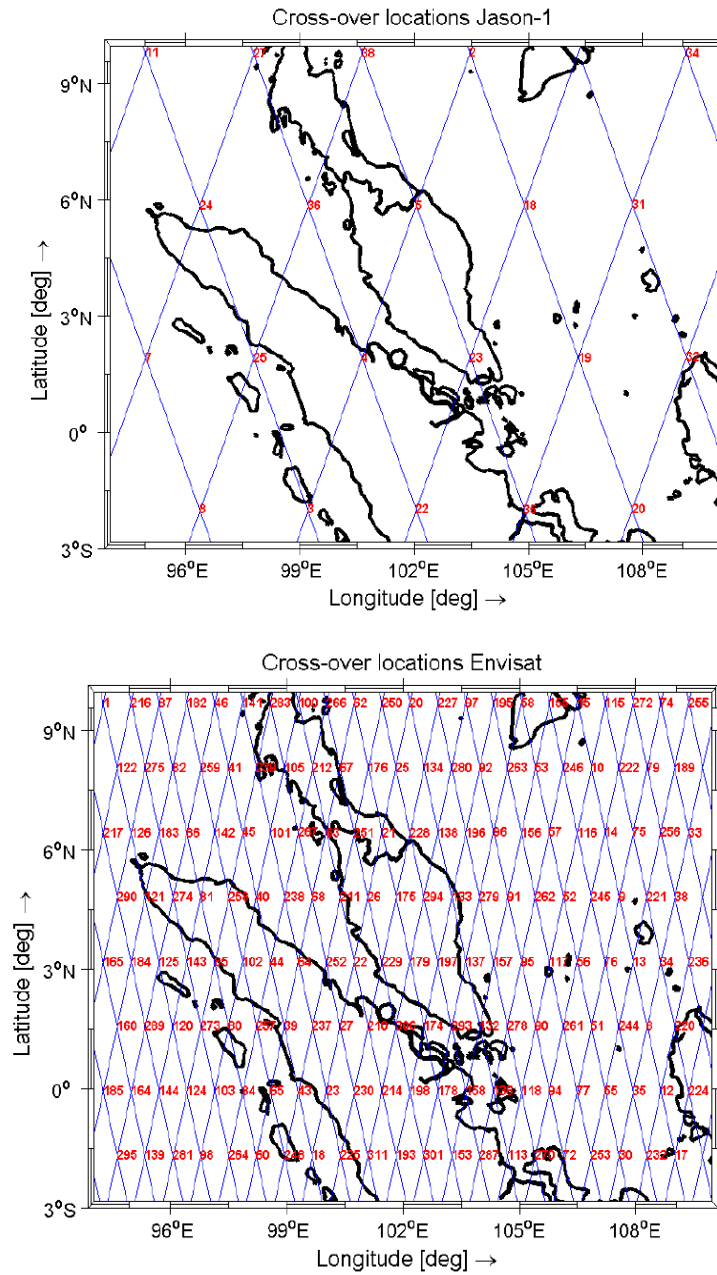


Figure 8.1 Jason-1 (upper) and Envisat (lower) ground tracks (cross-over points numbered for data processing).

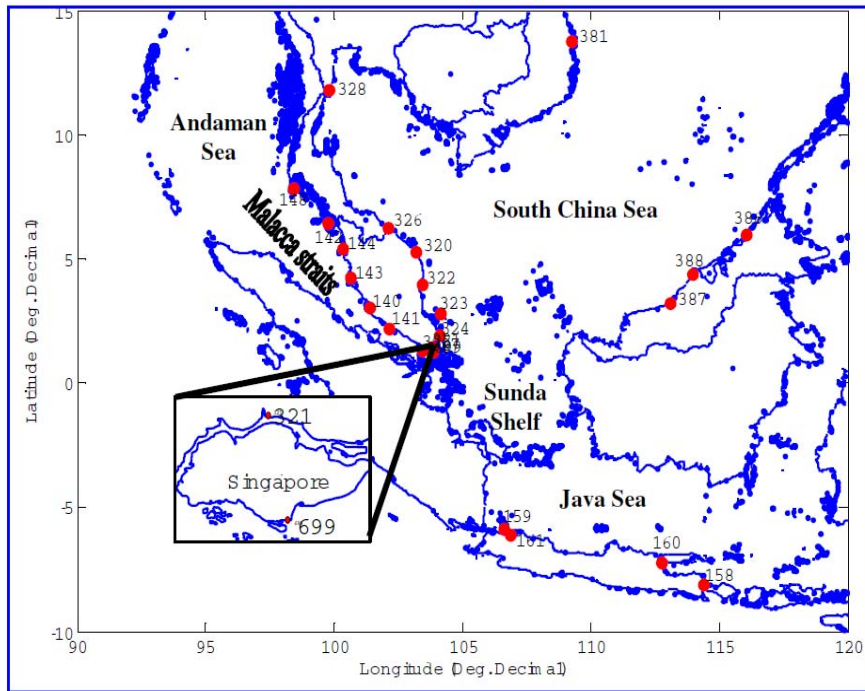


Figure 8.2 Locations of the UHSLC stations.

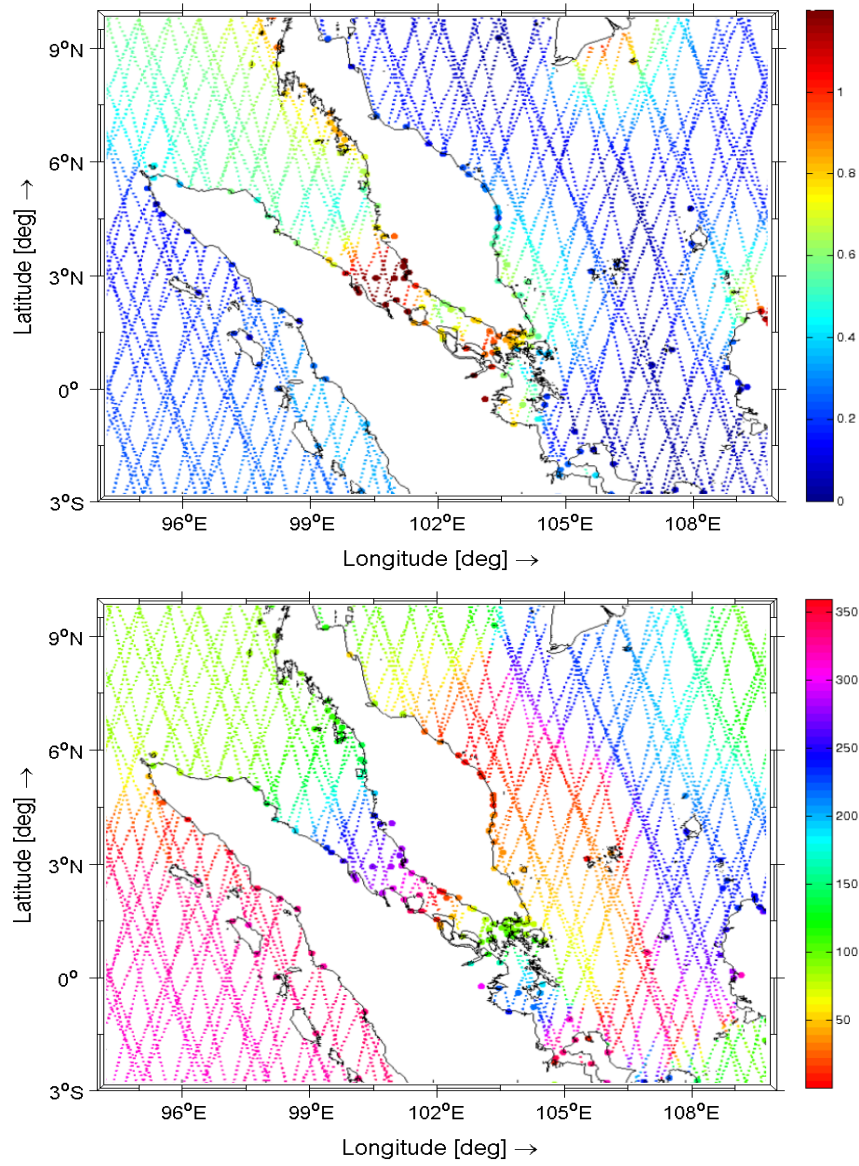


Figure 8.3 Amplitudes (upper) and phases (lower) of M2 from RADS altimeter data and from in-site measurements.

## CHAPTER 8. USE OF DATA ASSIMILATION IN UNDERSTANDING SEA LEVEL ANOMALIES

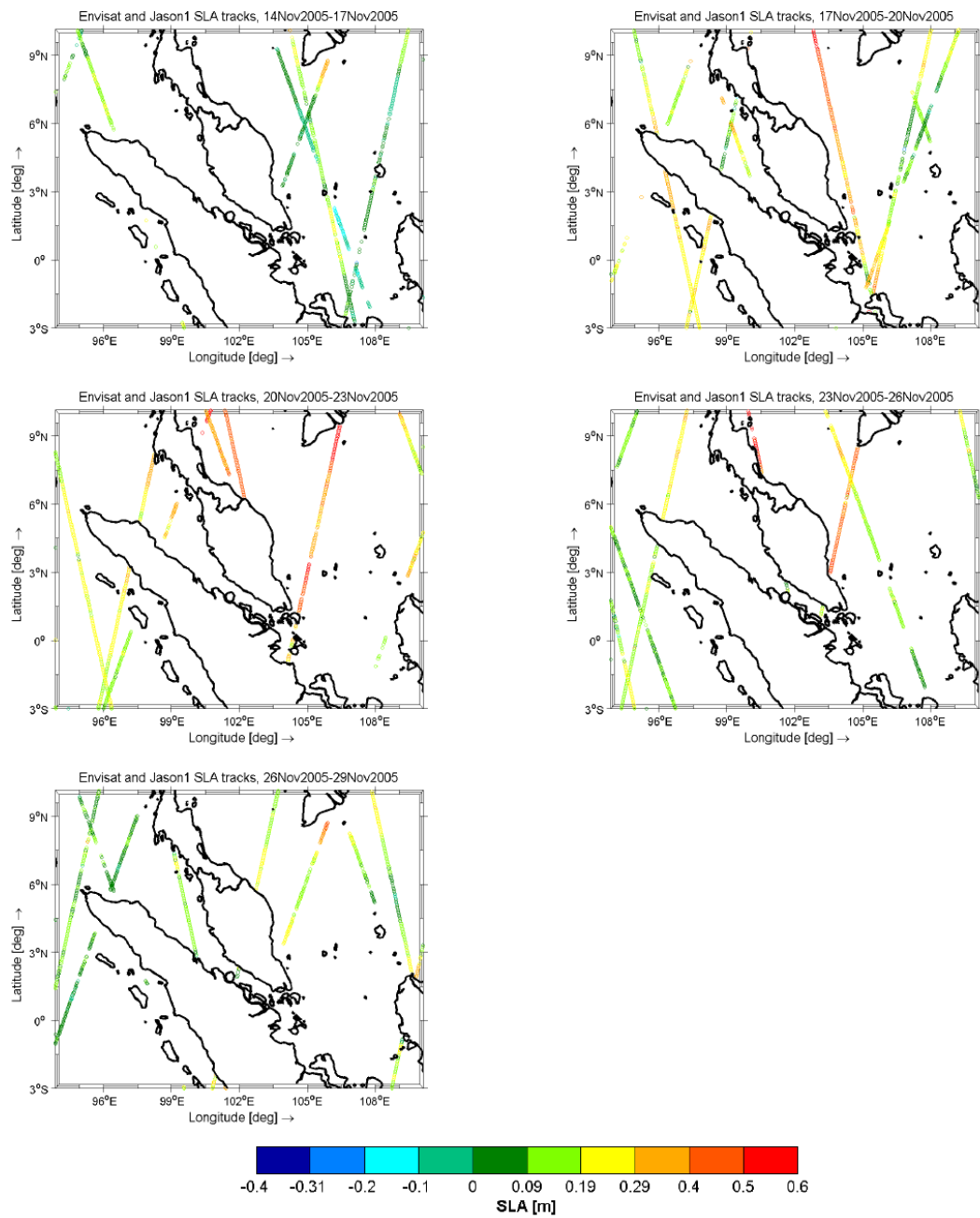


Figure 8.4 Along track RADS sea level anomalies for period from 14<sup>th</sup> to 29<sup>th</sup> November 2005.

CHAPTER 8. USE OF DATA ASSIMILATION IN UNDERSTANDING  
SEA LEVEL ANOMALIES

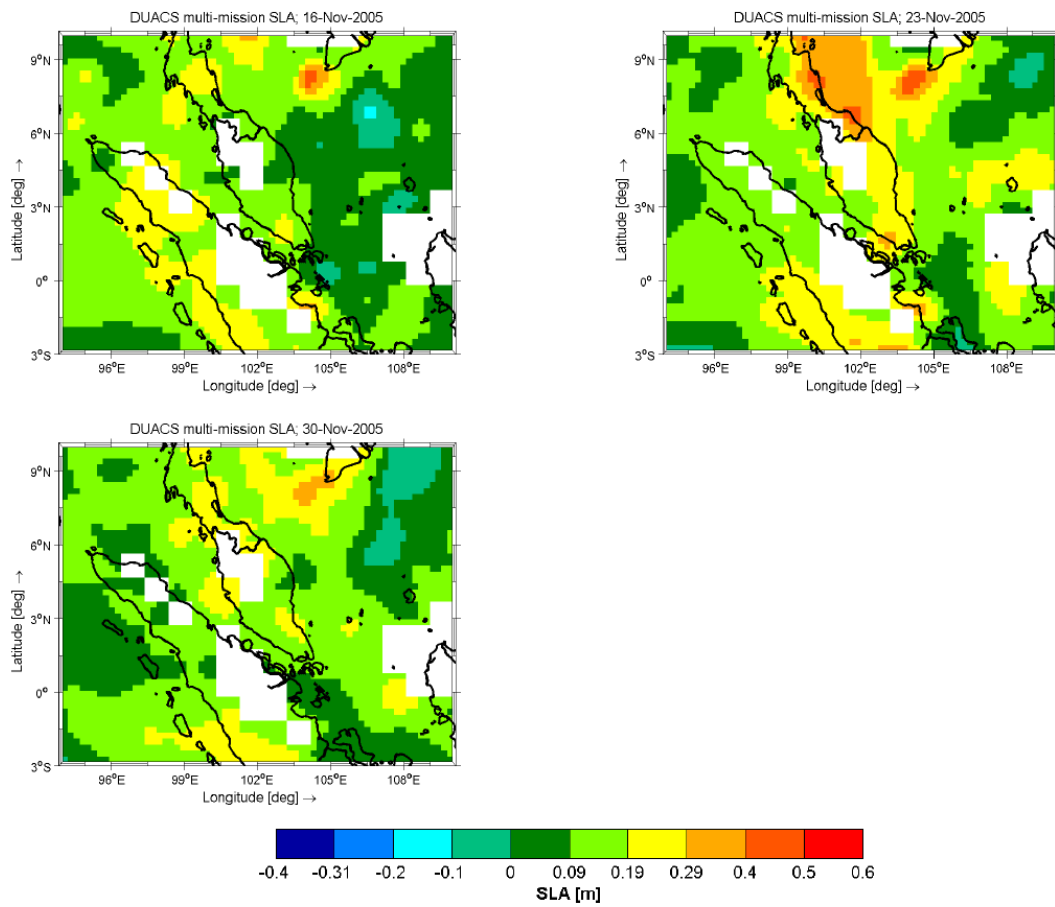


Figure 8.5 Gridded DUACS sea level anomalies for period from 16<sup>th</sup> to 30<sup>th</sup> November 2005.

## CHAPTER 8. USE OF DATA ASSIMILATION IN UNDERSTANDING SEA LEVEL ANOMALIES

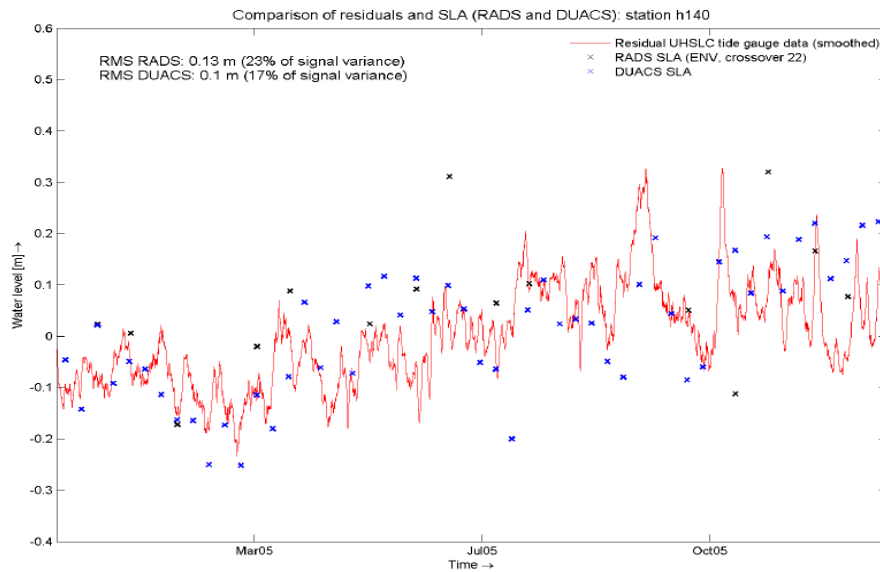


Figure 8.6 Comparison of sea level anomalies obtained from the RADS and DUACS data sets with sea level anomalies obtained from UHSLC in-situ measurements (Kelang/140; 2005).

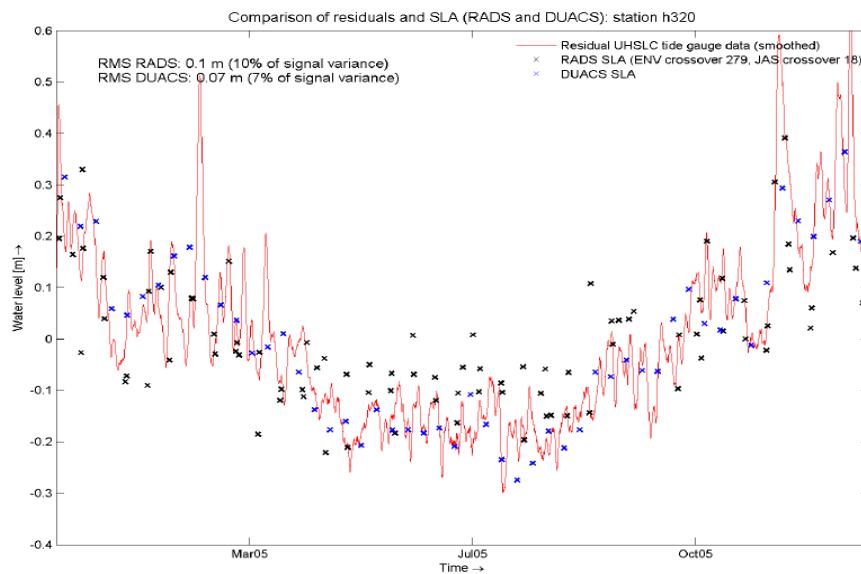


Figure 8.7 Comparison of sea level anomalies obtained from the RADS and DUACS data sets with sea level anomalies obtained from UHSLC in-situ measurements (Cendering /320; 2005).



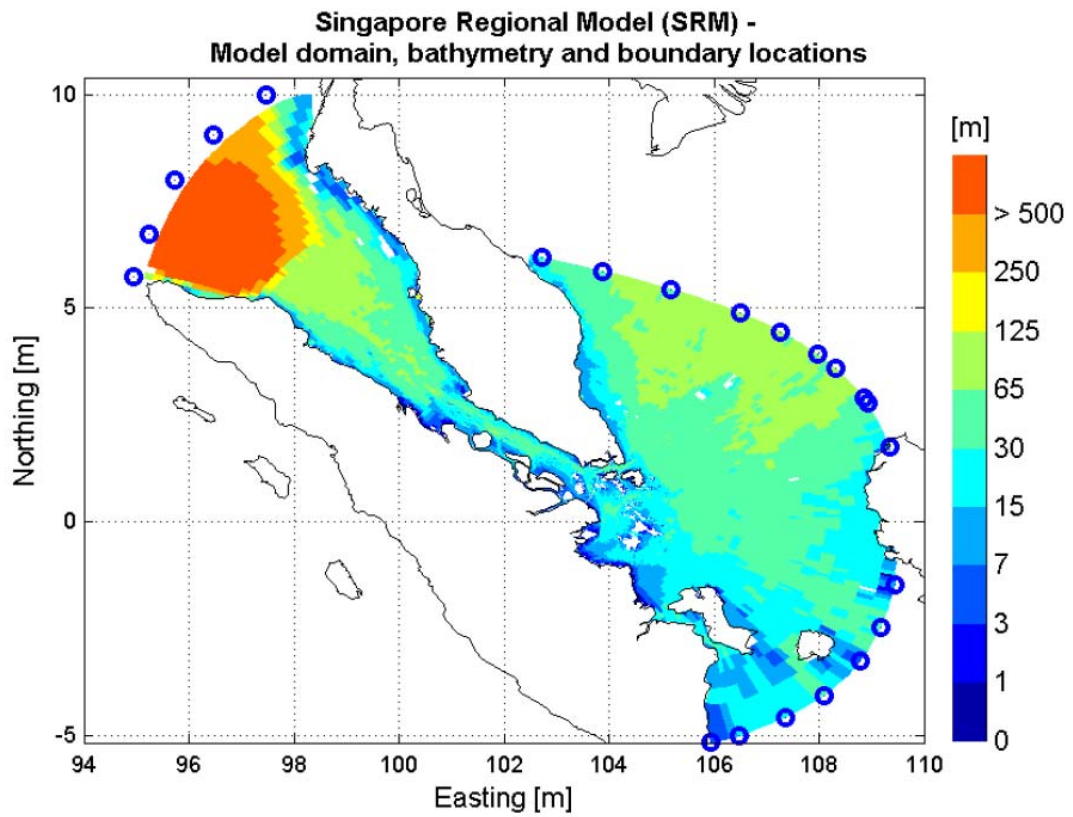


Figure 8.8 Extent, bathymetry of the Singapore Regional Model with 17 boundary support points.

## CHAPTER 8. USE OF DATA ASSIMILATION IN UNDERSTANDING SEA LEVEL ANOMALIES

---

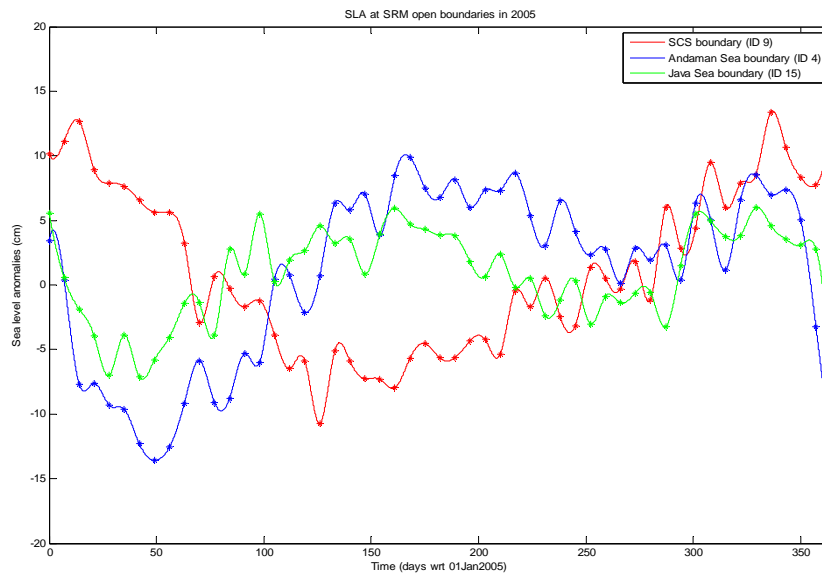


Figure 8.9 Extracted SLA at selected Singapore Regional Model SCS, Andaman Sea, and Java Sea boundary support points.

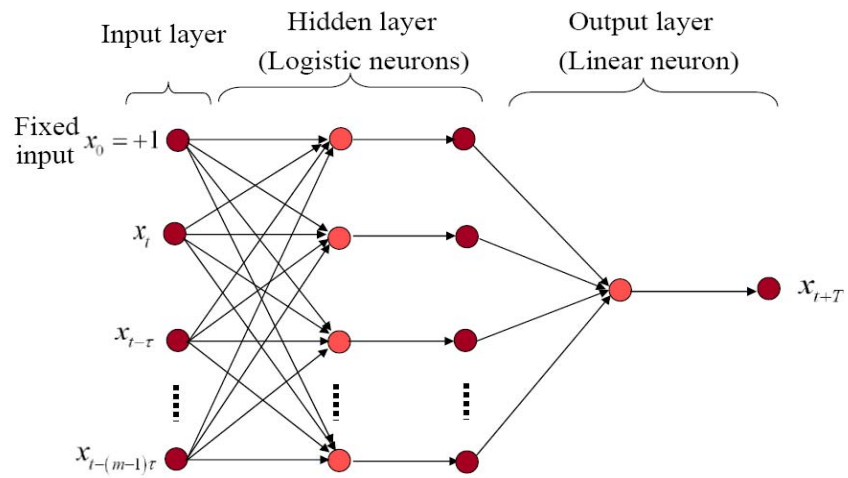


Figure 8.10 Architecture of multilayer perceptron in sea level anomaly prediction.

## CHAPTER 8. USE OF DATA ASSIMILATION IN UNDERSTANDING SEA LEVEL ANOMALIES

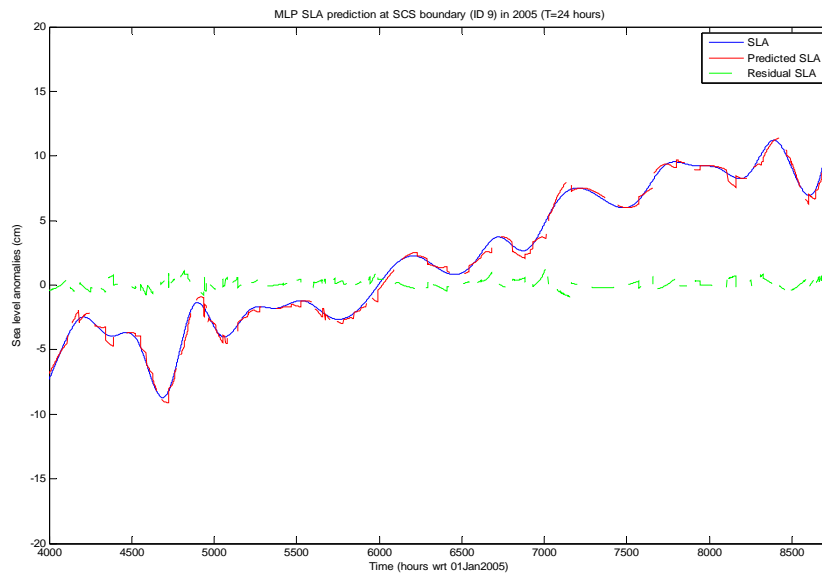


Figure 8.11 SLA prediction with multilayer perceptron at SCS boundary (ID 9; T=24 hours).

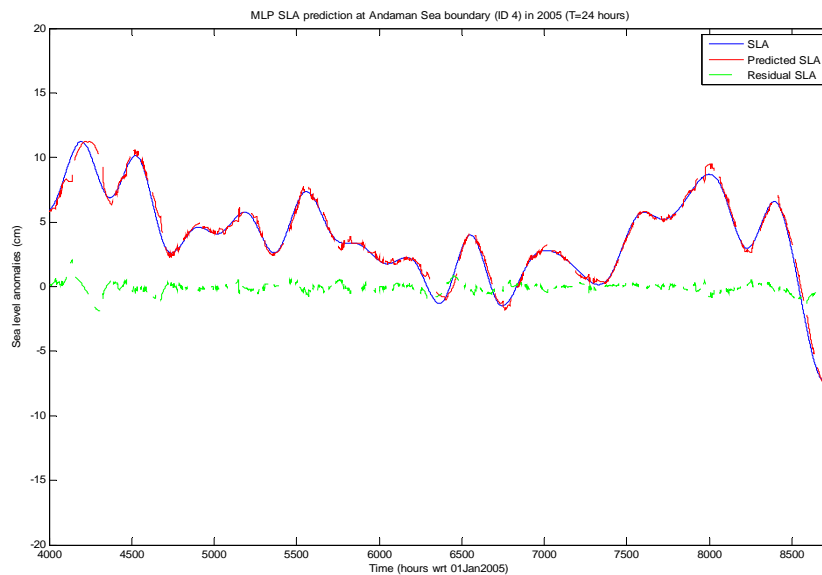


Figure 8.12 SLA prediction with multilayer perceptron at Andaman Sea boundary (ID 4; T=24 hours).

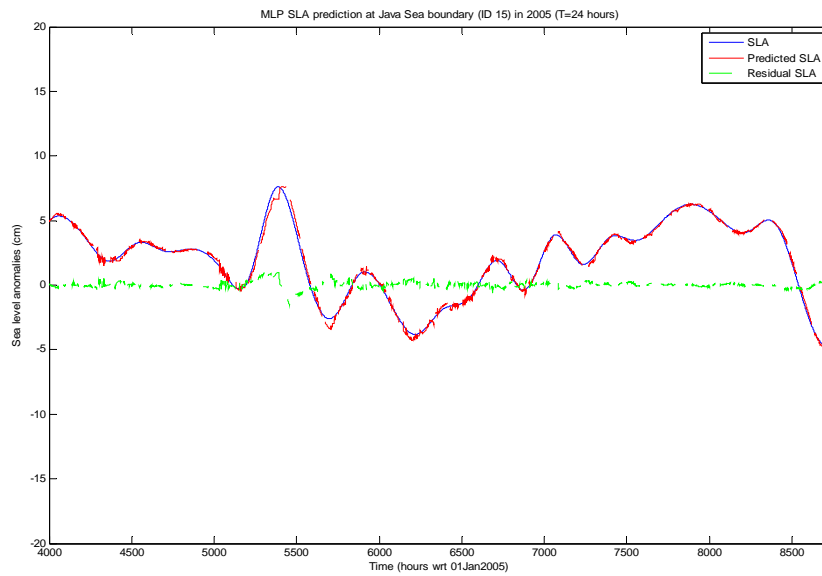


Figure 8.13 SLA prediction with multilayer perceptron at Java Sea boundary (ID 15; T=24 hours).

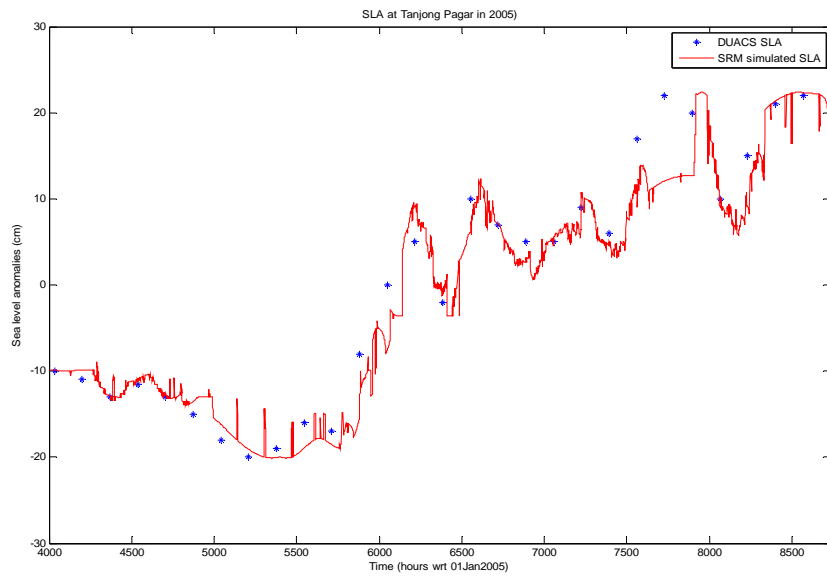
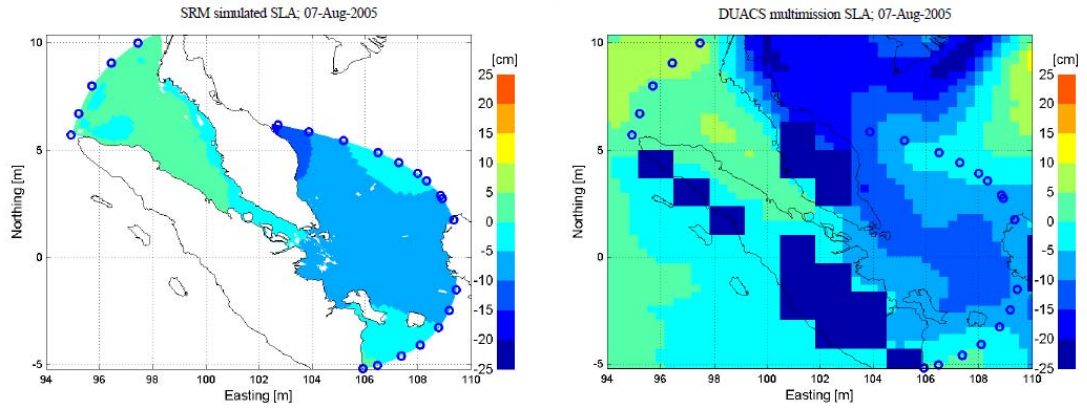
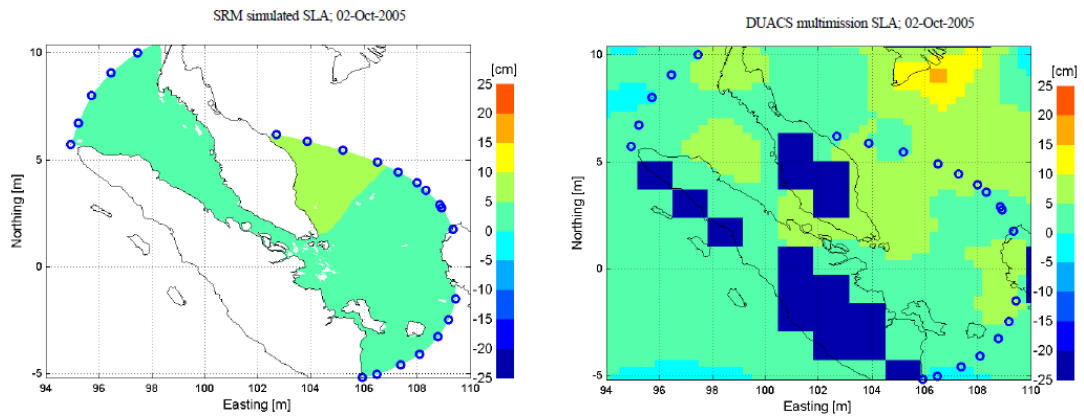


Figure 8.14 SRM simulated SLA (red line) compared to DUACS SLA (blue asterisks) at Tanjung Pagar.

### August SLA comparison



### October SLA comparison



### December SLA comparison

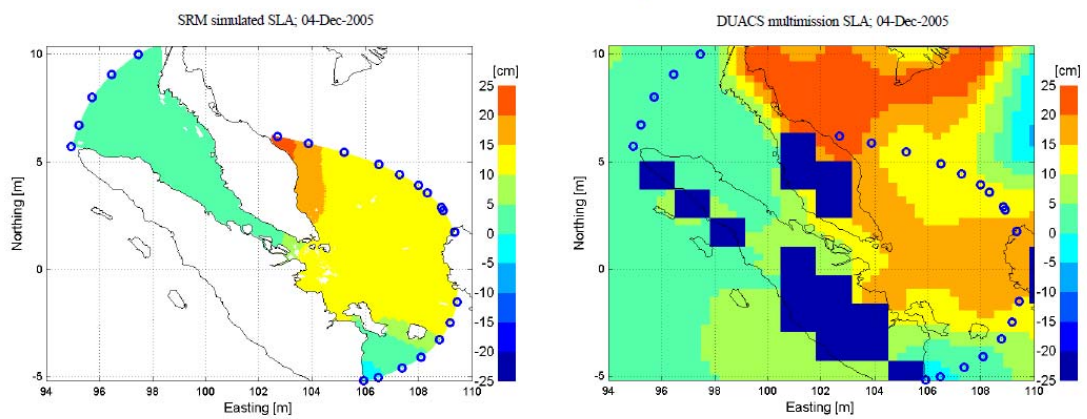


Figure 8.15 SRM simulated SLA (left panels) compared to DUACS SLA maps (right panels).

Table 8.1. General aspects of Jason-1 and Envisat.

	Jason-1	Envisat
Repeat cycle	9.9156 days	35 days
Number of passes per cycle	254	501
Ground track separation at Equator	315 km	85 km



CHAPTER 8. USE OF DATA ASSIMILATION IN UNDERSTANDING  
SEA LEVEL ANOMALIES

---

Table 9.2 Summary of statistical analysis results of the sea level anomalies.

Station name/code	Lat/long	Mean	Max	Min	RMS
Kelang/140	3.050/101.367	-3.19E-09	0.6221	-0.6498	<b>0.1691</b>
Keling/141	2.217/102.150	3.42E-09	0.2839	-0.235	0.0652
Langkawi/142	6.433/99.767	3.42E-10	0.3851	<b>-0.7854</b>	0.062
Lumut/143	4.233/100.617	-1.14E-10	0.2824	-0.2178	0.0595
Penang/144	5.417/100.350	-1.37E-09	0.5702	-0.5422	0.1267
Ko Taphao Noi/148	7.833/98.433	-3.98E-09	0.3646	-0.4919	0.0658
Surabaya/160	-7.217/112.733	-1.87E-09	0.6432	-0.5813	0.1297
Cendering/320	5.267/103.183	8.54E-09	0.533	-0.2721	0.0917
Johor Baharu/321	1.467/103.800	-4.02E-09	0.5473	-0.3155	0.0877
Kuantan/322	3.983/103.433	1.59E-09	0.6276	-0.5009	<b>0.1706</b>
Tioman/323	2.800/104.133	-3.01E-18	0.5719	-0.2799	0.0831
Sedili/324	1.933/104.117	-2.73E-09	<b>0.7996</b>	-0.5195	0.1652
Kukup/325	1.333/103.450	-2.96E-09	0.4055	-0.2736	0.0699
Getting/326	6.233/102.100	3.30E-09	0.5045	-0.3325	0.1023
Ko Lak/328	11.800/99.817	2.73E-09	0.5056	-0.4865	0.1181
Bintulu/387	3.217/113.067	3.76E-09	<b>0.1908</b>	<b>-0.1726</b>	<b>0.0478</b>
Tanjong Pagar/699	1.267/103.850	-8.99E-10	0.6342	-0.35	0.1042

## **Chapter 9**

### **Conclusions and Recommendations**

#### **9.1 Conclusions**

One primary objective of this study is to develop and implement applicable data assimilation methods to improve the forecasting accuracy of the Singapore Regional Model. A novel hybrid data assimilation scheme is proposed, which assimilates the observed data into the numerical model in two steps: (i) predicting the model errors at the measurement stations, and (ii) distributing the predicted errors to the non-measurement stations. Specifically, three approaches are studied, the local model approach (LM), the multilayer perceptron (MLP), and the Kalman filter (KF).

At the stations where observations are available, both the local model approach and the multilayer perceptron are utilized to forecast the model errors based on the patterns revealed in the phase spaces reconstructed by the past recordings. In cases of smaller prediction horizons, such as  $T = 2, 24$  hours, the local model approach outperforms the multilayer perceptron. However, due to the local model approach is less competent to capture the trajectories of the state vectors in the higher-dimensional phase spaces, the prediction accuracy of the local model approach decreases by a wider margin when  $T$

progresses to 48, 96 hours. Averaged over 5 different prediction horizons, both methods are able to remove more than 60% of the root mean square errors in the model error time series, while the multilayer perceptron performs slightly better.

To extend the updating ability to the remainder of the model domain, Kalman filter and the multilayer perceptron are used to spatially distribute the predicted model errors to the non-measurement stations. When the outputs of the Singapore Regional Model at the non-measurement stations and the measurement stations are highly correlated, such as at Bukom and Raffles, both approaches exhibit remarkable potentials of distributing the predicted errors to the non-measurement stations, resulting in an error reduction of more than 50% on average. However, the performance of Kalman filter in error distribution deteriorates at a rapid pace when the correlation decreases, with only about 40% of the root mean square errors removed at Sembawang and 20% at Horsburgh. Comparatively, the multilayer perceptron is less sensitive to the correlations with a more consistent performance, which removes more than 40% of the root mean square errors at Sembawang and Horsburgh. In addition, the error distribution study demonstrates for the first time that distributing the predicted errors from more measurement stations does not necessarily produce the best results due to the misleading information from less correlated stations. As suggested by this finding, to conduct a prior correlation analysis among possible sites is favorable when planning the future layout of the measurement stations.

Another major objective of this study is to analyze and predict the sea level anomalies by means of data assimilation. Sea level anomalies are extracted based on tidal analysis

from both altimeter data and in-situ measurements. A reasonable fit between the altimeter sea level anomalies and the in-situ sea level anomalies can be observed, indicating the coherence and consistency of different data sources. As a demonstration of the proposed data assimilation scheme, the sea level anomalies explored in this study are the spatially and temporally interpolated DUACS sea level anomalies.

At the open boundaries of the Singapore Regional Model, the sea level anomaly time series are predicted using multilayer perceptron with prediction horizon  $T = 24$  hours. Multilayer perceptron successfully captures the motion dynamics of the sea level anomalies, with more than 90% of the root mean squares removed on average. The sea level anomalies inside the model domain are then numerically modelled by imposing the sea level anomalies predicted at the open boundaries as driving force to the Singapore Regional Model. A reasonable correspondence are observed between the modelled sea level anomalies and the DUACS sea level anomalies, verifying that the internal sea level anomalies can be decently modelled through numerical simulation provided that the sea level anomalies are properly prescribed at the open boundaries.

## 9.2 Recommendations

Considering the data assimilation techniques are still at a developing stage and some study on sea level anomalies is still ongoing, the following aspects are recommended for future research.

- Analogous to the local model approach, multilayer perceptron can be coupled with genetic algorithm to determine simultaneously both the embedding parameters  $(m, \tau)$

and the parameters of the network and in the training algorithm. The performance of multilayer perceptron in model error and sea level anomaly prediction will be further improved in all probability with these optimal parameters determined.

- The assumptions about exponential and diagonal structures of model and measurement error covariances facilitate the application of Kalman filter in error distribution, but they may be too simplified to represent the real error covariances and hence limit the performance of Kalman filter. Therefore, to pursue a better substitute for error covariances is of necessity in the future research.
- Multilayer perceptron is a traditional type of artificial neural networks. Other categories of artificial neural networks, such as time-lag recurrent network (TLRN), radial basis function network (RBFN) and support vector machines (SVM), have developed rapidly and shown improved promise in pattern recognition and function approximation. Therefore, to investigate the applicability of the newly developed networks in data assimilation is another possible avenue of future work.
- Future studies should place stress on utilizing the complementary features of the altimeter data and in-situ measurements, to force the Singapore Regional Model with updated real sea level anomalies.
- More tests should be conducted in future to validate the hypothesis on the correlations between the sea level anomaly gradients and current anomalies based on numerical modelling.

## References

1. Abaranel, H. D. I., 1996. *Analysis of Observed Chaotic Data*. Springer-Verlag, New York.
2. Alligood, K., Sauer, T. and Yorke, J. A., 1997. *CHAOS: An Introduction to Dynamical Systems*. Springer-Verlag, New York.
3. Babovic, V., 2007. Data-model integration – An approach to assimilation of sea level anomaly data into an oceanographic model. In *Numerical Modelling of Hydrodynamics for Water Resources*, Taylor & Francis, London, 67-76.
4. Babovic, V. and Keijzer, M., 1999. Forecasting of river discharges in the presence of chaos and noise. In *Coping with Floods: Lessons Learned from Recent Experiences*, Kluwer, Dordrecht, 405-420.
5. Babovic, V., Cañizares, R., Jensen, H. R. and Kling, A., 2001. Neural networks as routine for error updating of numerical models. *Journal of Hydraulic Engineering*, 127, 181-193.
6. Babovic, V., Keijzer, M. and Stefannson, M., 2000. Optimal embedding using evolutionary algorithms. *Proceedings of the 4th International Conference on Hydroinformatics*, Iowa City.
7. Babovic, V., Sannasiraj, S. A. and Chan, E. S., 2005. Error correction of a predictive ocean wave model using local model approximation. *Journal of Marine Systems*, 53, 1-17.
8. Box, G. E. P. and Jenkins, G. M., 1976. *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco.
9. Calkoen, C., Twigt, D. J., Sisomphon, P., Wensink, H. and Mynett, A. E., 2009. Sea level anomalies from satellite altimetry – retrieval and validation. *Proceedings of the 8th Internal Conference on Hydroinformatics*, Concepción.

- 
10. Cañizares, R., 1999. *On the Application of Data Assimilation in Regional Coastal Models*. Ph.D. Thesis, Delft University of Technology, Delft.
  11. Cañizares, R., Heemink, A. W. and Vested, H. J., 1998. Application of advanced data assimilation methods for the initialization of storm surge models. *Journal of Hydraulic Research*, 36(4), 655-674.
  12. Cañizares, R., Madsen, H., Jensen, H. R. and Vested, H. J., 2001. Developments in operational shelf sea modelling in Danish waters. *Estuarine Coastal and Shelf Science*, 53, 595-605.
  13. Caputo, J. G. and Atten, P., 1986. Metric entropy: an experimental means for characterizing and quantifying chaos. *Physical Review A*, 35(3), 1311-1316.
  14. Chang, F. J., Chiang, Y. M. and Chang, L. C., 2007. Multi-step-ahead neural networks for flood forecasting. *Hydrological Sciences Journal*, 52(1), 114-130.
  15. Cipolla, E. T., 1995. Data mining: techniques to gain Insight into your data. *Enterprise Systems Journal*, 65, 18-24.
  16. Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2, 303-314.
  17. Daley, R., 1991. *Atmospheric Data Analysis*. Cambridge University Press, Cambridge.
  18. Deltares, 2009. *User Manual Delft3D-FLOW; Version 3.14*. Delft.
  19. Ding, G., Zhong, S. S. and Li, Y., 2008. Time series prediction using wavelet process neural network. *Chinese Physics B*, 17(6), 1998-2003.
  20. Evensen, G., 1994. Sequential data assimilation with a non-linear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5), 10143-10162.
  21. Evensen, G., 2003. The Ensemble Kalman Filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53, 343-367.
  22. Farmer, J. D. and Sidorowich, J. J., 1987. Predicting chaotic time series. *Physical Review Letters*, 59(8), 845-848.
  23. Fraser, A. M. and Swinney, H. L., 1989. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2), 1134-1140.

24. Gallager, R. G., 1968. *Information Theory and Reliable Communication*. John Wiley and Sons, New York.
25. Gerritsen, H., Calkoen, C., Twigt, D. J., Babovic, V. and Mynett, A. E., 2009. MHBox – analysis and prediction of sea level anomalies and associated currents in Singapore and Malacca Straits. *Proceedings of the 8th International Conference on Hydroinformatics*, Concepción.
26. Gerritsen, H., Schrama, E. J. O., Uittenbogaard, R. E., Boogaard van den, H. F. P. and Kaaij van der, T., 2000. *SAT2SEA*. Research Report, Delft Hydraulics, Delft.
27. Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Boston.
28. Grassberger, P., 1986. Do climatic attractors exist? *Nature*, 323, 609-612.
29. Grassberger, P. and Procaccia, I., 1983a. Estimation of the Kolmogorov entropy from a chaotic signal. *Physical Review A*, 28(4), 2591-2593.
30. Grassberger, P. and Procaccia, I., 1983b. Characterization of strange attractors. *Physical Review Letters*, 50(5), 346-349.
31. Grassberger, P. and Procaccia, I., 1983c. Measuring the strangeness of strange attractors. *Physica D*, 9(1-2), 189-208.
32. Hartnack, J. and Madsen, H., 2001. Data assimilation in river flow modelling. *Proceedings of the 4th DHK Software Conference*, Helsingør.
33. Haugen, V. E. J. and Evensen, G., 2002. Assimilation of SLA and SST data into an OGCM for Indian Ocean. *Ocean Dynamics*, 52(3), 133-151.
34. Haykin, S., 1999. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, New Jersey.
35. Haykin, S., 2001. *Kalman Filtering and Neural Networks*. John Wiley & Sons, New York.
36. Holland, J. H., 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Michigan.
37. Hong, T. P., Horng, C. Y., Wu, C. H. and Wang, S. L., 2009. An improved data mining approach using predictive itemsets. *Expert Systems with Applications*, 36(1), 72-80.



- 
38. Hopfield, J. J., 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79, 2554-2558.
  39. Hydrographic Dept. of Singapore, 1999. *Singapore Tide Tables and Port Information*. ISSN: 0219-0282, Singapore.
  40. IAHR Bulletin, 1994. *Guidelines for Validation Documents of the International Association for Hydraulic Research*. International Association for Hydraulic Research Bulletin, 24 pages.
  41. Jayawardena, A. W. and Lai, F., 1994. Analysis and prediction of chaos in rainfall and stream flow time series. *Journal of Hydrology*, 153, 23-52.
  42. Kalman, R. E., 1960. A new approach to linear filter and prediction theory. *Journal of Basic Engineering*, 82(D), 35-45.
  43. Kamath, C., 2006. Mining science data. *Journal of Physics: Conference Series*, 46, 500-504.
  44. Kantha, L. H. and Clayson, C. A., 1999. *Numerical Models of Oceans and Oceanic Processes*, Academic Press, San Diego.
  45. Kantz, H. and Schreiber, T., 2004. *Nonlinear time series analysis*, Cambridge University Press, Cambridge.
  46. Karunasinghe, D. S. K. and Liong, S. Y., 2006. Chaotic time series prediction with a global model: Artificial Neural Network. *Journal of Hydrology*, 323(1-4), 92-105.
  47. Kennel, M. B., Brown, R. and Abarbanel, H. D. I., 1992. Determining minimum embedding dimension using a geometrical construction. *Physical Review A*, 45, 3403-3411.
  48. Kernkamp, H. and Zijl, F., 2004. *Further Hydraulic model Studies for Pulau Ubin & Pulau Tekong reclamation scheme*. Interim Report on Hydrodynamic Modelling – Model Set-up and Calibration, Delft Hydraulics, Delft.
  49. Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.
  50. Le Dimet, F. X. and Talagrand, O., 1986. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus*, 38(A), 97-110.

- 
51. Leendertse, J. J. and Gritton, E. C., 1971. *A Water-Quality Simulation Model for Well Mixed Estuaries and Coastal Seas: Vol. II, Computation Procedures*. Rand Report, Rand Corporation, Santa Monica.
  52. Leendertse, J. J., 1967. *Aspects of a Computational Model for Long-Period Water-Wave Propagation*. Ph.D. Thesis, Rand Corporation, Santa Monica.
  53. Leendertse, J. J., Alexander, R. C. and Liu, S-K., 1973. *A Three-Dimensional Model for Estuaries and Coastal Seas: Volume I, Principles of Computations; Volume II, Aspects of Computations; Volume III, The Interim Program; Volume IV, Turbulent Energy Computation*. Rand Report, Rand Corporation, Santa Monica.
  54. Liong, S. Y., Phoon, K. K., Pasha, M. F. F. and Doan, D. D., 2005. Efficient implementation of inverse approach for forecasting hydrological time series using micro GA. *Journal of Hydroinformatics*, 7, 151-163.
  55. Lorenz, E. N., 1963. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2), 130-141.
  56. Lorenz, E. N., 1991. Dimension of weather and climate attractors. *Nature*, 353, 241-244.
  57. Luong, B., Blum, J. and Verron, J., 1998. A variational method for the resolution of a data assimilation problem in oceanography. *Inverse Problems*, 14, 979-997.
  58. Mancarella, D., Babovic, V., Keijzer, M. and Simeone, V., 2007. Data assimilation of forecasted errors in Hydrodynamic models using inter-model correlations. *International Journal for Numerical Methods in Fluids*, 56(6), 587-605.
  59. Madsen, H. and Cañizares R., 1999. Comparison of extended and ensemble Kalman filter for data assimilation in coastal area modelling. *International Journal for Numerical Methods in Fluids*, 31(6), 961-981.
  60. Madsen, H., Rosbjerg, D., Damgård, j. and Hansen, F. S., 2003. Data assimilation in the MIKE 11 flood forecasting system using Kalman filtering. *Proceedings of the International Symposium on Water Resource Systems – Hydrological Risk, Management and Development*, Sapporo, 281, 75-81.
  61. Marchuk, G. I., Schröter, J. and Zalesny, V. B., 2003. Numerical study of the global ocean equilibrium circulation. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 18(4), 307-335.
  62. Maybeck, P. S., 1979. *Stochastic Models, Estimation and Control*. Academic Press, New York.

- 
63. McCulloch, W. S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-133.
64. Minsky, M. L. and Papert, S. A., 1969. *Perceptrons*. MIT Press, Cambridge, MA.
65. Mitchell, H. L. and Houtekamer, P. L., 2000. An adaptive ensemble Kalman filter. *Monthly Weather Review*, 228, 416-433.
66. Nechaev, V. and Yaremchuk, M., 1994. Applications of the adjoint technique to processing of a standard section data set: world ocean circulation experiment section S4 along 67°S in the Pacific Ocean. *Journal of Geophysical Research*, 100(C1), 865-879.
67. Oke, P. R., Allen, J. S., Miller, R. N., Egbert, G. D. and Kosro, P. M., 2002. Assimilation of surface velocity data into a primitive equation coastal ocean model. *Journal of Geophysical Research*, 107(C9), 3122, doi: 10.1029/2000JC000511.
68. Ott, E., 1993. *Chaos in Dynamical Systems*. Cambridge University Press, Cambridge.
69. Palacio, C., Winter, C. and Mayerle, R., 2001. Set-up of a hydrodynamic model for the meldorf bight. *Proceedings of the World Water & Environmental Resources Congress*, Orlando.
70. Pasternack, G. B., 1999. Dose the river run wild? Assessing chaos in hydrological systems. *Advances in Water Resources*, 23, 253-260.
71. Pham, D. T., 2000. Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly Weather Review*, 129(5), 1194-1207.
72. Poncelet, P., Masegla, F. and Teisseire, M., 2007. *Data Mining Patterns: New Methods and Applications*. Information Science Reference, Hershey.
73. Powell, M. J. D., 1985. Radial basis functions for multivariable interpolation: a review. *Proceedings of the IMA Conference on Algorithms for the Approximation of Functions and Data*, Shrivensham, 143-167.
74. Pugh, D. T., 1996. *Tides, surges and mean sea-level*. John Wiley & Sons, Chichester.
75. Refsgaard, J. C., 1997. Validation and intercomparison of different updating procedures for real time forecasting. *Nordic Hydrology*, 28, 65-84.
76. Reichle, R. H., 2008. Data assimilation methods in the earth sciences. *Advances in Water Resources*, 31, 1411-1418.

- 
77. Robinson, A. R., Lermusiaux, P. F. J. and Sloan, N. Q., 1998. Data assimilation. *The Sea: The Global Coastal Ocean, Processes and Methods*, John Wiley and Sons, New York, 10, 541-594.
  78. Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386-408.
  79. Rumelhart, D. E., Hinton, G. E. and Williams, R. J., 1986. Learning representations of back-propagation errors. *Nature*, 323, 533-536.
  80. Rumelhart, D. E. and McClelland, J. L., 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge, MA.
  81. Salzman, B., 1962. Finite amplitude free convection as an initial value problem-I. *Journal of the Atmospheric Sciences*, 29, 329-341.
  82. Sannasiraj, S. A., Babovic, V. and Chan, E. S., 2005. Local model approximation in the real time wave forecasting. *Coastal Engineering*, 52(3), 221-236.
  83. Schreiber, T. and Schmitz, A., 1996. Improved surrogate data for nonlinearity tests. *Physical Review Letters*, 77(4), 635-638.
  84. Serafy, G. Y. H. E. and Mynett, A. E., 2008. Improving the operational forecasting system of the stratified flow in Osaka bay using an ensemble Kalman filter-based steady state Kalman filter. *Water Resources Research*, 44(6), W06416, doi: 10.1029/2006WR005412.
  85. Sørensen, J.V.T., Madsen, H. and Madsen, H., 2001. Data assimilation of tidal gauge data in a three-dimensional coastal model. *Proceedings of the 4th DHI Software Conference*, Helsingør.
  86. Sprott, J. C., 2003. *Chaos and Time-Series Analysis*. Oxford University Press, Oxford.
  87. Stelling, G. S., 1983. *On the Construction of Computational Methods for Shallow Water Flow Problems*. Ph.D. Thesis, Delft University of Technology, Delft.
  88. Sun, Y. B., Sisomphon, P., Babovic, V. and Chan E. S., 2009. Comparison of Kalman filter and inter-model correlation method for data assimilation in tidal prediction. *Proceedings of the 8th Internal Conference on Hydroinformatics*, Concepción.
  89. Takens, F., 1981. Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence, Lecture Notes in Mathematics*, Springer-Verlag, Berlin, 898, 366-381.

- 
90. Theiler, J., 1990. Statistical precision of dimension estimators. *Physical Review A*, 41(6), 3038-3050.
  91. Vapnik, V. N., 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.
  92. Verlaan, M., 1998. *Efficient Kalman Filtering Algorithms for Hydrodynamics Models*. Ph.D. Thesis, Delft University of Technology, Delft.
  93. Verlaan, M. and Heemink, A. W., 1997. Tidal flow forecasting using reduced rank square root filters. *Stochastic Hydrology and Hydraulics*, 11, 349-368.
  94. Verlaan, M. and Heemink, A. W., 2001. Non-linearity in data assimilation applications: a practical method for analysis. *Monthly Weather Review*, 129(6), 1578-1589.
  95. Verlaan, M., Zijderfeld, A., de Vries, H. and Kroos, J., 2005. Operational storm surge forecasting in the Netherlands: developments in the last decade. *Philosophical Transactions of the Royal Society A*, 363, 1441-1453.
  96. Wang, X. K., Lu, W. Z., Cao, S. Y. and Fang, D., 2007. Using time-delay neural network combined with genetic algorithms to predict runoff level of Linshan Watershed, Sichuan, China. *Journal of Hydrologic Engineering*, 12(2), 231-236.
  97. Wang, X. Z., 1999. *Data Mining and Knowledge Discovery for Process Monitoring and Control*. Springer-Verlag, London.
  98. Welch, G. and Bishop, G., 2001. *An introduction to the Kalman filter*. Technical Report, University of North Carolina, Chapel Hill.
  99. Werbos, P. J., 1974. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph.D. Thesis, Harvard University, Cambridge, MA.
  100. Williams, G. P., 1997. *Chaos theory tamed*. Taylor & Francis, London.
  101. Wolf, A., Swift, J. B., Swinney, H. L. and Vastano, J. A., 1985. Determining Lyapunov exponents from a time series. *Physica D*, 16, 285-317.
  102. World Meteorological Organization, 1992. *Simulated Real-time Intercomparison of Hydrological Models*. Operation Hydrology Report, WMO, Geneva.

## Appendix A Back-Propagation Algorithm

Back-propagation algorithm is a computationally efficient method of training artificial neural networks, specifically multilayer perceptrons, to perform a desired task. The development of the back-propagation algorithm put to rest the pessimism about learning in multilayer perceptrons, and hence led to a renaissance in artificial neural networks. It was first developed by Werbos in 1974 (Werbos, 1974), however it wasn't until 1986, through the publication of the seminar book entitled *Parallel Distributed Processing* (Rumelhart and McClelland, 1986), that it gained recognition.

Denote  $d_j(n)$  as the desired response for an output neuron  $j$  at iteration  $n$  (i.e. presentation of the  $n$ th training example), and  $y_j(n)$  as the actual output of the output neuron  $j$  at iteration  $n$ , the instantaneous error energy  $E(n)$  for the network can be defined by

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) = \frac{1}{2} \sum_{j \in C} (d_j(n) - y_j(n))^2, \quad (\text{A.1})$$

where  $e_j(n) = d_j(n) - y_j(n)$  is the error signal at the output neuron  $j$ , and the set  $C$  includes all the neurons in the output layer of the network. The average squared error

energy is obtained by summing  $E(n)$  over all  $n$  and then normalizing with respect to the set size  $N$  (total number of training examples), as shown by

$$E_{av} = \frac{1}{N} \sum_{n=1}^N E(n) = \frac{1}{2N} \sum_{n=1}^N \sum_{j \in C} e_j^2(n). \quad (\text{A.2})$$

For a given training set,  $E_{av}$  represents the cost function as a measure of learning performance. The objective of the learning process is to adjust the synaptic weights of the network to minimize  $E_{av}$ .

During the learning process, one complete presentation of the entire training set is called an epoch. Back-propagation learning may thus proceed in two basic ways for a given training set, i.e. Sequential Mode and Batch Mode. In the sequential mode the synaptic weights are updated after the presentation of each training example, whereas in the batch mode the synaptic weight updating is performed after the presentation of all the training examples that constitute an epoch. Following will consider the sequential mode for the derivation of the back-propagation algorithm.

In the sequential mode, the adjustments to the synaptic weights are made in accordance with the respective errors computed for each pattern presented to the network. The arithmetic average of these individual synaptic weight changes over the training set is therefore an estimate of the true change that would result from modifying the weights based on minimizing the cost function  $E_{av}$  over the entire training set. In this context, a correction  $\Delta w_{ji}(n)$  applied to the synaptic weight  $w_{ji}(n)$  can be defined by the delta rule,

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)}, \quad (\text{A.3})$$

where  $\eta$  is the learning rate parameter, and the minus sign accounts for gradient descent in the synaptic weight space. Reformulate the correction  $\Delta w_{ji}(n)$  according to the chain rule of calculus as

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (\text{A.4})$$

In Equation (A.4),  $v_j(n)$  is the induced local field produced at the input of the activation function  $\varphi_j(\cdot)$  associated with neuron  $j$ , i.e.

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n), \quad (\text{A.5})$$

where  $y_i(n)$  is the input signal of neuron  $j$ , and  $m$  is the total number of inputs applied to neuron  $j$ . Defining the local gradient  $\delta_j(n)$  as

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)}, \quad (\text{A.6})$$

and substituting Equation (A.5) into Equation (A.4) yields

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n). \quad (\text{A.7})$$

Equation (A.7) is the universal equation derived for the back-propagation algorithm. Next step is to find a proper expression for the local gradient  $\delta_j(n)$ . Unlike Equations (A.1) and (A.2), notation  $j$  in Equations (A.3) – (A.7) represents a general neuron in the network, which can be either an output neuron or a hidden neuron. Two distinct cases are therefore identified depending on where in the network neuron  $j$  is located.

- Case 1 Neuron  $j$  Is an Output Node



Consider Figure A.1, which depicts an output neuron  $j$  being fed by a set of signals produced by a layer of neurons to its left. Rewrite Equation (A.6) as

$$\delta_j(n) = -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)}. \quad (\text{A.8})$$

In Equation (A.8),  $y_j(n)$  is the output signal of neuron  $j$  calculated by

$$y_j(n) = \varphi_j(v_j(n)). \quad (\text{A.9})$$

Substituting Equations (A.1) and (A.9) into Equation (A.8), the local gradient  $\delta_j(n)$  for output neuron  $j$  can be finalized as

$$\delta_j(n) = e_j(n) \varphi_j'(v_j(n)). \quad (\text{A.10})$$

The local gradient  $\delta_j(n)$  for output neuron  $j$  is equal to the product of the corresponding error signal  $e_j(n)$  for that neuron and the derivative  $\varphi_j'(v_j(n))$  of the associated activation function.

- Case 2 Neuron  $j$  Is a Hidden Node

Consider Figure A.2, which depicts a hidden neuron  $j$  connected to an output neuron  $k$ . Rewrite Equation (A.6) as

$$\delta_j(n) = -\frac{\partial E(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)}. \quad (\text{A.11})$$

From Figure A.2, the error energy  $E(n)$  can be calculated by

$$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n). \quad (\text{A.12})$$

Substituting Equations (A.12) and (A.9) into Equation (A.10) yields

$$\delta_j(n) = -\sum_{k \in C} e_k(n) \frac{\partial e_k(n)}{\partial y_j(n)} \varphi_j'(v_j(n)). \quad (\text{A.13})$$

Reformulate Equation (A.13) according to the chain rule as

$$\delta_j(n) = -\sum_{k \in C} e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \varphi_j'(v_j(n)). \quad (\text{A.14})$$

From Figure A.2, it can be noticed that

$$e_k(n) = d_k(n) - y_k(n), \quad (\text{A.15})$$

$$y_k(n) = \varphi_k(v_k(n)), \quad (\text{A.16})$$

and

$$v_k(n) = \sum_{j=0}^m w_{kj}(n) y_j(n), \quad (\text{A.17})$$

where  $y_j(n)$  is the input signal of neuron  $k$ , and  $m$  is the total number of inputs applied to neuron  $k$ .

Substituting Equations (A.15) – (A.17) into Equation (A.14), and making use of the definition of the local gradient  $\delta_j(n)$  given in Equation (A.10) with the index  $k$  substituted for  $j$ , i.e.

$$\delta_k(n) = e_k(n) \varphi_k'(v_k(n)), \quad (\text{A.18})$$

the local gradient  $\delta_j(n)$  for hidden neuron  $j$  can be finalized as

$$\delta_j(n) = \sum_{k \in C} \delta_k(n) w_{kj}(n) \varphi_j'(v_j(n)). \quad (\text{A.19})$$

The local gradient  $\delta_j(n)$  for hidden neuron  $j$  is equal to the product of the weighted sum of the  $\delta_k(n)$ s computed for the neurons in the layer to the immediate right of that neuron and the derivative  $\phi_j'(v_j(n))$  of the associated activation function.

The back-propagation algorithm provides an approximation to the trajectory in synaptic weight space computed by the method of steepest descent. Small learning rate parameter  $\eta$  tends to be desirable to make the trajectory smooth. However, this merit is attained at the cost of a slow learning rate. With the intention to speed up the learning rate yet avoid the danger of instability, Rumelhart et al. (1986) modified the delta rule of Equation (A.7) into the generalized delta rule, as shown by

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n), \quad (\text{A.20})$$

where  $\alpha$  is referred to as the momentum constant, restricted to the range  $0 \leq |\alpha| < 1$ , and  $\alpha \Delta w_{ji}(n-1)$  is called the momentum term.

Figure A.3 presents the back-propagation algorithm cycle for the sequential mode. The corresponding steps can be summarized as follows,

- Initialization.

Initialize the synaptic weights in the network. If no prior information is available, synaptic weights are usually assumed to follow the uniform distribution with zero mean and specified variance.

- Presentations of Training Examples.

Present the network with an epoch of training examples. For each training example, perform the sequence of forward and backward computations described as follows.

- Forward Computation.

By proceeding forward through the network layer by layer, compute the induced local fields and function signals of the network

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n), \quad (\text{A.21})$$

and

$$y_j(n) = \varphi_j(v_j(n)). \quad (\text{A.22})$$

For output neuron  $j$ , compute the error signal

$$e_j(n) = d_j(n) - y_j(n), \quad \text{for output neuron } j. \quad (\text{A.23})$$

- Backward Computation.

By passing the error signals backward through the network layer by layer, compute recursively the local gradients  $\delta$ s of the network

$$\delta_j(n) = e_j(n) \varphi_j'(v_j(n)), \quad \text{for output neuron } j, \quad (\text{A.24})$$

and

$$\delta_j(n) = \sum_{k \in C} \delta_k(n) w_{kj}(n) \varphi_j'(v_j(n)), \quad \text{for hidden neuron } j. \quad (\text{A.25})$$

Adjust the synaptic weights of the network in accordance with the generalized delta rule

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n), \quad (\text{A.26})$$

where

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n). \quad (\text{A.27})$$

- Iteration.

Iterate Presentations of Training Examples, Forward Computation and Backward Computation until the stopping criterion is met. Possible stopping criteria include: the synaptic weights stabilize, the generalization performance is adequate, the average squared error energy  $E_{av}$  is less than some critical value, the absolute rate of change in the average squared error energy  $E_{av}$  is sufficiently small, etc.

In the batch mode of back-propagation learning, the average squared error energy  $E_{av}$  is defined as the cost function, i.e.

$$E_{av} = \frac{1}{2N} \sum_{n=1}^N \sum_{j \in C} e_j^2(n). \quad (\text{A.28})$$

The adjustment  $\Delta w_{ji}(n)$  applied to the synaptic weight  $w_{ji}(n)$  can therefore be formulated according to the delta rule

$$\Delta w_{ji}(n) = -\eta \frac{\partial E_{av}}{\partial w_{ji}(n)} = -\frac{\eta}{N} \sum_{n=1}^N e_j(n) \frac{\partial e_j(n)}{\partial w_{ji}(n)}, \quad (\text{A.29})$$

where  $\partial e_j(n)/\partial w_{ji}(n)$  can be calculated in the same way proceeded in the sequential mode. According to Equation (A.29), the adjustment  $\Delta w_{ji}(n)$  is made only after the entire training set has been presented to the network.

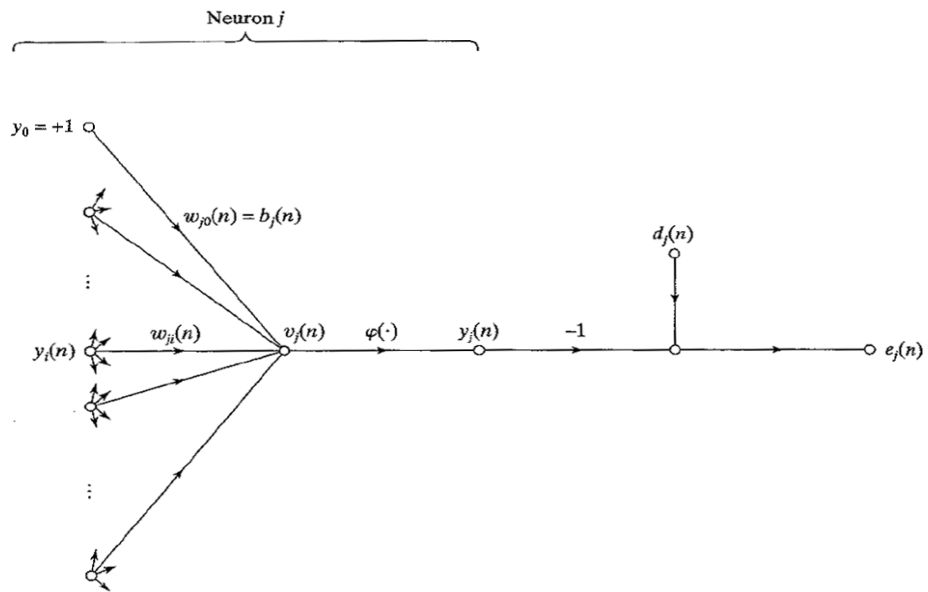


Figure A.1 Signal-flow graph of output neuron  $j$ .

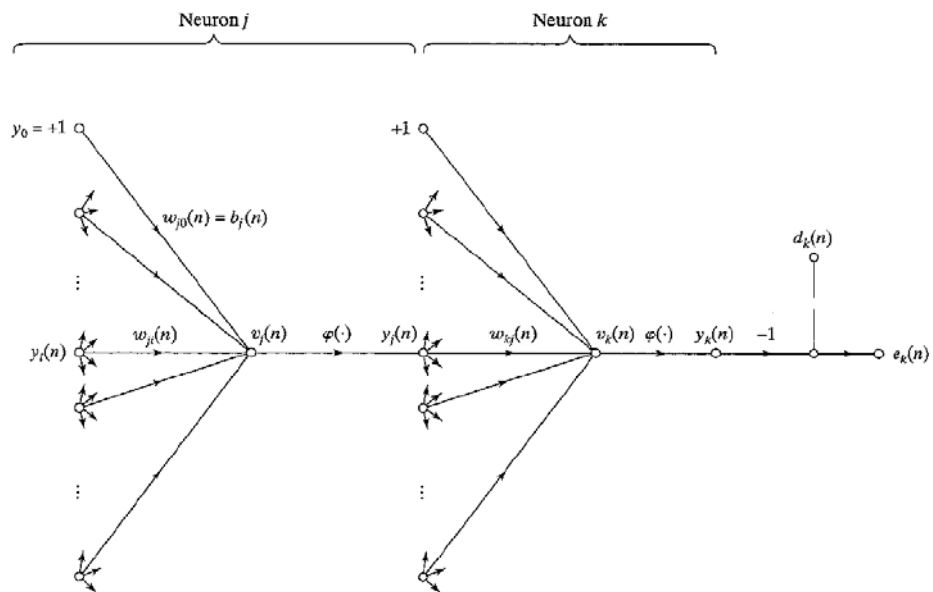


Figure A.2 Signal-flow graph of hidden neuron  $j$  connected to output neuron  $k$ .

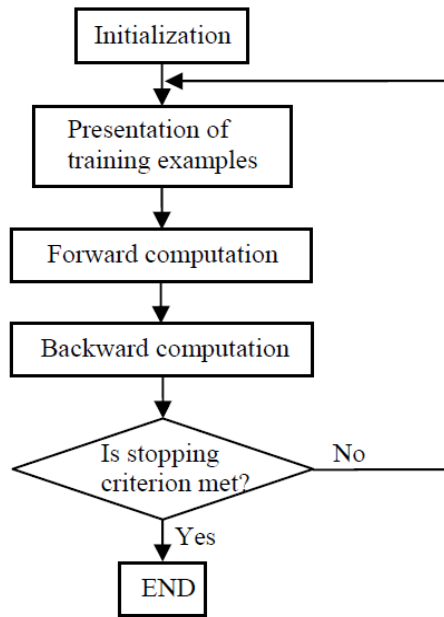


Figure A.3 Back-propagation algorithm cycle.

## Appendix B Linear Kalman Filter Algorithm

Linear dynamic system is controlled by the coupled equations in the state-space form

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_{k-1}, \quad (\text{B.1})$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \quad (\text{B.2})$$

With a linear estimator as the objective, the analysis state estimate  $\mathbf{x}_k^a$  can be expressed as a linear combination of the forecast state estimate  $\mathbf{x}_k^f$  and the measurement  $\mathbf{z}_k$

$$\mathbf{x}_k^a = \mathbf{K}'_k \mathbf{x}_k^f + \mathbf{K}_k \mathbf{z}_k, \quad (\text{B.3})$$

where  $\mathbf{K}'_k$  and  $\mathbf{K}_k$  are the multiplying factors to be determined.

Applying the principle of orthogonality yields (Haykin, 2001)

$$E[\mathbf{e}_k^f \cdot \mathbf{z}_i^T] = \mathbf{0} \text{ for } i = 1, 2, \dots, k, \quad (\text{B.4})$$

$$E[\mathbf{e}_k^a \cdot \mathbf{z}_i^T] = \mathbf{0} \text{ for } i = 1, 2, \dots, k, \quad (\text{B.5})$$

where  $\mathbf{e}_k^f$ ,  $\mathbf{e}_k^a$  are the forecast and analysis errors calculated by

$$\mathbf{e}_k^f = \mathbf{x}_k - \mathbf{x}_k^f, \quad (\text{B.6})$$

$$\mathbf{e}_k^a = \mathbf{x}_k - \mathbf{x}_k^a. \quad (\text{B.7})$$

Using Equations (B.2), (B.3), and (B.7), Equation (B.5) can be rewritten as

$$E\left[\left(\mathbf{x}_k - \mathbf{K}'_k \mathbf{x}_k^f - \mathbf{K}_k \mathbf{H}_k \mathbf{x}_k - \mathbf{K}_k \mathbf{v}_k\right) \cdot \mathbf{z}_i^T\right] = \mathbf{0} \text{ for } i = 1, 2, \dots, k. \quad (\text{B.8})$$



As the measurement noise  $\mathbf{v}$  is assumed to be Gaussian, it follows that

$$E[\mathbf{v}_k \cdot \mathbf{z}_i^T] = \mathbf{0}. \quad (\text{B.9})$$

With Equations (B.4), (B.6) and (B.9), Equation (B.8) transforms into,

$$(\mathbf{I} - \mathbf{K}'_k - \mathbf{K}_k \mathbf{H}_k) E[\mathbf{x}_k \cdot \mathbf{z}_i^T] = \mathbf{0} \text{ for } i = 1, 2, \dots, k. \quad (\text{B.10})$$

For arbitrary values of the state  $\mathbf{x}_k$  and measurement  $\mathbf{z}_i$ , Equation (B.10) can be satisfied only if

$$\mathbf{I} - \mathbf{K}'_k - \mathbf{K}_k \mathbf{H}_k = \mathbf{0}, \quad (\text{B.11})$$

or equivalently define  $\mathbf{K}'_k$  in terms of  $\mathbf{K}_k$  as

$$\mathbf{K}'_k = \mathbf{I} - \mathbf{K}_k \mathbf{H}_k. \quad (\text{B.12})$$

Substituting Equation (B.12) into Equation (B.3), the analysis state estimate  $\mathbf{x}_k^a$  can be formulated as

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^f), \quad (\text{B.13})$$

where matrix  $\mathbf{K}_k$  is call the Kalman gain.

There now remains the problem of deriving an explicit formula for the Kalman gain  $\mathbf{K}_k$ , such that the analysis error covariance

$$\mathbf{P}_k^a = E[\mathbf{e}_k^a \cdot \mathbf{e}_k^{aT}] \quad (\text{B.14})$$

can be minimized.

Substituting Equations (B.2), (B.6), (B.7) and (B.13) into Equation (B.14), the analysis error covariance spreads to

$$\begin{aligned}
 \mathbf{P}_k^a &= E \left\{ \left[ (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{e}_k^f - \mathbf{K}_k \mathbf{v}_k \right] \cdot \left[ (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{e}_k^f - \mathbf{K}_k \mathbf{v}_k \right]^T \right\} \\
 &= E \left\{ (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{e}_k^f \left[ \mathbf{e}_k^{fT} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T - \mathbf{v}_k^T \mathbf{K}_k^T \right] - \mathbf{K}_k \mathbf{v}_k \left[ \mathbf{e}_k^{fT} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T - \mathbf{v}_k^T \mathbf{K}_k^T \right] \right\}
 \end{aligned} \tag{B.15}$$

Model and measurement errors are assumed to be independent, i.e.

$$E \left[ \mathbf{e}_k^f \cdot \mathbf{v}_k^T \right] = E \left[ \mathbf{v}_k \cdot \mathbf{e}_k^{fT} \right] = 0. \tag{B.16}$$

Substituting Equation (B.16) into Equation (B.15), Equation (B.15) can be simplified to

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^f (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T, \tag{B.17}$$

where  $\mathbf{R}_k$ ,  $\mathbf{P}_k^f$  are the measurement error covariance and forecast estimate error covariance defined as

$$\mathbf{R}_k = E \left[ \mathbf{v}_k \cdot \mathbf{v}_k^T \right], \tag{B.18}$$

$$\mathbf{P}_k^f = E \left[ \mathbf{e}_k^f \cdot \mathbf{e}_k^{fT} \right]. \tag{B.19}$$

To minimize the analysis error covariance  $\mathbf{P}_k^a$ , it is equivalent to minimize the scalar sum of its diagonal elements, i.e. the trace of  $\mathbf{P}_k^a$ . To find  $\mathbf{K}_k$  which produces a minimum, the partial derivative of  $tr(\mathbf{P}_k^a)$  with respect to  $\mathbf{K}_k$  is equated to zero

$$\frac{\partial \left[ tr(\mathbf{P}_k^a) \right]}{\partial \mathbf{K}_k} = 0. \tag{B.20}$$

Substituting Equation (B.17) into Equation (B.20), and noticing the following relations in matrix calculus

$$\frac{\partial}{\partial \mathbf{A}} \left[ tr(\mathbf{A} \mathbf{B} \mathbf{A}^T) \right] = 2 \mathbf{A} \mathbf{B} \text{ for matrices } \mathbf{A} \text{ and } \mathbf{B} \text{ where } \mathbf{B} \text{ is symmetric,} \tag{B.21}$$

$$\frac{\partial(\mathbf{AB})}{\partial \mathbf{A}} = \mathbf{B}^T \text{ for matrices } \mathbf{A} \text{ and } \mathbf{B}, \quad (\text{B.22})$$

Equation (B.20) can be transformed into

$$-2(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^f \mathbf{H}_k^T + 2\mathbf{K}_k \mathbf{R}_k = 0. \quad (\text{B.23})$$

Solving Equation (B.23) for  $\mathbf{K}_k$  yields

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^T \left[ \mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{R}_k \right]^{-1}. \quad (\text{B.24})$$

Substituting Equation (B.23) into Equation (B.17), the analysis error covariance can be formulated as

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^f. \quad (\text{B.25})$$

The initial conditions for the linear Kalman filter can be specified as

$$\mathbf{x}_0^a = E[\mathbf{x}_0], \quad (\text{B.26})$$

$$\mathbf{P}_0^a = E\left[ (\mathbf{x}_0 - \mathbf{x}_0^a) (\mathbf{x}_0 - \mathbf{x}_0^a)^T \right]. \quad (\text{B.27})$$

In the forecast step, the forecast state estimate and the forecast error covariance are projected forward through time

$$\mathbf{x}_k^f = \mathbf{A}_k \mathbf{x}_{k-1}^a + \mathbf{B}_k \mathbf{u}_k, \quad (\text{B.28})$$

$$\mathbf{P}_k^f = \mathbf{A}_k \mathbf{P}_{k-1}^a \mathbf{A}_k^T + \mathbf{Q}_k. \quad (\text{B.29})$$

Once  $\mathbf{x}_k^f$  and  $\mathbf{P}_k^f$  are calculated, the analysis state estimate and the analysis error covariance can be updated in the analysis step

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^T \left[ \mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{R}_k \right]^{-1}, \quad (\text{B.30})$$

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^f), \quad (\text{B.31})$$

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^f. \quad (\text{B.32})$$

The process of forecast and analysis is repeated recursively until the desired time step is reached.

## List of Publications

Part of this thesis has been published in or submitted for possible publication to the following international journals or conferences:

### International Journals

- Sun, Y., Sisomphon, P., Babovic, V. and Chan E. S., 2009. Applying local model approach for tidal prediction in a deterministic model. *International Journal for Numerical Methods in Fluids*, 60(6), 651-667.
- Sun, Y., Sisomphon, P., Babovic, V. and Chan E. S., 2009. Efficient data assimilation method based on chaos theory and Kalman filter with an application in Singapore Regional Model. *Journal of Hydro-environment Research*, 3(2), 85-95.
- Sun, Y., Babovic, V. and Chan E. S., 2010. Multi-step-ahead model error prediction using time-delay neural networks combined with chaos theory. *Journal of Hydrology*, in press.
- Sun, Y. B., Babovic, V. and Chan E. S., 2010. Neural networks as routine for error correction with an application in Singapore Regional Model. *Continental Shelf Research*, submitted for possible publication.

- Sun, Y. B., Babovic, V. and Chan E. S., 2010. Prediction of sea level anomalies using local model approach in chaos theory. *Journal of Hydroinformatics*, submitted for possible publication.

**International Conferences**

- Sun, Y., Sisomphon, P., Babovic, V. and Chan E. S., 2008. Enhancing tidal prediction accuracy in Singapore Regional Model using local model approach (Abstract). *Proceedings of the 5th Asia Oceania Geosciences Society Annual Meeting*, Busan.
- Sun, Y., Sisomphon, P., Babovic, V. and Chan E. S., 2008. Enhancing tidal prediction accuracy in Singapore Regional Model using local model approach. *Proceedings of the 7th WSEAS International Conference on Non-linear Analysis, Non-linear Systems and Chaos*, Corfu, 165-170.
- Sun, Y., Sisomphon, P., Babovic, V. and Chan E. S., 2009. Comparison of Kalman filter and inter-model correlation method for data assimilation in tidal prediction. *Proceedings of the 8th International Conference on Hydroinformatics*, Concepción.
- Sun, Y., Babovic, V. and Chan E. S., 2009. Neural networks as routine for error correction in Singapore Regional Model (Abstract). *Proceedings of the 6th Asia Oceania Geosciences Society Annual Meeting*, Singapore.
- Sun, Y., Babovic, V., Chan E. S. and Sisomphon, P., 2010. Model error prediction using neural networks combined with chaos theory. *Proceedings of the 9th International Conference on Hydroinformatics*, Tianjin.

- Sun, Y., Zemskyy, P., Ooi, S. K., Sisomphon, P. and Gerritsen, H., 2011. Study on the correlations between current anomaly and sea level anomaly gradients in Singapore and Malacca Straits. *Proceedings of the 4th ASCE-EWRI International Perspective on Water Resources & the Environment*, Singapore.