POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Real-time Anomaly Detection in the Behaviour of Structures

LUONG HA NGUYEN
Département des génies civil, géologique et des mines

Thèse présentée en vue de l'obtention du diplôme de *Philosophiae Doctor*
Génie civil

September 2019

© Luong Ha Nguyen, 2019.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Real-time Anomaly Detection in the Behaviour of Structures

présentée par **Luong Ha NGUYEN**
en vue de l'obtention du diplôme de *Philosophiae Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Pierre LÉGER, président
James-A. GOULET, membre et directeur de recherche
Martin TRÉPANIER, membre
Lijun SUN, membre externe

ACKNOWLEDGEMENTS

First, I would like to express my gratitude and appreciation to my advisor, James-A. Goulet, for being a constant source of knowledge, guidance, positive energy, and encouragement. During my time as a student under his supervision, James has invested a great amount of time and effort in my research as well as in my personal development. I could not have asked for a better mentor.

I would like to thank my defense committee members Pierre Léger, Lijun Sun, and Martin Trépanier for their time and support in evaluating my thesis. I would like to further thank Pierre Léger for generously sharing his expertise in the field of dam engineering with me.

I would like to thank Natural Sciences and Engineering Research Council of Canada, Hydro Québec, Hydro Québec's Research Institute, the Institute For Data Valorization for the financial support of this research. I would like to further thank Hydro Québec for providing the datasets. Many thanks to Benjamin Miquel, Vincent Roy, and Patrice Côté from Hydro Québec for their suggestions, help, and encouragement in the research project.

I would like to thank my friends and colleagues at Polytechnique Montréal with whom I have had the pleasure of working with over the years. These include Saeid Amiri, Simon Brousseau, Bhargob Deka, Pierre Escoffres, Ianis Gaudot, Zachary Hamida, Shervin Khazaeli, Catherine Paquin, and Alyssa Quach.

Finally, I would like to thank my friends and family for their unconditional support over the years. To my friends, Cecilia, Chris, Duc, Hoang, Kévin, Lucie, Marie, Nelson, Sayouba, Tram, and Tuan, thank you for sticking with me for so many years. To Chris and Lucie, thank you for providing a welcome distraction from university. To my brother, Hai, thank you for always being there with me during difficult times. To my parents for being my biggest supporters, and for always encouraging me to pursue my passions. Your love and support mean the world to me.

RÉSUMÉ

Les structures telles que les ponts, les barrages, les bâtiments et les tunnels sont des composantes majeures du réseau d'infrastructures qui contribuent à la croissance économique d'un pays. La détérioration des infrastructures est associée à un impact négatif sur l'économie causé par les coûts directs de maintenance, ainsi que par les coûts indirects liés à la production de biens et de services. L'adoption de stratégies de maintenance préventive et prédictive est une solution à long terme pour atténuer les effets de la détérioration, prolonger la durée de vie en service et minimiser le coût du cycle de vie des infrastructures vieillissantes. Toutes les structures se détériorent et se dégradent au cours de leur vie. Ainsi, la détection d'anomalies dans le taux de changement de dégradation permet de déclencher des actions d'entretien préventif et d'interventions. Cependant, il n'existe actuellement aucune méthodologie générique d'interprétation des données qui est capable de détecter des anomalies en temps réel sans être affectées par de fausses alarmes.

Le projet de recherche a pour objectif de développer des méthodes basées sur des données pour la surveillance de l'état des structures ou *Structural Health Monitoring* (SHM) en anglais, qui sont capables de suivre le comportement d'une structure en temps réel. Afin d'atteindre cet objectif principal, cette thèse propose des méthodes d'apprentissage automatique qui permettent (i) d'isoler le comportement d'une structure à partir de ses données brutes, y compris les effets externes périodiques causés par les conditions environnementales ainsi que les erreurs d'observation, (ii) d'apprendre automatiquement des paramètres du modèle et (iii) de détecter les anomalies en temps réel sans supervision humaine et sans avoir besoin des données étiquetées reflétant les conditions normales et anormales.

Les méthodes proposées sont validées avec les données enregistrées sur des structures réelles. Les résultats illustrent que ces méthodes réussissent à détecter en temps réel les changements dans le comportement de données de déplacement enregistrées sur un barrage, ainsi qu'à modéliser des signaux périodiques, non harmoniques et complexes tels que les données de charge de trafic enregistrées sur un pont. De plus, les méthodes d'apprentissage sont capables d'identifier automatiquement les paramètres optimaux du modèle ainsi que d'estimer, en ligne et hors ligne, la fonction de densité de probabilité à postériori de paramètres du modèle. En outre, les méthodes proposées sont facilement transférables d'une structure à une autre et d'un type de mesure à un autre. En résumé, les méthodes proposées offrent une voie potentielle associée au déploiement à grande échelle de systèmes SHM permettant de surveiller en temps réel l'état d'une population de structures.

ABSTRACT

Civil structures such as bridges, dams, buildings, and tunnels are major components of infrastructure networks which establish a foundation for the economic growth of a country. Infrastructure deterioration is associated with a negative economic impact caused by the direct maintenance costs, as well as the indirect costs for production and transport of goods and services. One long-term solution to mitigate the effects of deterioration, to prolong the service life, and to minimize life-cycle cost of ageing infrastructure is to adopt preventive and predictive maintenance strategies. All structures experience deterioration and degradation during their lifetime and thus, early detection of anomalies in the rate of change of degradation could be used as a trigger for preventive maintenance and interventions. Yet, there is currently no generic data-interpretation methodology capable of detecting anomalies in real time without also being adversely affected by false alarms.

The goal of the research project is to develop data-driven methods for *Structural Health Monitoring* (SHM) that are capable of tracking a structure's behaviour in real time. To achieve this main goal, this thesis proposes machine learning methods that allow (i) isolating the structural behaviour from raw data including the periodic external effects caused by environmental conditions as well as observation errors, (ii) automatically learning unknown model parameters from data, and (iii) detecting anomalies in real time without human supervision and without requiring labeled training data.

The proposed methodologies are validated with data recorded on real structures. The results show that these methodologies succeed in detecting the changes in the behaviour of the displacement data collected on a dam in real time, as well as in modeling complex non-harmonic periodic patterns such as traffic load data recorded on a bridge. Furthermore, the learning methods are able to automatically find the optimal model parameters as well as to approximate, online and offline, the posterior probability density function of model parameters. Moreover, the proposed methodologies are easily transferable from one structure to another and from one measurement type to another. Putting this all together, the proposed methodologies offer a promising path toward the large-scale deployment of SHM systems for monitoring health and conditions of a population of structures in real time.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

RÉSUMÉ iv

ABSTRACT v

TABLE OF CONTENTS vi

LIST OF TABLES ix

LIST OF FIGURES x

LIST OF SYMBOLS AND ACRONYMS xv

LIST OF APPENDICES xx

CHAPTER 1 INTRODUCTION 1
- 1.1 Motivation 1
- 1.2 Anomaly Detection 2
- 1.3 Research Objectives 3
- 1.4 Thesis Outline 4
- 1.5 Co-Authored Papers 4

CHAPTER 2 LITERATURE REVIEW 5
- 2.1 Structural Health Monitoring 5
- 2.2 Regression Models 6
  - 2.2.1 Linear Regression 6
  - 2.2.2 Neural Networks 7
  - 2.2.3 Common Limitations 8
- 2.3 State-Space Models 10
  - 2.3.1 Autoregressive Models 11
  - 2.3.2 Kalman/Particle Filter-Based Models 12
  - 2.3.3 Common Limitations 12
  - 2.3.4 Bayesian Dynamic Linear Models 13
- 2.4 Conclusion 13

CHAPTER 3 BAYESIAN DYNAMIC LINEAR MODELS 16
3.1 Introduction 16
3.2 Theory of Bayesian Dynamic Linear Models 16
3.2.1 Kalman Filter/Smoother 17
3.2.2 Modeling External Effects 19
3.3 Theory of Switching Kalman Filter 20
3.3.1 SKF-filter Step 21
3.3.2 SKF-collapse Step 22
3.4 Generic Model Architecture for Anomaly Detection 25
3.5 Hidden Dynamic Regression 27
3.6 Kernel Regression 28
3.6.1 Periodic Kernel Regression 29
3.6.2 Periodic Kernel Regression with BDLMs 30
3.7 Applications 32
3.7.1 Comparison of Approaches for Modeling a Simple Non-Harmonic Pattern 34
3.7.2 Modeling a Complex Non-Harmonic Periodic Pattern 38
3.7.3 Offline Anomaly Detection 42
3.8 Conclusion 47

CHAPTER 4 BATCH LEARNING 48
4.1 Introduction 48
4.2 Likelihood 49
4.3 Maximum Likelihood Estimation 49
4.3.1 Batch Gradient Ascent 50
4.3.2 Stochastic Gradient Ascent 50
4.3.3 Common Challenges 51
4.3.4 Optimization Algorithm 53
4.3.5 Practical Implementation for BDLMs 56
4.4 Bayesian Estimation 64
4.4.1 Hamiltonian Monte Carlo (HMC) 64
4.4.2 Laplace Approximation (LAP) 66
4.4.3 Gaussian Mixture Approximation 67
4.4.4 Framework Architecture for BDLMs 68
4.5 Transformation of Model Parameter 70
4.6 Applications 72
4.6.1 Comparison of Optimization Algorithms 73

4.6.2 Comparison of LAP-P with HMC-P on a Simulated Dataset 78
4.6.3 Comparison of LAP-P with HMC-P on a Real Dataset 86
4.7 Conclusion 91

CHAPTER 5 ONLINE LEARNING 93
5.1 Introduction 93
5.2 Rao-Blackwellized Particle Filter 94
5.3 Framework Architecture 96
5.4 Applications 99
5.4.1 Horizontal Displacement of a Dam 99
5.4.2 Horizontal Displacement of a Dam with Artificial Anomalies 103
5.4.3 Three-Dimensional Displacement of a Dam 108
5.5 Conclusion 118

CHAPTER 6 CONCLUSION 119
6.1 Summary of Research Findings 119
6.2 Limitations 120
6.2.1 Criterion for Detecting Anomalies 120
6.2.2 Non-Structural Anomaly 121
6.2.3 Initializing Model Parameters and Hyperparameter Tuning 121
6.3 Future Research 121
6.3.1 Topology Learning 122
6.3.2 Anomaly Detection and Information Redundancy 122
6.3.3 Non-Periodic Pattern Modeling 122

REFERENCES 125

APPENDICES 137

LIST OF TABLES

Table 3.1 Comparison of three approaches of the forecast capacity for modeling an external effect. Model #1: kernel regression; Model #2: hidden dynamic regression; Model #3: Fourier form; RMSE: root-mean-square error; MAE: mean absolute error; LPD: Log Predictive Density. . . . 38

Table 3.2 Evaluation of forecast accuracy with respect to the different forecast periods for the traffic load data. RMSE: root-mean-square error; MAE: mean absolute error; LPD: log predictive density. 42

Table 4.1 Summary of the initial configurations for the model parameters. . . . 76

Table 4.2 Optimal vectors of model parameters for each optimization algorithm. BGA: batch gradient ascent; MGA: mini-batch gradient ascent; NR: Newton-Raphson; MMT: momentum; RMSProp: root mean square propagation; ADAM: adaptive moment estimation; AMMT: adaptive momentum; CPU: central processing unit. 77

Table 4.3 The amount of data points with respect to the training set length (TSL) for the simulated dataset. 80

Table 4.4 The amount of data points with respect to the training set length (TSL) for the real dataset. 88

LIST OF FIGURES

Figure 2.1 The left figure presents the errors with respect to time between observations and predicted values for the training set. The right figure shows the histogram and underlying probability density function for errors observed during the training period. 9

Figure 2.2 Illustration of the hypothesis-testing procedure for anomaly detection using a training set (Tr) and three test sets T1, T2, and T3 where anomalies only occur for T2 and T3. 9

Figure 2.3 Illustration of structural responses (a) that are decomposed in a set of hidden state variables (b): baseline component ($\mathbf{x}^{\mathtt{B}}$), periodic component ($\mathbf{x}^{\mathtt{PD}}$), and autoregressive component ($\mathbf{x}^{\mathtt{AR}}$). The mean values of the hidden state variables, $\mu_{t|t}$, and its bound uncertainty, $\mu_{t|t} \pm \sigma_{t|t}$ are presented by the solid line and shaded region. 14

Figure 3.1 Examples of realizations of $\mathbf{x}^{\mathtt{F}}$ for a Fourier form component with parameters $\sigma_w^{\mathtt{F}} = 0.01$ and $\mathtt{P} = 10$. 19

Figure 3.2 Examples of (a) harmonic and (b) non-harmonic periodic pattern. . . 20

Figure 3.3 Illustration of (a) how the number of state sequences grows exponentially with time and (b) the Generalized Pseudo Bayesian algorithm of order 2. 21

Figure 3.4 Illustration of the SKF algorithm for two states each having its own transition model. $^{(\cdot)}$ indicates the filtering model being used for computation. 24

Figure 3.5 Illustration of the anomaly detection for the baseline behaviour. (1) and (2) represents the normal and abnormal states of the baseline behaviour. 26

Figure 3.6 Example spline fitted using master and slave control points defined over three sub-segments that are separated by the vertical symmetry lines. 28

Figure 3.7 Examples of periodic kernels. 29

Figure 3.8 Examples of application of kernel regression. 30

Figure 3.9 An example of using Kernel regression with BDLMs. (a) Periodic pattern with a period of 365 days; (b) Evolution of 10 hidden state variables associated with the control point's values for 4 time steps; (c) Left orthographic projection of 10 hidden state variables associated with the control point's values; (d) Estimated values of kernel pattern for 4 time steps; (e) Kernel pattern. 33

Figure 3.10 Location plan of the sensor employed to monitor the dam behaviour. 34

Figure 3.11 Illustration of the raw displacement dataset. 35

Figure 3.12 Illustration of displacement forecasts in the test set. (a) Model #1: Kernel regression; (b) Model #2: hidden dynamic regression; (c) Model #3: Fourier form. 37

Figure 3.13 Traffic load on the Tamar Bridge in the United Kingdom. 39

Figure 3.14 Illustration of the estimation of hidden state variables for the traffic load data: (a) Baseline component, $x_t^{\mathtt{B}}$; (b) Kernel pattern, $x_{t,0}^{\mathtt{KR}}$; (c) Autoregressive component, $x_t^{\mathtt{AR}}$; (d) Traffic load. 41

Figure 3.15 X-direction displacement data collected over the period of 13 years and 1 month. 42

Figure 3.16 Time-step size is presented in a log scale. 43

Figure 3.17 Probabilities of the two states are evaluated using SKF algorithm for the entire dataset. 45

Figure 3.18 Expected values $\mu_{t|t}$ and uncertainty bound $\mu_{t|t} \pm \sigma_{t|t}$ for hidden state variables of a combination of the normal and abnormal models are evaluated using SKF algorithm. 46

Figure 4.1 The log-likelihood of (a) the batch gradient ascent and (b) stochastic gradient ascent algorithm during training. 51

Figure 4.2 Illustration of the impact of the learning rate on the model parameter optimization. (a) a small learning rate; (b) a large learning rate. Each circle presents an update step. The ordering update is presented by the arrows. 52

Figure 4.3 Examples of (a) the local maxima and (b) the saddle points. 52

Figure 4.4 Illustration of how the momentum algorithm addresses the limitations of classic gradient ascent variants. Each arrow presents an update step made by the optimization algorithm. 54

Figure 4.5 Illustration of two metrics employed in BDLMs to evaluate the performance of the optimization algorithm. (a) training log-likelihood value (L_{tr}); (b) validation log-likelihood value (L_v). 57

Figure 4.6 Illustration of the log-likelihood function of the training set and validation set. 58

Figure 4.7 Illustration of the selection of the mini-batch of data for the mini-batch gradient ascent algorithm. g_p: the first derivative of $\mathcal{F}(\boldsymbol{\theta}, \mathbf{y}_{1:\mathrm{T}})$ with respect to $\boldsymbol{\theta}(p)$; h_p: the second derivative of $\mathcal{F}(\boldsymbol{\theta}, \mathbf{y}_{1:\mathrm{T}})$ with respect to $\boldsymbol{\theta}(p)$; L_v: the validation log-likelihood. 61

Figure 4.8 Illustration of the batch optimization procedure using the multi-pass. 63

Figure 4.9 Illustration of the Gaussian mixture approximation for the hidden state variables. 68
Figure 4.10 Illustration of the general procedure for approximating the posterior density of parameters $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathtt{T}})$ and the mean values of the hidden state variables and its covariance matrix at each time t, that is, $\{\hat{\boldsymbol{\mu}}_{t|t}, \hat{\boldsymbol{\Sigma}}_{t|t}\}$ using the combination of the Laplace approximation and Gaussian Mixture Approximation (GMAP). BGA: batch gradient ascent; MGA: mini-batch gradient ascent. 69
Figure 4.11 The two main steps for approximating the posterior density of parameters $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathtt{T}})$ and the mean values of hidden state variables and its covariance matrix at each time t, that is, $\{\hat{\boldsymbol{\theta}}_{t|t}, \hat{\boldsymbol{\Sigma}}_{t|t}\}$ using the combination of the HMC method with Gaussian Mixture Approximation (GMAP). BGA: batch gradient ascent; MGA: mini-batch gradient ascent. 71
Figure 4.12 Illustration of transformation function of model parameters. (a) logistic sigmoid function with $a = 0$ and $b = 1$; (b) base-10 logarithm function; (c) nature logarithm function. `TR` stands for transformation. 72
Figure 4.13 Illustration of the simulated datasets. (a) displacement; (b) temperature. 74
Figure 4.14 Illustration of the average validation-log-likelihood over three runs with respect to the optimization algorithm. BGA: batch gradient ascent; MGA: mini-batch gradient ascent; NR: Newton-Raphson; ADAM: adaptive moment estimation; RMSProp: root mean square propagation; MMT: momentum; AMMT: adaptive momentum. 78
Figure 4.15 Illustration of 4 years of simulated data. 79
Figure 4.16 Illustration of prior distribution choices for model parameters in the original space. 81
Figure 4.17 Each column illustrates the kernel density estimate of the posterior PDFs for each model parameter $p(\theta_i|\mathbf{y}_{1:\mathtt{T}})$ in the original space with respect to the training-set length of simulated data. 82
Figure 4.18 Illustration of the kernel density estimate of the posterior PDFs for each model parameter in the original space with the training set of 1095 days. 83
Figure 4.19 Expected value $\hat{\mu}$ and standard deviation $\hat{\sigma}$ for baseline (left) and autoregressive (right) components using the Laplace approximation procedure (LAP-P) and the Hamiltonian Monte Carlo procedure (HMC-P) with respect to the training-set length of the simulated data. . . . 85

Figure 4.20 Computation time of LAP-P and HMC-P for approximating the parameter's posterior PDF in the simulated dataset. LAP: Laplace Approximation; HMC: Hamiltonian Monte Carlo. 86
Figure 4.21 Raw displacement data. 87
Figure 4.22 Time step size 87
Figure 4.23 Illustration of the kernel density estimate of the posterior PDFs for each model parameter $p(\theta_i|\mathbf{y}_{1:\mathrm{T}})$ in the original space with respect to the training-set length. The data are collected on a dam in Canada. HMC: Hamiltonian Monte Carlo; LAP: Laplace Approximation. . . . 89
Figure 4.24 Expected value $\hat{\mu}$ and standard deviation $\hat{\sigma}$ for baseline (left) and autoregressive (right) components using Laplace approximation procedure (LAP-P) and Hamiltonian Monte Carlo procedure (HMC-P) with respect to the training-set length. The data are collected on a dam in Canada. 90
Figure 4.25 Computation time of LAP-P and HMC-P for approximating the parameter's posterior PDF on the real dataset. LAP: Laplace Approximation; HMC: Hamiltonian Monte Carlo. 91
Figure 5.1 Illustration of the general framework of the online learning for the the Bayesian Dynamic Linear Models. RBPF: Rao-Blackwellized Particle Filter; MCMC: Markov Chain Monte Carlo; LAP: Laplace Approximation; GMAP: Gaussian Mixture Approximation. 97
Figure 5.2 Probability of abnormal state for the displacement data. 100
Figure 5.3 Hidden state estimates. 101
Figure 5.4 Illustration of the estimation of model parameters using Rao-Blackwellized particle filter. 102
Figure 5.5 Kernel density estimate of the posterior PDFs for each model parameter taken at 3 am on June 30, 2014. (a) Autoregression coefficient; (b) Kernel lengthscale; (c) Autoregression standard deviation; (d) Local acceleration standard deviation; (e) Transition probability of the normal state; (f) Transition probability of the abnormal state. 103
Figure 5.6 The superposition of the original and artificial-anomaly dataset. . . . 104
Figure 5.7 Probability of abnormal state for the displacement data. 104
Figure 5.8 Illustration of the estimation of the hidden state variables. 106
Figure 5.9 Estimation of model parameters using Rao-Blackwellized particle filter. 107
Figure 5.10 Illustration of the raw displacement data in three directions. 108
Figure 5.11 Superposition of three time-step lengths. 109

Figure 5.12 Probability of the abnormal state for the displacement data in three directions. 111
Figure 5.13 Estimation of the hidden state variables for X-direction displacement. 112
Figure 5.14 Estimation of the hidden state variables for Y-direction displacement. 113
Figure 5.15 Estimation of the hidden state variables for Z-direction displacement. 114
Figure 5.16 Estimation of model parameters for the X-direction using Rao-Blackwellized particle filter. 115
Figure 5.17 Estimation of model parameters for the Y-direction using Rao-Blackwellized particle filter. 116
Figure 5.18 Estimation of model parameters for the Z-direction using Rao-Blackwellized particle filter. 117
Figure 6.1 Illustration of multiple-datasets analysis (MDA) framework in the context of Structural Health Monitoring. 123
Figure 6.2 Illustration of non-periodic patterns. (a) Flow-rate data; (b) Pressure data. 123

LIST OF SYMBOLS AND ACRONYMS

Symbols

$\mathbf{A}$	Transition matrix
$\mathbf{a}$	Matrix of model parameters
a	An action
`AR`	Autoregressive component
`B`	Baseline component
$\mathbf{b}$	Matrix of model parameters
$\mathbf{C}$	Observation matrix
$\mathcal{C}$	Set of master control point
$\mathbf{c}$	Converged vector
cov	Covariance operator
$\mathbf{D}$	Covariance matrix of the artificial noise of model parameters
$\mathcal{D}$	Set of observations
`D`	Total number of observations
`DR`	Dynamic regression
$\mathbf{d}$	Vector of distance measures
diag	Create diagonal matrix or get diagonal elements of matrix
$\mathbb{E}$	Expected operator
$\mathbf{e}$	Vector of error
`F`	Fourier form
$\mathcal{F}$	Objective function
$\mathbf{f}$	Transition model
$\mathbf{g}(\mathbf{x})$	Function of $\mathbf{x}$
g_p	First derivative of model parameter θ_p
$H(\boldsymbol{\theta}, \mathbf{r})$	Hamiltonian function
$\mathcal{H}$	Hessian matrix
$\mathbf{h}$	Observation model
$\bar{\mathbf{h}}$	Vector of the exponential average of the previous diagonal Hessian terms
h_i	Normalized hidden covariate value
h_p	Second derivative of model parameter θ_p
$h(\mathcal{C}, t)$	Hidden response function
`HDR`	Hidden dynamic regression component

$\mathbf{I}$	Identity matrix
$\mathbf{J}$	Backward Kalman gain matrix
$\mathbf{K}$	Kalman gain matrix
$\mathcal{K}$	Set of the indices of particles
$\mathtt{K}$	Total number of particles/samples
$\tilde{\mathbf{k}}$	Normalized kernel
$\mathtt{KR}$	Kernel regression component
L	Log-likelihood
$\mathcal{L}$	Marginal likelihood
L_{tr}	Log-likelihood of training set
L_v	Log-likelihood of validation set
$\mathtt{L}_s$	Total number of steps for the leapfrog method
$\mathtt{LA}$	Local acceleration
$\mathtt{LT}$	Local trend
$l_{\mathtt{MB}}$	Length of the mini-batch
ℓ	Kernel length
$\mathbf{M}$	Joint probability matrix
$\mathcal{M}$	Mass matrix
$\mathbf{M}_{\boldsymbol{\theta}}$	Matrix of all possible updates of model parameters
$\mathcal{N}$	Gaussian distribution
$\mathtt{N}_{\mathtt{maxM}}$	Maximal number of mini-batch
$\mathtt{N}_{\mathtt{maxEpoch}}$	Maximal number of epoch
$\mathbf{O}$	Innovation covariance matrix
$\mathbf{o}$	Inovation vector
$\mathtt{P}$	Period
p	Probability density function or index of model parameters
$\mathtt{p}$	Total number of unknown model parameters
$\mathtt{PD}$	Periodic component
$\mathbf{Q}$	Model error covariance matrix
$\mathbf{R}$	Observation error covariance matrix
$\hat{\mathbf{R}}$	Estimated Potential Scale Reduction
$\mathbf{r}$	Momentum vector
$\mathtt{S}$	Total number of states
$\mathbf{s}$	Vector of the exponential average of the previous gradients
$\hat{\mathbf{s}}$	Bias correction of the exponential average of the previous gradients
s	Markov-switching variable

$T(\boldsymbol{\theta}, \mathbf{r})$	Kinetic energy
$\mathtt{T}$	Total number of time stamps
t	Time stamp
tol	Convergence tolerance
$\mathbf{u}$	Artificial noise of model parameters
$V(\boldsymbol{\theta}, \mathbf{r})$	Potential energy
$\mathbf{v}$	Observation error
$\mathbf{W}$	State switching probability
$\mathbf{w}$	Model transition error or importance weight
$\mathbf{x}$	Vector of covariates/hidden state variables
$\mathbf{y}$	Vector of observations
$\hat{\mathbf{y}}$	Predicted observations vector
$\mathbf{Z}$	Transition probability matrix
α	Time-scaling factor
β	Momentum contribution factor
β_1	Length scale of the moving average of the past squared gradients
β_2	Length scale of the moving average of the past gradients
$\beta^{\mathtt{R}}$	Regression coefficient
δ_θ	Gradient step size of the model parameter θ
$\boldsymbol{\epsilon}$	Vector of small constants
η	Learning rate
γ^s	Scaling factor of stationary model parameters
γ^d	Scaling factor of non-stationary model parameters
κ	Kernel regression coefficient
μ	Expected value
$\boldsymbol{\mu}$	Vector of expected values
ω	Angular frequency
$\phi^{\mathtt{AR}}$	Autoregression coefficient
π	State probability
$\boldsymbol{\nu}$	Vector of the exponential average of the past squared gradients
$\hat{\boldsymbol{\nu}}$	Bias correction of the exponential average of the past squared gradients
σ	Standard deviation
$\boldsymbol{\theta}$	Vector of model parameters
ξ	Step size for the leapfrog method
ζ	Probability threshold
Δt	Time-step length

$\Delta_{\boldsymbol{\theta}}$	Update step of model parameters
$\psi(\mathbf{x})$	Function taking covariates $\mathbf{x}$ as input
Σ	Covariance matrix
∇	First derivative operator
∇^2	Second derivative operator
$\odot$	Element-wise operator
∞	Infinity

Acronyms

ADAM	Adaptive Moment Estimation
AMMT	Adaptive Momentum
ARM	Autoregressive Model
ARMA	Autoregressive Moving Average Model
ARX	Autoregressive Exogenous
BDLM	Bayesian Dynamic Linear Models
BE	Bayesian Estimation
BGA	Batch Gradient Ascent
CI	Confidence Interval
CP	Control Point
CPU	Central Processing Unit
DOF	Degree of Freedoms
EKF	Extended Kalman Filter
EPSR	Estimated Potential Scale Reduction
GDP	Gross Domestic Product
GMAP	Gaussian Mixture Approximation
GPB	Generalized Pseudo Bayesian
GPU	Graphics Processing Unit
HMC	Hamiltonian Monte Carlo
HMC-P	Hamiltonian Monte Carlo Procedure
HST	Hydrostatic, Seasonal, Time
HTT	Hydrostatic, Temperature, Time
KF	Kalman Filter
KS	Kalman Smoother
LAP	Laplace Approximation
LAP-P	Laplace Approximation Procedure

LPD	Log Predictive Density
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MCMC	Markov Chain Monte Carlo
MDA	Multiple-Datasets Analysis
MGA	Mini-batch Gradient Ascent
MLE	Maximum Likelihood Estimation
MMT	Momentum
NHCV	Normalized Hidden Covariate Value
NN	Neural Network
NR	Newton-Raphson
OLF	Online Learning Framework
PDA	Posterior Density Approximation
PDF	Probability Density Function
PF	Particle Filter
RMSE	Root Mean Square Error
RMSprop	Root Mean Square Propagation
RBPF	Rao-Blackwellized Particle Filter
SGA	Stochastic Gradient Ascent
SHM	Structural Health Monitoring
SIS	Importance Sampling
SKF	Switching Kalman Filter
SSM	State-Space Model
SKS	Switching Kalman Smoother
TLS	Training Set Length
UM	Uncertainty Marginalization
UKF	Unscented Kalman Filter

LIST OF APPENDICES

Appendix A MODEL MATRICES 137
Appendix B OPTIMIZATION ALGORITHMS FOR BDLMS 142
Appendix C MEASURE OF FORECAST ACCURACY 147

CHAPTER 1 INTRODUCTION

1.1 Motivation

Civil structures such as bridges, dams, buildings, and tunnels are major components of infrastructure networks which establish a foundation for the economic growth of a country [1–3]. Infrastructure deterioration is associated with a negative economic impact caused by the direct maintenance costs, as well as the indirect costs for production and transport of goods and services [2]. The health and conditions of structures keep deteriorating as a result of ageing, usage, and environmental changes. According to the Canadian's infrastructure report card published in 2016 [4], approximately 35% of assets are in need of attention. In the USA, the most recent report card [5] claims that American's infrastructure is in poor conditions. According to this report, an estimated investment of US $4.59 trillions is required by 2025 to improve the USA's infrastructure.

In several developed countries, the investment rate in infrastructure has been declining for years [4–6]. In Canada, an independent report published in 2011 by Risk Analytica [7] concluded that to achieve a maximum *Gross Domestic Product* (GDP) growth over the course of 50 years, the Canadian government needs to annually invest in infrastructure with a total amount of 5.1% of GDP where 22% of this investment should be for repairs and maintenance. The repair and maintenance here are defined as the work required to retrain a structure at its designed performance level. Over the last decade, the Canadian government has spent an average of 3% of GDP as annual investment on infrastructure, and approximately 17% of this total investment were spent on repairs and maintenance [8]. It remains significantly below the aforementioned target of 5.1% of GDP. With this investment gap, a number of ageing structures remains in service without optimal maintenance, which is likely to reduce the length of their service life [4].

One long-term solution to mitigate the effects of deterioration, to prolong the service life, and to minimize life-cycle cost of ageing infrastructure is to adopt preventive and predictive maintenance strategies. A key aspect for achieving this goal is to have generic and low-cost methodologies capable of tracking the health and conditions of any structures in real time. As early as in the 1970s, many structures have been monitored to improve the understanding of their behaviour [9, 10]. This research field is known as *Structural Health Monitoring* (SHM). A typical SHM system consists of three main components: a sensor network, a data processing system, and a data-interpretation system [11]. A sensor network contains multiple sensors to monitor different structural responses. A data processing system provides

the means for data acquisition, transmission, and storage. A data-interpretation system includes methodologies that allow tracking and diagnosing the health and conditions of a structure. Quantities monitored on a structure are typically displacement, strains, inclination and accelerations [12, 13]. Sensing technologies have evolved over the last decades [11, 14] and are now cheap and widely available. Despite the notable hardware developments, innovations in data-interpretation methodologies have been evolving at a slower pace. There is currently no data-interpretation methodology capable of detecting anomalies in real time without also being adversely affected by false alarms. An anomaly is defined as an unexpected change in the behaviour of a structure. A false alarm is a false detection of an anomaly that would require the attention of the infrastructure manager, and that consequently incurs costs. Another key aspect is that data-interpretation methods must be easily transferable from one structure to another and from one measurement type to another in order to be financially viable for practical applications as well as to be able to allow for massively distributed SHM systems across thousands of structures.

This research focuses on anomaly detection for civil structures. As aforementioned, all structures will experience deterioration and degradation during their lifetime and thus, there is an economic incentive for the early detection of anomalies in the rate of change of degradation. This is because the identification of such events could be used as a trigger for preventive maintenance and intervention.

1.2 Anomaly Detection

Anomaly detection consists in finding patterns in data that do not conform to expected behaviour [15]. In the context of SHM, the anomalies can be referred as, among others, changes in the structural behaviour, outliers, and sensor drifts. The changes in the structural behaviour are commonly caused by ageing, usage, and environmental changes. On the other hand, the presence of an outlier might be due to either a high variability in the measurements or by malfunctioning sensors. The sensor drift can be caused by environmental conditions, installation conditions, and physical changes in the sensor. In practice, both the outliers and sensor drifts do not lead to a long-term deterioration of a structure. Therefore, the main interest of this thesis focuses on detecting changes in the structural behaviour. As a matter of fact, detecting early-stage changes and providing infrastructure maintenance in time have the potential to extend the infrastructure service life, avoiding costly replacement and service disruption.

In different application domains, the anomaly detection problem is known to be a challenging task [15]. In the context of SHM, challenges associated with anomaly detection

are:

- The generic notation of abnormal and normal behaviours is not well defined.
- Datasets with labeled anomalies are rarely available [16].
- Changes in the structural behaviour are commonly hidden under larger changes due to a high variability in environmental conditions such as temperature [17, 18].
- Distinguishing changes in the structural behaviour from the observation noise is difficult [18].

Although many methodologies from the field of applied statistics and machine learning have been proposed for detecting anomalies, they have only been applied to specific problems. The complete review of these methods are presented in Chapter 2. For the context of SHM, an anomaly detection system should be designed to meet the following requirements:

- It is capable of isolating the structural behaviour from raw structural responses including the effect of environmental conditions and observation noise.
- It provides a generic model architecture for the normal and abnormal behaviour without requiring labeled training data.
- It is capable of performing real-time anomaly detection without triggering any false alarms and without human supervision.
- It allows using the information redundancy contained in multiple datasets to provide the overall picture of the state of a structure.
- It is easily transferable from one structure to another.

1.3 Research Objectives

The research project aims at developing data-driven methods for SHM that are capable of tracking a structure's behaviour in real time. To achieve this main goal, four specific objectives need to be completed:

1. Develop a methodology for modeling complex periodic external effects caused by environmental conditions that influence the observed structural responses.
2. Develop a methodology for efficiently learning unknown model parameters from data.
3. Develop a methodology for detecting anomalies in real time and which is robust towards false alarms.
4. Validate the proposed methodologies with data collected on the full-scale structures.

1.4 Thesis Outline

This thesis is organized as follows. Chapter 2 provides a literature review for the field of SHM and exposes the strengths and limitations of existing methodologies for interpreting the observed structural responses. Chapter 3 presents the theory behind Bayesian Dynamic Linear Models (BDLMs) on which this thesis builds for detecting anomalies in the context of SHM. The goal of Chapter 3 is to provide (i) a mathematical formulation of existing BDLMs, (ii) a generic model architecture for anomaly detection, and (iii) new approaches for modeling non-harmonic periodic external effects. Chapter 4 introduces different algorithms for optimizing unknown model parameters in BDLMs. Chapter 5 introduces the theory and general framework proposed for performing real-time anomaly detection. For each chapter, the proposed methodologies are validated using either simulated or real datasets. Finally, Chapter 6 provides overall conclusions, discussions, and directions for future research.

1.5 Co-Authored Papers

Most of the work in this thesis has already been presented in the following co-authored publications:

- Nguyen, L.H. and Goulet, J.-A.. Structural health monitoring with dependence on non-harmonic periodic hidden covariates. *Engineering Structures*. 166:187 - 194, 2018.
- Nguyen, L.H. and Goulet, J.-A.. Anomaly detection with the switching Kalman filter for structural health monitoring. *Structural Control and Health Monitoring*. 25:e2136, 2018.
- Nguyen, L.H., Gaudot, I., Khazaeli, S., and Goulet, J.-A.. A Kernel-based method for modeling non-harmonic periodic phenomena in Bayesian dynamic linear models. *Frontiers in Built Environment*. 5:8, 2019.
- Nguyen, L.H., Gaudot, I., and Goulet, J.-A.. Uncertainty quantification for model parameters and hidden state variables in Bayesian dynamic linear models. *Structural Control and Health Monitoring*. 26(3):e2309, 2019.
- Nguyen, L.H. and Goulet, J.-A.. Real-time anomaly detection with Bayesian dynamic linear models. *Structural Control and Health Monitoring*. e2404. 2019.

CHAPTER 2 LITERATURE REVIEW

2.1 Structural Health Monitoring

Structural Health Monitoring (SHM) [19] consists in evaluating the state of civil infrastructures based on the measurements of structural responses in order to support their management. One of the goals with SHM is to gain insightful information about the state of a structure by interpreting its observed responses over time. SHM traditionally focuses on the physic-based models [20, 21] that use structural measurements to update a numerical model of a structure and then employs the model to predict the structural behaviour. For such an approach, abnormal events in a structure are identified by changes associated with modal parameters (e.g., mass, damping ratio, and stiffness) and structural vibration characteristics (e.g., natural frequencies) [22]. A key challenge of the physic-based models is that such models are not available for the majority of structures such as bridges and dams because they require prior detailed information about these structures [23]. Another challenge is that these models may not be representative of real-life structures for which the behaviour is affected by the environmental conditions (e.g., temperature and humidity), installation conditions (e.g., bridge supports and dam foundation), and ageing. These factors are commonly difficult to be incorporated in a physic-based model, and thus reduce the predictive capacity of the model on the structural behaviour [24].

Data-driven approaches [25–27] that employ machine learning techniques for interpreting structural responses, offer a promising path to mitigate the challenges of the physic-based models. These approaches interpret structural responses without requiring the specific details about structural properties. Furthermore, they can potentially model the dependency of structural responses on environmental conditions in order to increase model accuracy. The anomaly detection for a structure using the data-driven approach is commonly separated in two phases. The first phase consists of extracting the normal behaviour of a structure using a training dataset for which the abnormal events are absent. In the second phase, the changes in the structural behaviour are identified by comparing new observations with the resulting pattern behaviour. Although many methodologies based on the data-driven approaches have been proposed for interpreting the observed structural responses, it remains a challenge to reliably detect anomalies without also being adversely affected by false alarms.

Most of the methodologies employed to interpret SHM data take their origin in the fields of Machine Learning [28, 29]. Machine learning is commonly categorized into three types: *supervised learning*, *unsupervised learning*, and *reinforcement learning*. Supervised learning

consists in learning models that maps input data to desired output values. On the other hand, unsupervised learning is about finding an underlying structure in the data without knowing the right answer should be. Reinforcement learning involves an autonomous agent taking actions in a dynamic and uncertain environment by maximizing a long-term cumulative reward [30]. In the context of SHM, the majority of methodologies focus on the supervised and unsupervised learning. More specifically, main families of SHM methods based on supervised and unsupervised learning are *regression models* and *state-space models*. The strength and limitations of each of these families are described in the following section.

2.2 Regression Models

In the context of SHM, regression models consists in building models of the type

$$\mathbf{y} = \mathbf{g}(\mathbf{x}), \tag{2.1}$$

that is linking observed structural responses $\mathbf{y}$ to time-dependent covariates $\mathbf{x}$ such as temperature, traffic load in the case of bridges, or water level in the case of dams. Even if dozens of regression models exist, we focus our attention on the most commons: *Linear Regression* and *Neural Networks*. For both types of approaches, the standard procedure is to build a model $\mathbf{g}(\mathbf{x})$ using a training set, $\mathcal{D} = \{\mathbf{x}_i, y_i\}, \forall i = 1 : \mathtt{D}$, and then predict state of the structure for new covariates $\mathbf{x}_j, \forall j = 1 : \mathtt{P}$. The presence of anomalies are detected by comparing discrepancy between observations y_j and model predictions $\mathbf{g}(\mathbf{x}_j)$.

2.2.1 Linear Regression

Linear regression models are defined as

$$\mathbf{g}(\mathbf{x}) = \mathbf{a} \cdot \psi(\mathbf{x}) + \mathbf{b}, \tag{2.2}$$

where $\mathbf{a}$ and $\mathbf{b}$ are matrices containing unknown model parameters. Linear regression bears its name because the model is linear with respect to functions $\psi(\cdot)$ that take covariates as inputs. Because functions $\psi(\cdot)$ can be nonlinear, linear regression is not limited to modeling linear relationships between $\mathbf{x}$ and $\mathbf{y}$. Linear regression is employed to interpret the data recorded on structures where covariates are for example, water level, traffic load, temperature, etc.

In the field of dam monitoring, linear regression is employed in HST (Hydrostatic, Seasonal, Time) methods. HSTs have been employed in many case studies [31–34] to interpret

displacement, pressure, and flow-rate observations. The main idea of HST is to separate the observations into reversible (hydrostatic and seasonal) and irreversible components. Classic HST formulations cannot handle the situation where the observations depends on non-harmonic periodic covariates [35]. Similar methods such as *Hydrostatic-Temperature-Time* (HTT) [36, 37] and HST-Grad [35] employs directly the observed external effects such as concrete and water temperatures for addressing this limitation. When those data are not available, a superposition of harmonic functions can be employed for constructing in the non-harmonic periodic covariates [38]. However, it requires a large number of harmonic functions to deal with complex non-harmonic periodic covariates. In the field of bridge monitoring, multiple linear regression [26, 39, 40] is employed to describe the influence of the environmental conditions (e.g., temperature and traffic load) on the structural responses (e.g., strain and acceleration). In addition to the multiple linear regression, the robust regression [41, 42] is used for detecting structural damage. The robust regression method provides better regression coefficients for the model $\mathbf{g}(\mathbf{x})$ in the presence of outliers.

The key limitations of linear regression models are well known in the field of Machine Learning [28]:

- It does not distinguish between interpolating between observed data and extrapolating beyond observations.
- It is sensitive to outliers.
- It requires an explicit definition of basic function ψ.
- It requires observing covariates.
- It is unable to handle *auto-correlation* which is omnipresent in time-series data.

2.2.2 Neural Networks

Neural Networks (NNs) build a function $\mathbf{g}(\mathbf{x})$ by a succession of interconnected hidden layers. The advantages of NN methods are that (i) they allow modeling complex and nonlinear phenomena, and (ii) they can provide a better accuracy than the linear regression models for the prediction task. Model parameters associated with neural networks are learned using the *backpropagation* algorithm [43]. Since 2006, Deep Neural Networks, that is, *deep learning* have been at the forefront of the fields of machine learning and artificial intelligence [44]. The main reasons for this interest are the advances in computational power combined with the explosion of dataset sizes. In the field of dam monitoring, the potential of the NN approaches have been illustrated on interpreting the displacement [45–49], piezometric water level [50], and flow rates [51]. In the field of bridge monitoring, Pandey and Barai [52] have illustrated

the application of the NN method on detecting damage in a truss bridge. Gu et al. [53] have employed the NN method for differentiating the changes in natural frequencies caused by damage from those induced by temperature variations in two-span grid model. Recently, Tang et al. [54] have used the convolution neural network, that is, a variant of NN method, for improving the anomaly detection accuracy in a long-span cable-stayed bridge. Other authors [55–57] have applied the NN method to damage detection in structures based on modal parameters and structural vibration characteristics.

The main limitation of NN models in the context of SHM is that they can contain thousands of parameters. To learn these parameters, datasets containing up to millions of points are required in order for them to outperform other methodologies [44]. This aspect limits the utilization of neural network methods as a regression tool for SHM.

2.2.3 Common Limitations

As the subfield of the supervised machine learning, a first limitation common to all regression models is that once the model $\mathbf{g}(\mathbf{x})$ is built using a training set, it stops evolving as new data are collected. It means that the regression models are unable to improve themselves using the data collected beyond the training period.

A second limitation is that the anomaly detection method is based on a *hypothesis-testing* procedure [15]. In common cases, an error vector $\mathbf{e} = \mathbf{y} - \mathbf{g}(\mathbf{x})$ is evaluated for the training set $\mathcal{D} = \{x_i, y_i\}, \forall i = 1 : \mathtt{D}$. When the error vector $\mathbf{e}$ follows a Gaussian process, $\mathbf{e} \sim \mathcal{N}(\mu, \sigma^2)$, μ and σ^2 are estimated as

$$\begin{aligned} \mu &= \frac{1}{\mathtt{D}} \sum_{i=1}^{\mathtt{D}} e_i \\ \sigma^2 &= \frac{1}{\mathtt{D}-1} \sum_{i=1}^{\mathtt{D}} (e_i - \mu)^2. \end{aligned}$$

Figure 2.1 shows an example of the error vector $\mathbf{e}$ that is evaluated using a training set $\mathcal{D}$, along with its probability density function (PDF). $f(e_{\mathrm{Tr}})$ is employed to estimate a reference confidence interval that is defining the normal condition. Once the training is completed, the PDF of errors $f(e_{\mathrm{T}})$ is estimated for successive discrete test periods. The presence of anomalies is then tested based on the distance between the training and test-set confidence regions. Figure 2.2 shows an example where the normal statistics are learned using a training set and where anomalies are sought during three test-sets T1, T2, and T3. Here, anomalies are only present in the sets T2 and T3. The solid line presents the mean error and its confidence interval is presented by the dashed line. The main limitation of this hypothesis-testing-based approach is that it involves two counteracting phenomena. If one chooses to employ long

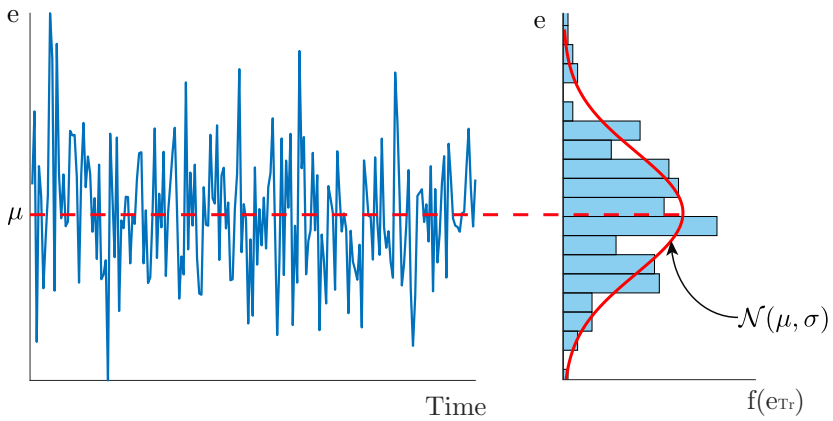


Figure 2.1 The left figure presents the errors with respect to time between observations and predicted values for the training set. The right figure shows the histogram and underlying probability density function for errors observed during the training period.

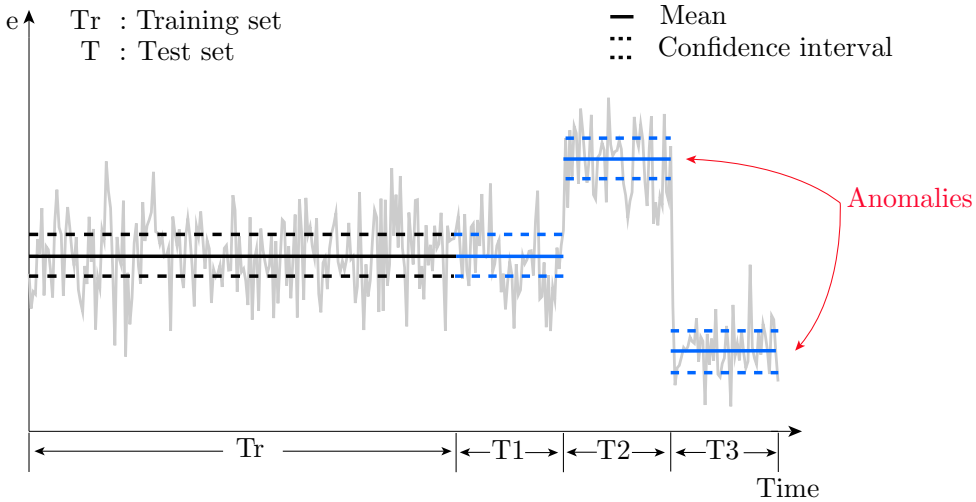


Figure 2.2 Illustration of the hypothesis-testing procedure for anomaly detection using a training set (Tr) and three test sets T1, T2, and T3 where anomalies only occur for T2 and T3.

test-window lengths to increase the robustness toward false alarms, it will delay the detection of anomalies. On the other hand, if one chooses to use short test window lengths, the approach becomes prone to false alarms caused by outliers.

A third limitation is that the regression models are not capable of learning non-stationary model parameters and thus affects their anomaly detectability in real time. The majority of regression models operate using a batch learning procedure [58] in which the model parameters are assumed to be stationary, and are estimated by minimizing a cost function, such as the prediction error, within a fixed training period. This assumption can no longer be held in the case where the underlying process in streaming data exhibits non-stationary behaviour over time [59, 60]. For this reason, the model parameters need to be relearned for each new data point and this is computationally demanding for large datasets. Therefore, the batch learning procedure is not well suited for performing real-time anomaly detection. Sejnowski and Rosenberg [61] has proposed a sliding window technique in which a small dataset from the past is kept for the learning purpose. This technique enables the model to learn continuously using only the data in the selected window and the new data point. However, the window length may become an issue because a short window-length might not have enough information for the learning purpose, while a long window-length can slow down the learning process.

2.3 State-Space Models

As an unsupervised learning method, state-space models (SSMs) do not stop learning after the training-set; they keep evolving over time as a new data point is collected. A SSM can be described in a generic form by the following functions

$$\begin{aligned} \mathbf{x}_t &= \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{w}_t) \\ \mathbf{y}_t &= \mathbf{h}(\mathbf{x}_t, \mathbf{v}_t), \end{aligned} \tag{2.3}$$

where $\mathbf{y}_t$ is a vector containing the observed structural responses at time t, $\mathbf{x}_t$ is the vector of hidden state variables, $\mathbf{f}$ is the transition model, $\mathbf{h}$ is the observation model, $\mathbf{w}_t$ is the model transition error, and $\mathbf{v}_t$ is the observation error. *Hidden state variables* are defined as the quantities of interest that are not directly observed and that are employed for explaining the observed behaviour of the structure. In comparison with the regression models presented in Section 2.2.1, the advantage of SSMs is to enable performing a dynamic estimation of the hidden state variables where $\mathbf{x}_t$ depends on $\mathbf{x}_{t-1}$, where the *Markov Hypothesis* [28] is commonly employed to reduce the complexity of models. The Markov Hypothesis supposes that the future is independent of the past given the present. In practice, it means that $\mathbf{x}_t$

only depends on $\mathbf{x}_{t-1}$ rather than on $\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{t-1}$. The most common SSMs in the context of civil infrastructure are autoregressive models, Kalman/particle filter-based models, and Bayesian dynamic linear models.

2.3.1 Autoregressive Models

Autoregressive models (ARMs) are employed to estimate the time-dependent responses of structures with the purpose of detecting anomalies based on hidden-state estimates. The basic model structure for ARMs is

$$\mathbf{x}_t = \boldsymbol{\theta} \cdot \mathbf{x}_{t-1} + \mathbf{w}_t, \tag{2.4}$$

where $\boldsymbol{\theta}$ is a vector of parameters to be estimated and $\mathbf{w}_t$ a vector of model errors at time t. A special case of ARM is the Autoregressive Moving Average Model (ARMA). It combines in a single model, an autoregressive and a moving average part. The moving average part limits the effect of outliers that the autoregressive part alone cannot handle. Another type of ARMs is the autoregressive model with exogenous terms (ARX). The ARX combines an autoregressive term with dependent variables that allows including the effects of covariates such as temperature, loading, etc. For all types of ARMs, the anomaly detection procedure relies on the comparison of the current estimates for hidden state variables with reference values estimated during a training period.

Carden and Brownjohn [62] have employed an ARMA model to classify damage scenarios on experimental data for the IASC-ASCE benchmark a four-storey frame structure, the Z24 bridge in Switzerland and the Malaysia-Singapore second link bridge. Omenzetter and Brownjohn [63] applied ARIMA to interpret strain histories from a full-scale bridge. Bodeaux and Golinval [64] have also illustrated the ARMA capacity for damage detection on the Steel-Quake structure at the Joint Research Center in Ispra (Italy). A hybrid model of ARMA and Kalman filter is used for interpreting the strain signal recording on the Malaysia-Singapore Second Link bridge [65]. Peeters et al. [66, 67] have employed an ARX model to monitor the frequency data collected on the Z24 bridge.

A first limitation of ARMs for SHM applications is related to their limited predictive capacity. Because ARMs contains no information specific to the data being modeled, it typically leads to a poor predictive capacity. A second limitation is that the model parameters are learned using the batch learning procedure, so the ARMs offer a limited performance for the real-time anomaly detection. A third limitation is that although the hidden state estimation is performed dynamically, the anomaly detection remains based hypothesis-testing

procedures similar to the one presented in §2.2.3.

2.3.2 Kalman/Particle Filter-Based Models

The Kalman Filter (KF) has been widely used in time series analysis for estimating the posterior probability of the current hidden state variables $\mathbf{x}_t$ given past and current observations $\mathbf{y}_{1:t}$. The KF are employed for linear models and the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are employed for handling the nonlinear models. Particle Filter (PF) is a sequential Monte Carlo methods for estimating the hidden state variables as well as parameters of nonlinear models.

In the field of civil engineering, those methods are commonly employed for estimating the structural responses associated with the modal parameters of a system over time, given observations. For example, Tien et al. [68] have employed a KF to estimate the inter-story drift of a 10-story shear-type building. Wu and Smyth [69] have illustrated the EKF and UKF performances on the nonlinear hysteretic Bouc-Wen system with a single-degree freedom, a linear system with two degree-of-freedoms (DOFs), and a nonlinear elastic system with two DOFs. Chatzi and Smyth [70, 71] have examined the PF performance on a three-mass-damped system. In addition, the Kalman/Particle filter-based models are applied to the online parameter estimation in several applications [72–76]. For these applications, the anomalies are identified in real time by changes in modal parameters.

Most applications of Kalman/particle filter-based models are illustrated using either the simulated data obtained from a numerical model or data collected on laboratory structures for which a physic-based model with some assumptions has been established. However, such a physic-based model has seldom been available for full-scale structures such as bridges and dams because of their complexity and time-consuming nature. Another limitation of this type of approach is that the anomaly detection is based on changes in the modal parameters in comparison with reference values defined during a training period. Like other regression models described in §2.2.3, this anomaly detection procedure is based on hypothesis-testing.

2.3.3 Common Limitations

The common limitation of current state-space methods is that the model complexity is not adapted to the task. In the case of ARM, models are overly simple, limiting their predictive capacity. In the case of Kalman/particle filter-based models, models require advanced knowledge about a structure, which is not available for full-scale civil-engineering structures. Another limitation of current state-space methods is that although they allow overcoming

some of the limitations associated with regression models, the anomaly detection procedures remain based on hypothesis-testing, which is subject to the same limitation as described in §2.2.3.

2.3.4 Bayesian Dynamic Linear Models

The current factor limiting widespread SHM applications is the lack of generic data-interpretation methods that can be employed at low cost, for any structures. Goulet [77] proposed to address this challenge by building on the work done in the fields of machine learning in what is known as state-space models [28], and in applied statistics what is known as *Bayesian Dynamic Linear Models* (BDLMs) [78–81]. This methodology consists in the decomposition of the time series recorded on structures into a vector of hidden state variables. The vector of possible generic components includes, a *baseline* component to model the structural behaviour, a *local trend* to model the rate of change, a *periodic component* to model the effect of periodic external effects such as temperature, an *autoregressive component* to describe time-dependent model errors, and a *regression component* to include the effect of an *observed covariates*, e.g., loading and water temperature, on the structural response. An example of BDLMs for decomposing a structure response into a set of hidden state variables is illustrated in Figure 2.3. The mean values of hidden state variables and its uncertainty bounds are presented by the solid line and shaded region. In this example, the displacement data are decomposed into a baseline, a periodic component, and an autoregressive component.

BDLMs can handle harmonic periodic patterns such as the effect of temperature on a structural response. Moreover, this can be achieved whether or not the temperature is observed. However, one limitation of BDLMs is that it is unable to handle complex non-harmonic periodic patterns unless they are directly observed. The requirement that non-harmonic periodic patterns must be directly observed is often a difficult constraint for SHM applications. Another limitation is that the current form of BDLMs can only model behaviour of time series under stationary conditions. To detect the occurrence of anomalies, it needs to be extended to operate in non-stationary conditions. Furthermore, the task of the model parameter calibration in BDLMs still relies on the batch learning procedure which is sensitive to initial parameter values and local maxima.

2.4 Conclusion

This literature review identifies that current data-driven methodologies for SHM can be separated into two main categories, regression models and state-space models. Both of models

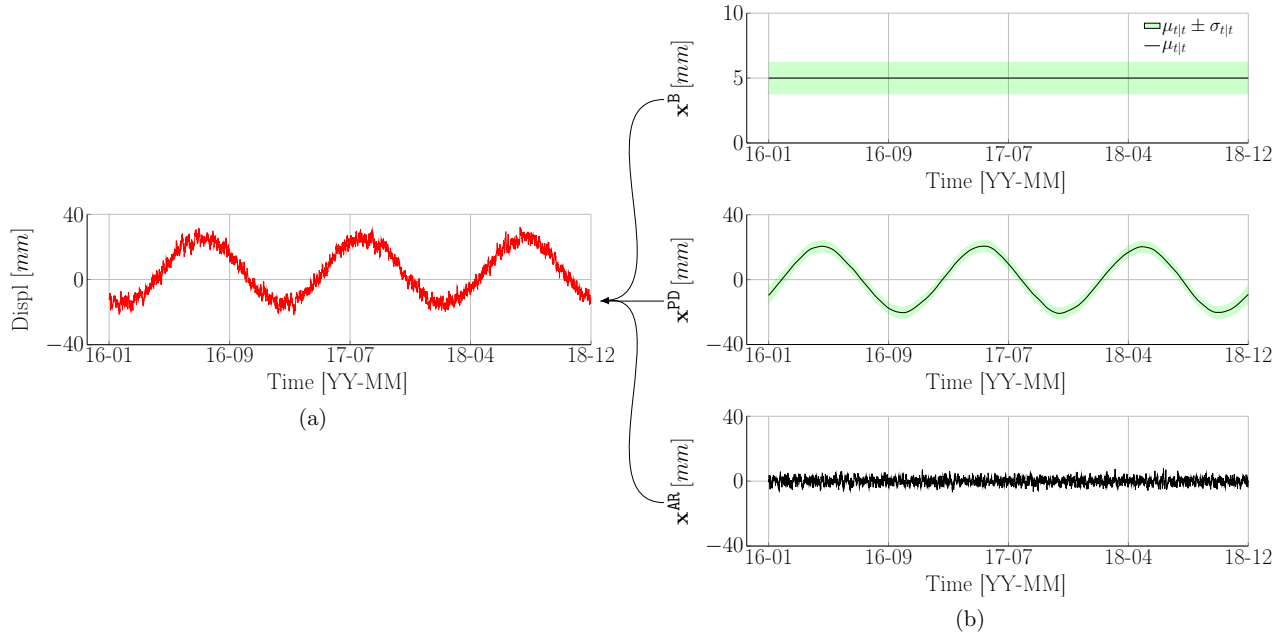


Figure 2.3 Illustration of structural responses (a) that are decomposed in a set of hidden state variables (b): baseline component ($\mathbf{x}^{\mathtt{B}}$), periodic component ($\mathbf{x}^{\mathtt{PD}}$), and autoregressive component ($\mathbf{x}^{\mathtt{AR}}$). The mean values of the hidden state variables, $\mu_{t|t}$, and its bound uncertainty, $\mu_{t|t} \pm \sigma_{t|t}$ are presented by the solid line and shaded region.

share two common limitations; there is a lack of generic methodologies for detecting anomalies and model parameters are assumed to be stationary. In common cases, anomalies are detected using hypothesis-testing based approaches. These approaches are (i) prone to false alarms, (ii) incur long anomaly detection time, and (iii) computationally inefficient for performing the real-time anomaly detection. The assumption of invariant model parameters can no longer be held in the presence of anomalies, where several model parameters exhibit a non-stationary behaviour.

The majority of the methods proposed for SHM have only been validated using either the simulated measurements obtained from the numerical model or the measurements collected from laboratory structures. Yet, they have seldom been tested using the measurements recorded on the real-life structures. Furthermore, regression models have a limited predictive ability beyond the training period. Although the state-space models solve some of the limitations associated regression models, it remains that current methods employ models that are either too simple or too complex for the task of detecting anomalies in the context of SHM. Simplistic models are limited by their poor predictive capacity and over-complex

models require detailed information about a structure, which is not suited for a widespread deployment across thousands of bridges and dams that are all different from one to another.

CHAPTER 3 BAYESIAN DYNAMIC LINEAR MODELS

3.1 Introduction

This chapter presents the theory behind existing *Bayesian Dynamic Linear Models* (BDLMs) on which this dissertation builds for detecting anomalies in time series. More specifically, Section 3.2 reviews the mathematical background as well as the generic components of the BDLMs. This section also exposes the common limitations in the current form of the BDLMs for detecting anomalies and for handling external effects (e.g., temperature and traffic loading). Section 3.3 presents the theory of existing *Switching Kalman Filter* (SKF) that, in common cases, is used for handling non-stationary time series. Section 3.4 then proposes a generic model architecture for coupling the SKF with BDLMs to detect anomalies. In the context of SHM, changes in the structural behaviour are commonly masked by larger changes due to a high variability in the external effects. Therefore, a well separation between the external effects and the structural behaviour is a key aspect for anomaly detection. For this purpose, Section 3.5 and 3.6 present the new methodologies that enable the existing BDLMs to model efficiently the external effects while overcoming their common limitations. The main contributions of this chapter are:

- proposing a generic model architecture for anomaly detection without requiring labeled training data.
- developing the methodologies for handling the non-harmonic periodic external effects.
- validating the proposed methodologies with data recorded on real-life structures.

3.2 Theory of Bayesian Dynamic Linear Models

This section reviews the theory behind BDLMs, which are a special case of state-space models (SSMs). A BDLM consists of two linear models defined by an *observation* model and a *transition* model. The observation model is employed to describe the relation between observations $\mathbf{y}_t$ and hidden state variables $\mathbf{x}_t$ at time $t \in [1 : \mathtt{T}]$. The transition model describes the dynamics of the hidden states variables over time. The mathematical formulations for both models are defined as

Observation model

$$\mathbf{y}_t = \mathbf{C}_t\mathbf{x}_t + \mathbf{v}_t, \quad \begin{cases} \mathbf{y}_t \sim \mathcal{N}(\mathbb{E}[\mathbf{y}_t], \mathrm{cov}[\mathbf{y}_t]) \\ \mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma_t}) \\ \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R_t}) \end{cases} \tag{3.1}$$

Transition model

$$\mathbf{x}_t = \mathbf{A}_t\mathbf{x}_{t-1} + \mathbf{w}_t, \quad \left\{ \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_t}), \right. \tag{3.2}$$

where $\mathbf{C}_t$ is the observation matrix, $\mathbf{v}_t$ is the Gaussian observation error with mean zero and covariance matrix $\mathbf{R}_t$, $\mathbf{A}_t$ is the transition matrix, and $\mathbf{w}_t$ is the Gaussian model error with mean zero and covariance matrix $\mathbf{Q}_t$. The main strength of BDLMs for SHM applications is the capacity to model a variety number of structural responses from a limited vector of hidden state variables such as a baseline component, local trend, periodic component, and regression components. Further details regarding the hidden state variables are provided by Goulet [77] and West & Harrison [79].

3.2.1 Kalman Filter/Smoother

In BDLMs, the hidden state variables $\mathbf{x}_t$ are estimated using either the Kalman filter [28] or the UD filter [82]. Note that the UD filter yields a numerically more stable performance than the Kalman filter yet, it is slightly more computationally demanding. This Kalman/UD filter algorithm is a two-step iterative process that estimates the posterior mean vector $\boldsymbol{\mu}_{t|t}$ and covariance matrix $\boldsymbol{\Sigma}_{t|t}$ so that

Prediction step

$$\begin{aligned} p(\mathbf{x}_t|\boldsymbol{\theta}_t, \mathbf{y}_{1:t-1}) &= \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}) && \text{Prior state estimate} \\ \boldsymbol{\mu}_{t|t-1} &\triangleq \mathbf{A}_t\boldsymbol{\mu}_{t-1|t-1} && \text{Prior expected value} \\ \boldsymbol{\Sigma}_{t|t-1} &\triangleq \mathbf{A}_t\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{A}_t^{\intercal} + \mathbf{Q}_t && \text{Prior covariance,} \end{aligned} \tag{3.3}$$

where $\boldsymbol{\theta}_t$ is a vector of unknown model parameters included in the model matrices $\{\mathbf{A}_t, \mathbf{C}_t, \mathbf{Q}_t, \mathbf{R}_t\}$.

Measurement step

$$
\begin{aligned}
p(\mathbf{x}_t|\boldsymbol{\theta}_t, \mathbf{y}_{1:t}) &= \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}) && \text{Posterior state estimate} \\
\boldsymbol{\mu}_{t|t} &= \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t \mathbf{o}_t && \text{Posterior expected value} \\
\boldsymbol{\Sigma}_{t|t} &= (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t)\boldsymbol{\Sigma}_{t|t-1} && \text{Posterior covariance} \\
\mathbf{o}_t &\triangleq \mathbf{y}_t - \hat{\mathbf{y}}_t && \text{Innovation vector} \\
\hat{\mathbf{y}}_t &\triangleq \mathbb{E}[\mathbf{y}_t|\mathbf{y}_{1:t-1}] = \mathbf{C}_t \boldsymbol{\mu}_{t|t-1} && \text{Predicted observations vector} \\
\mathbf{K}_t &\triangleq \boldsymbol{\Sigma}_{t|t-1}\mathbf{C}_t^{\intercal}\mathbf{O}_t^{-1} && \text{Kalman gain matrix} \\
\mathbf{O}_t &\triangleq \mathbf{C}_t \boldsymbol{\Sigma}_{t|t-1}\mathbf{C}_t^{\intercal} + \mathbf{R}_t && \text{Innovation covariance matrix.}
\end{aligned}
\tag{3.4}
$$

The Kalman filter algorithm uses the *Kalman gain* $\mathbf{K}_t$ to weight the information coming from observations $\mathbf{y}_t$, in comparison with the information coming from prior knowledge. The KF algorithm is summarized in its short form as

$$
\left(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}, \mathcal{L}_t\right) = \text{Filter}\left(\boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1}, \mathbf{y}_t, \mathbf{A}_t, \mathbf{C}_t, \mathbf{Q}_t, \mathbf{R}_t\right), \tag{3.5}
$$

where $\mathcal{L}_t$ is the marginal likelihood at time t. With BDLMs, the marginal likelihood is defined by a multivariate Gaussian distribution following

$$
\begin{aligned}
\mathcal{L}_t &= p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_t) \\
&= \mathcal{N}(\mathbf{y}_t; \mathbf{C}_t \boldsymbol{\mu}_{t|t-1}, \mathbf{C}_t \boldsymbol{\Sigma}_{t|t-1}\mathbf{C}_t^{\intercal} + \mathbf{R}_t).
\end{aligned}
\tag{3.6}
$$

The vector of model parameters, $\boldsymbol{\theta}_t$, needs to be estimated from the data. The common approaches for this estimation task can be the point-estimation and Bayesian methods such as *Maximum Likelihood Estimation* (MLE) and *Markov Chain Monte Carlo* (MCMC). The details of these methods are presented in Chapter 4.

The offline estimation for the hidden state variables $\mathbf{x}_t$ at time t is performed using the Kalman Smoother (KS) [28]. The mathematical formulation of the KS algorithm is written as

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{y}_{1:\mathrm{T}}) &= \mathcal{N}(\mathbf{x}_\mathrm{T}; \boldsymbol{\mu}_{t|\mathrm{T}}, \boldsymbol{\Sigma}_{t|\mathrm{T}}) \\
\boldsymbol{\mu}_{t|\mathrm{T}} &= \boldsymbol{\mu}_{t|t} + \mathbf{J}_t\left(\boldsymbol{\mu}_{t+1|\mathrm{T}} - \boldsymbol{\mu}_{t+1|t}\right) && \text{Posterior expected value} \\
\boldsymbol{\Sigma}_{t|\mathrm{T}} &= \boldsymbol{\Sigma}_{t|t} + \mathbf{J}_t\left(\boldsymbol{\Sigma}_{t+1|\mathrm{T}} - \boldsymbol{\Sigma}_{t+1|t}\right)\mathbf{J}_t^{\intercal} && \text{Posterior covariance} \\
\mathbf{J}_t &\triangleq \boldsymbol{\Sigma}_{t|t}\mathbf{A}_t^{\intercal}\boldsymbol{\Sigma}_{t+1|t}^{-1} && \text{Backward Kalman gain matrix.}
\end{aligned}
\tag{3.7}
$$

The difference between KF and KS lies on the estimation of the hidden state variables. The smoothed hidden states are conditioned over all observations $\mathbf{y}_{1:\mathrm{T}}$, while the filtered hidden

states are conditional only on the observations from the previous time steps $\mathbf{y}_{1:t}$. The initial values for the hidden state variables estimated using the KS are values obtained from the last step of the Kalman filter, that is, $\left(\boldsymbol{\mu}_{\mathtt{T}|\mathtt{T}}, \boldsymbol{\Sigma}_{\mathtt{T}|\mathtt{T}}\right)$. The smoother step is summarized in the following short form,

$$\left(\boldsymbol{\mu}_{t|\mathtt{T}}, \boldsymbol{\Sigma}_{t|\mathtt{T}}\right) = \text{Smoother}(\boldsymbol{\mu}_{t+1|\mathtt{T}}, \boldsymbol{\Sigma}_{t+1|\mathtt{T}}, \boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}, \mathbf{A}_t, \mathbf{Q}_t). \tag{3.8}$$

3.2.2 Modeling External Effects

In the context of SHM, the observed structural responses are commonly dependent on the environmental and operational conditions, that is, *external effects*, such as temperature, traffic load, wind, and humidity [83–85]. This section covers the existing approaches in the current form of the BDLMs for modeling periodic external effects.

Fourier Form

The *Fourier form* component allows modeling sine-like phenomena. Its mathematical formulation is written as

$$\mathbf{x}_t^{\mathtt{F}} = \begin{bmatrix} x_t^{\mathtt{F}_1} \\ x_t^{\mathtt{F}_2} \end{bmatrix}, \ \mathbf{A}_t^{\mathtt{F}} = \begin{bmatrix} \cos\omega & \sin\omega \\ -\sin\omega & \cos\omega \end{bmatrix}, \ \mathbf{C}_t^{\mathtt{F}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^{\mathtt{T}}, \ \mathbf{Q}_t^{\mathtt{F}} = (\sigma_w^{\mathtt{F}})^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where $\omega = \frac{2\pi \cdot \Delta t}{\mathtt{P}}$ is the angular frequency defined by the period $\mathtt{P}$ of the phenomena modeled, and the time-step length, Δt. Note that the Fourier form component is represented by two hidden state variables $x_t^{\mathtt{F}_1}$ and $x_t^{\mathtt{F}_1}$ [77]. Yet, only the first component contributes to the observation. Figure 3.1 presents examples of realizations of a Fourier form component. Although the Fourier form component is computationally efficient, it is limited to modeling

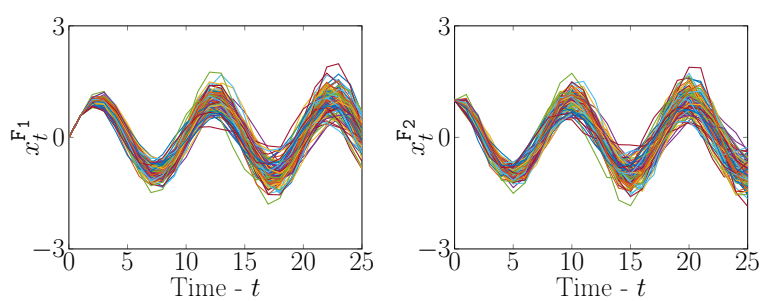


Figure 3.1 Examples of realizations of $\mathbf{x}^{\mathtt{F}}$ for a Fourier form component with parameters $\sigma_w^{\mathtt{F}} = 0.01$ and $\mathtt{P} = 10$.

simple harmonic periodic phenomena such as the one presented in Figure 3.2a. It is possible

to model simple non-harmonic periodic phenomena such as the one presented in Figure 3.2b using a superposition of Fourier form components each having a different period. Nevertheless this process is difficult to employ in practice, especially when the complexity of the periodic pattern increases because it requires a significant increase in the number of hidden state variables as well as of an unknown period parameter to be identified, which in turn decreases the overall computationally efficiency of the approach.

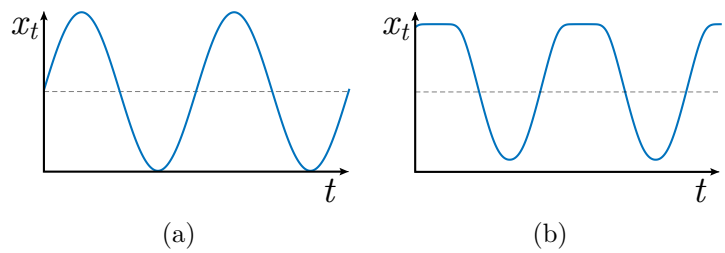


Figure 3.2 Examples of (a) harmonic and (b) non-harmonic periodic pattern.

Dynamic Regression

When observed, external effects influencing the responses of structures, that is, *covariates* can be included as a regressor in the BDLMs. One approach to include the effect of these observed covariates is to employ a *Dynamic Regression Component* [79]. In the dynamic regression component, the dynamic regression coefficient is treated as an unknown state variable $x_t^{\mathtt{DR}}$, whose temporal evolution follows a random walk. This random walk is parameterized by a transition matrix $\mathbf{A}_t^{\mathtt{DR}} = 1$, and by a model error covariance matrix $\mathbf{Q}_t^{\mathtt{DR}} = (\sigma^{\mathtt{DR}})^2$. For $\sigma^{\mathtt{DR}} = 0$, the dynamic regression coefficient $x_t^{\mathtt{DR}}$ is assumed to be stationary in time, for $\sigma^{\mathtt{DR}} > 0$, the dynamic regression coefficient $x_t^{\mathtt{DR}}$ is assumed to be changing over time (non-stationary). The regressor, that is, the observed covariate $y_t^{\mathtt{DR}}$, is placed directly in the observation matrix so that $\mathbf{C}_t^{\mathtt{DR}} = y_t^{\mathtt{DR}}$.

3.3 Theory of Switching Kalman Filter

The literature review in Chapter 2 has identified that most anomaly detection approaches share a common limitation; they employ a hypothesis-testing procedure. This section presents the theory of existing *Switching Kalman Filter* (SKF) [86] which has the potential to overcome the limitations of the traditional hypothesis-testing procedure. The SKF models non-stationary systems by estimating the probability of multiple model classes over time steps. However, a limitation of SKF is that the number of possible sequences of transitions between model

classes grows exponentially with time steps. Figure 3.3a presents the exponential increase in the number of sequences of states. Here, starting from $\mathtt{S} = 2$ possible initial states at time t, it leads to 8 possible sequences at $t+2$. It means that the number of sequences of states is 2^n at the nth time step. This exponential increase leads to an intractable inference. The key aspect for addressing this limitation is to employ the *collapsing* approximation known as the *Generalized Pseudo Bayesian* (GPB) algorithm of order r [87, 88], where it only considers a total of $\mathtt{S}^r$ possible sequences of states that then collapse into $\mathtt{S}$ states after each time step. Figure 3.3b presents an example of the GPB algorithm of order 2 ensuring that 2^2 possible sequences of states merge into 2 states after each time step. At each time step, GPB-2

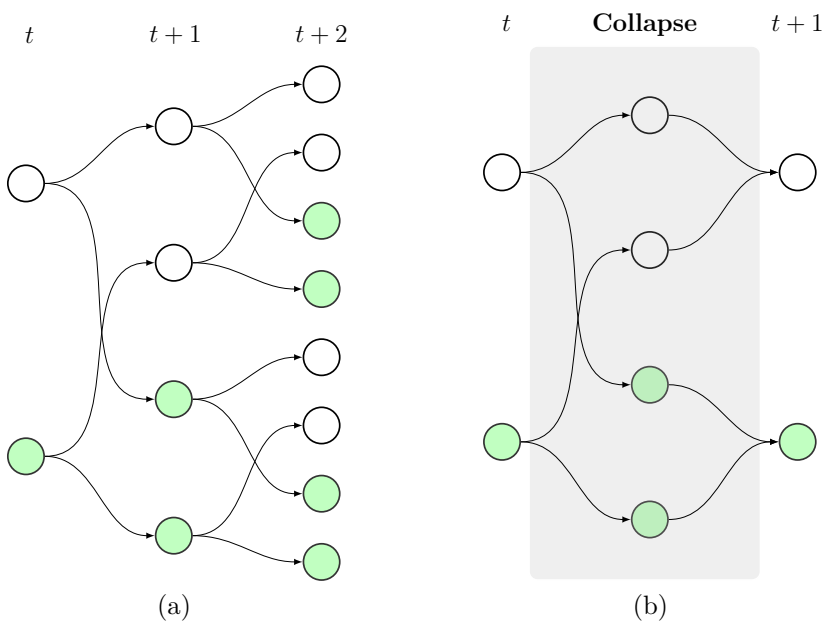


Figure 3.3 Illustration of (a) how the number of state sequences grows exponentially with time and (b) the Generalized Pseudo Bayesian algorithm of order 2.

estimates the current probability of each state irrespectively of the state at previous time steps. In practice, this is achieved by approximating the mixture of 4 Gaussian PDFs with a mixture of 2 Gaussian PDFs in which each Gaussian PDF is approximated by a mixture of 2 Gaussian PDFs [87, 89, 90].

3.3.1 SKF-filter Step

For a time series where $t \in [1 : \mathtt{T}]$, the observation and transition equations for the SKF remain identical to those presented in Section 3.2 for the BDLM. However, the notation for the

Kalman filter (KF) algorithm needs to be adapted to include the *Markov-switching variable* $s_t \in \{1, 2, 3, \ldots, \mathtt{S}\}$ corresponding to *filtering models*. Assume that the Markov-switching variables at time t and $t-1$ are $s_{t-1} = i$ and $s_t = j$. We use the *superscript* inside the parentheses $^{i(j)}$ to denote the current state j at the time t given the state i at time $t-1$. For the SKF, the short form of the Kalman filter algorithm presented in Equation 3.5 are rewritten as

$$\left(\mathbf{x}_{t|t}^{i(j)}, \boldsymbol{\Sigma}_{t|t}^{i(j)}, \mathcal{L}_t^{i(j)}\right) = \text{Filter}\left(\boldsymbol{\mu}_{t-1|t-1}^{i}, \boldsymbol{\Sigma}_{t-1|t-1}^{i}, \mathbf{A}_t^{i(j)}, \mathbf{C}_t^{i(j)}, \mathbf{Q}_t^{i(j)}, \mathbf{R}_t^{i(j)}\right), \tag{3.9}$$

where $\mathcal{L}_t^{i(j)}$ measures the marginal likelihood that the state at time $t-1$ was $s_{t-1} = i$ and that it switches to $s_t = j$ at time t. The marginal likelihood of such as switch $\mathcal{L}_t^{i(j)}$ is defined as

$$\begin{aligned} \mathcal{L}_t^{i(j)} &= p(\mathbf{y}_t | s_t = j, s_{t-1} = i, \mathbf{y}_{1:t-1}) \\ &= \mathcal{N}\left(\mathbf{y}_t; \mathbf{C}_t^{i(j)} \boldsymbol{\mu}_{t|t-1}^{i(j)}, \mathbf{C}_t^{i(j)} \boldsymbol{\Sigma}_{t|t-1}^{i(j)} \left(\mathbf{C}_t^{i(j)}\right)^{\mathsf{T}} + \mathbf{R}_t^{i(j)}\right). \end{aligned} \tag{3.10}$$

For common cases, the model uncertainty is described by the model error covariance matrix $\mathbf{Q}_t^{i(j)}$ which depends on the state i at time $t-1$ and the state j at time t. The matrices defining the transition and observation models are only dependent on the arrival state j at time t,

$$\mathbf{A}_t^{i(j)} = \mathbf{A}_t^{j}, \quad \mathbf{C}_t^{i(j)} = \mathbf{C}_t^{j}, \quad \mathbf{R}_t^{i(j)} = \mathbf{R}_t^{j}. \tag{3.11}$$

Therefore, Equation 3.12 becomes

$$\left(\mathbf{x}_{t|t}^{i(j)}, \boldsymbol{\Sigma}_{t|t}^{i(j)}, \mathcal{L}_t^{i(j)}\right) = \text{Filter}\left(\boldsymbol{\mu}_{t-1|t-1}^{i}, \boldsymbol{\Sigma}_{t-1|t-1}^{i}, \mathbf{A}_t^{j}, \mathbf{C}_t^{j}, \mathbf{Q}_t^{i(j)}, \mathbf{R}_t^{j}\right). \tag{3.12}$$

The short form for the Kalman smoother in Equation 3.8 is adapted as

$$\left(\boldsymbol{\mu}_{t|\mathtt{T}}^{(j)k}, \boldsymbol{\Sigma}_{t|\mathtt{T}}^{(j)k}\right) = \text{Smoother}(\boldsymbol{\mu}_{t+1|\mathtt{T}}^{k}, \boldsymbol{\Sigma}_{t+1|\mathtt{T}}^{k}, \boldsymbol{\mu}_{t|t}^{j}, \boldsymbol{\Sigma}_{t|t}^{j}, \mathbf{A}_t^{k}, \mathbf{Q}_t^{(j)k}), \tag{3.13}$$

where k is the state at time $t+1$.

3.3.2 SKF-collapse Step

The mean vector $\boldsymbol{\mu}_{t|t}^{j}$ and covariance matrix $\boldsymbol{\Sigma}_{t|t}^{j}$ are computed by collapsing the filtering models according to their weights $p(s_{t-1} = i | s_t = j, \mathbf{y}_{1:t})$. Note that the filtering model uses the classic Kalman filter algorithm for estimating the mean vector, covariance matrix, and

marginal likelihood. To further describe the collapse step, let us introduce the notation

$$\begin{aligned}
p(s_{t-1}=i|\mathbf{y}_{1:t-1}) &= \pi^{i}_{t-1|t-1} && \text{Previous state probability} \\
p(s_t=j|s_{t-1}=i) &= Z^{i(j)} && \text{Transition probability} \\
p(s_{t-1}=i, s_t=j|\mathbf{y}_{1:t}) &= \mathbf{M}^{i(j)}_{t-1,t|t} && \text{Joint probability} \\
p(s_{t-1}=i|s_t=j, \mathbf{y}_{1:t}) &= \mathbf{W}^{i(j)}_{t-1|t} && \text{State switching probability.}
\end{aligned} \tag{3.14}$$

The transition probability matrix $\mathbf{Z}_t$ is defined by the number of states. For $\mathtt{S}$ states $s_t \in \{1, 2, 3 \cdots, \mathtt{S}\}$, the matrix $\mathbf{Z}_t$ is defined as

$$\mathbf{Z}_t = \begin{bmatrix} Z^{11} & Z^{12} & \cdots & Z^{1\mathtt{S}} \\ Z^{21} & Z^{22} & \cdots & Z^{2\mathtt{S}} \\ \vdots & \vdots & \ddots & \vdots \\ Z^{\mathtt{S}1} & Z^{\mathtt{S}2} & \cdots & Z^{\mathtt{S}\mathtt{S}} \end{bmatrix}$$

where $Z^{ij} \equiv Z^{i(j)}$ with $\sum_{j=1}^{\mathtt{S}} Z^{ij} = 1$. The joint probability of $s_t = j$ and $s_{t-1} = j$, given $\mathbf{y}_{1:t}$ is evaluated as

$$\mathbf{M}^{i(j)}_{t-1,t|t} = \frac{\mathcal{L}^{i(j)}_{t|t} \cdot Z^{i(j)} \cdot \pi^{i}_{t-1|t-1}}{\sum_{j=1}^{\mathtt{S}} \sum_{i=1}^{\mathtt{S}} \mathcal{L}^{i(j)}_{t|t} \cdot Z^{i(j)} \cdot \pi^{i}_{t-1|t-1}}, \tag{3.15}$$

The denominator of Equation 3.15 is a normalization constant ensuring that

$$\sum_{j=1}^{\mathtt{S}} \sum_{i=1}^{\mathtt{S}} \mathbf{M}^{i(j)}_{t-1,t|t} = 1.$$

The marginal probability of $s_t = j$ is obtained through marginalization following

$$\pi^{j}_{t|t} = \sum_{i=1}^{\mathtt{S}} \mathbf{M}^{i(j)}_{t-1,t|t}. \tag{3.16}$$

The collapsed mean vector $\boldsymbol{\mu}^{j}_{t|t}$ and covariance matrix $\boldsymbol{\Sigma}^{j}_{t|t}$ are obtained using Gaussian mixture approximation

$$\begin{aligned}
\boldsymbol{\mu}^{j}_{t|t} &= \sum_{i=1}^{\mathtt{S}} \boldsymbol{\mu}^{i(j)}_{t|t} \cdot \mathbf{W}^{i(j)}_{t-1|t} \\
\boldsymbol{\Sigma}^{j}_{t|t} &= \sum_{i=1}^{\mathtt{S}} \left[\mathbf{W}^{i(j)}_{t-1|t} \cdot \left(\boldsymbol{\Sigma}^{i(j)}_{t|t} + \mathbf{m}\mathbf{m}^{\intercal} \right) \right],
\end{aligned} \tag{3.17}$$

where

$$\begin{aligned} \mathbf{W}_{t-1|t}^{i(j)} &= \frac{\mathbf{M}_{t-1,t|t}^{i(j)}}{\pi_{t|t}^{j}} \\ \mathbf{m} &= \boldsymbol{\mu}_{t|t}^{i(j)} - \boldsymbol{\mu}_{t|t}^{j}. \end{aligned} \tag{3.18}$$

The short-form notation for the collapse step is

$$\left(\boldsymbol{\mu}_{t|t}^{j}, \boldsymbol{\Sigma}_{t|t}^{j}, \pi_{t|t}^{j}\right) = \text{Collapse}\left(\boldsymbol{\mu}_{t|t}^{i(j)}, \boldsymbol{\Sigma}_{t|t}^{i(j)}, \mathbf{W}_{t-1|t}^{i(j)}\right).$$

For the purpose of simplicity, SKF-filter and SKF-collapse steps are summarized in a short form as

$$(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}, \mathcal{L}_t, \boldsymbol{\pi}_{t|t}) = \text{SKF}(\boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1}, \mathbf{y}_t, \mathbf{A}_t, \mathbf{C}_t, \mathbf{Q}_t, \mathbf{R}_t, \mathbf{Z}_t, \boldsymbol{\pi}_{t-1|t-1}). \tag{3.19}$$

An example of the SKF employed for describing the transition between two possible models is presented in Figure 3.4. The goal is to evaluate the mean vector $\boldsymbol{\mu}_{t|t}^{j}$ and the covariance

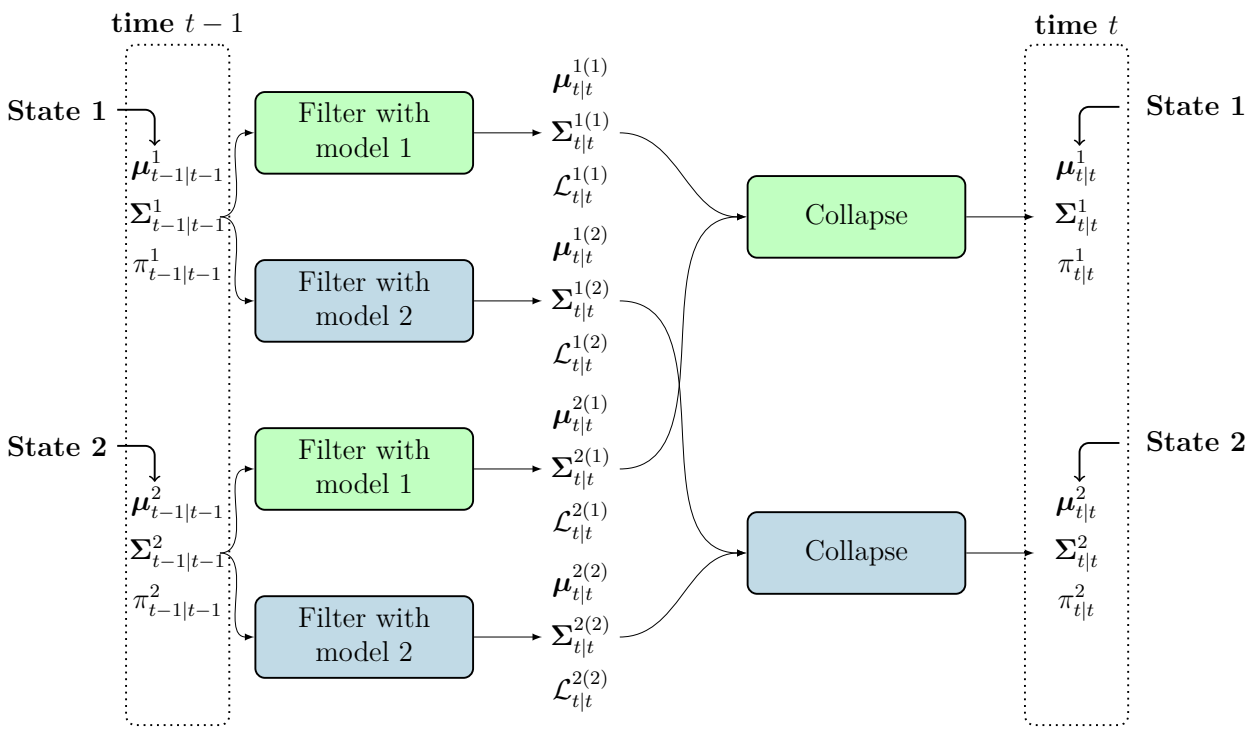


Figure 3.4 Illustration of the SKF algorithm for two states each having its own transition model. $^{(\cdot)}$ indicates the filtering model being used for computation.

matrix $\boldsymbol{\Sigma}_{t|t}^{j}$ for each model, $j \in \{1, 2\}$ along with the probability $\pi_{t|t}^{j}$ of each model at the

time t, given the same two models at time $t-1$. When going from $t-1$ to t, there are four possibilities of transitions from a starting state $s_{t-1} = i$ to an arrival state $s_t = j$, each leading to its own mean vector $\boldsymbol{\mu}_{t|t}^{i(j)}$, covariance matrix $\boldsymbol{\Sigma}_{t|t}^{i(j)}$ and marginal likelihood $\mathcal{L}_{t|t}^{i(j)}$. In the collapse step, the prior probability of each origin state is combined with the transition probability and the marginal likelihood of each transition using Equation 3.15. The end result of the collapse step is a mean vector, covariance matrix, and a probability for each model. Although SKF is well documented in the field of machine learning, it has never been coupled with BDLM and it lacks a generic model architecture for detecting anomalies in the field of SHM.

3.4 Generic Model Architecture for Anomaly Detection

This section proposes a generic model architecture for anomaly detection in BDLMs. As mentioned in §1.2, an anomaly is defined as a change in the underlying process (e.g., baseline behaviour) in time series. If the speed and acceleration of the underlying process can be modeled over time, the anomaly detection task is done by tracking the changes in these time-varying quantities. Figure 3.5 presents a baseline behaviour with a descending trend, its speed, and its acceleration over time. The normal and abnormal states are denoted as (1) and (2). The presence of the abnormal state (2) in the baseline behaviour from the time t_s to the time t_e, causes the changes in its speed and acceleration in the corresponding period. After the time t_e, the baseline behaviour returns to its normal state (1) where the speed and acceleration show the same behaviour as before the switching state at the time t_s.

In BDLMs, the baseline behaviour is represented by the baseline component. Its speed and acceleration are described by the local trend and local acceleration. Therefore, the key parts of the anomaly detection methodology for the BDLMs lie in the model architecture of these components employed for each state, as well as in the transition probability matrix, $\mathbf{Z}_t$, describing the switch between states. For the model architecture, each state has its own transition matrix $\mathbf{A}_t$ and model error covariance matrix $\mathbf{Q}_t$. According to the SKF theory in Section 3.3, the parameters from the matrix $\mathbf{Q}_t^{i(j)}$ need to be identified both for the stationary case, that is, $i = j$, and for the case of a state transition, that is, $i \neq j$. A state transition is here defined by a change in speed and acceleration in the baseline behaviour. Other hidden state variables describing the dependence on environmental conditions such as temperature remain unchanged. In the presence of a state transition, the model architecture must allow for an increase in the uncertainty for the local trend and local acceleration. For this purpose, the standard deviations included in $\mathbf{Q}_t^{i(j),\,\text{baseline}}$ are treated as unknown parameters to be inferred from observations.

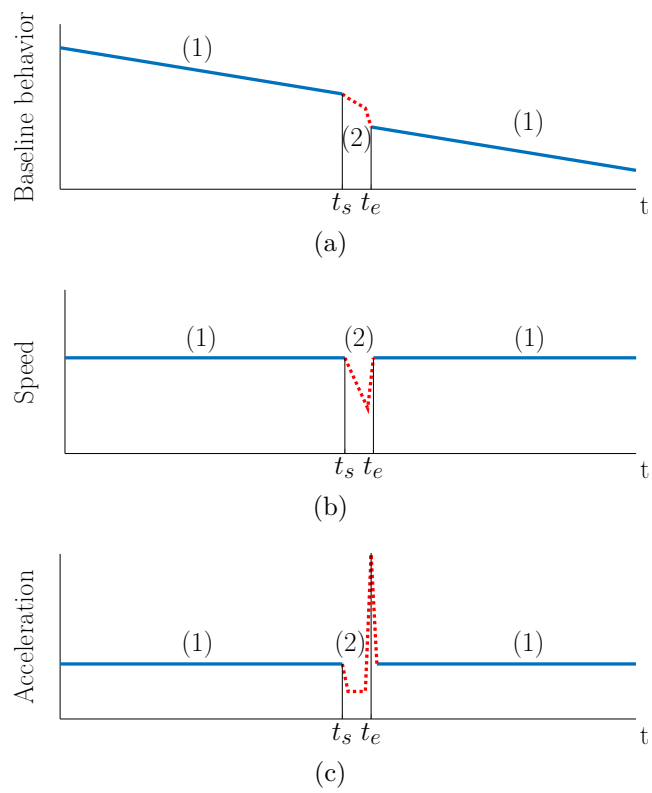


Figure 3.5 Illustration of the anomaly detection for the baseline behaviour. (1) and (2) represents the normal and abnormal states of the baseline behaviour.

For common cases, we assume that there are only two states: *normal* and *abnormal* where the anomaly is represented by the abnormal state. The transition matrix and model covariance matrix corresponding to the normal (denoted as 1) and abnormal (denoted as 2) states are written as

$$\mathbf{A}_t^1 = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{Q}_t^{1(1)} = (\sigma_w^{\mathtt{LT}})^2 \cdot \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{Q}_t^{2(1)} = (\sigma_w^{\mathtt{LTT}})^2 \cdot \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{A}_t^1 = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q}_t^{2(2)} = (\sigma_w^{\mathtt{LA}})^2 \cdot \begin{bmatrix} \frac{\Delta t^2}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}, \quad \mathbf{Q}_t^{1(2)} = (\sigma_w^{\mathtt{LAT}})^2 \cdot \begin{bmatrix} \frac{\Delta t^2}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}, \tag{3.20}$$

where $\sigma_w^{\mathtt{LT}}$ is the local trend standard deviation, $\sigma_w^{\mathtt{LTT}}$ is the local-trend transition standard deviation, $\sigma_w^{\mathtt{LA}}$ is the local acceleration standard deviation, $\sigma_w^{\mathtt{LAT}}$ is the local-acceleration

transition standard deviation, and Δt is the time-step length. The mathematical development for obtaining the covariance matrix $\mathbf{Q}_t$ is described by Bar-Shalom et al. [87] and Labbe [91]. The transition probability matrix, $\mathbf{Z}_t$, is defined following

$$\mathbf{Z}_t = \begin{bmatrix} Z^{11} & 1 - Z^{11} \\ 1 - Z^{22} & Z^{22} \end{bmatrix}, \tag{3.21}$$

where $Z^{ij} = p(s_t = j | s_{t-1} = i), \forall i, j = 1, 2$ is the transition probability from a state i at the time $t-1$ to a state j at the time t. The vector of unknown model parameters, $\boldsymbol{\theta} = [\sigma_w^{\mathtt{LT}}\ \sigma_w^{\mathtt{LA}}\ \sigma_w^{\mathtt{LTT}}\ \sigma_w^{\mathtt{LAT}}\ Z^{11}\ Z^{22}]^\intercal$, needs to be learned from data.

3.5 Hidden Dynamic Regression

This section proposes a methodology that builds on the formulation of the dynamic regression (§3.2.2) in order to handle hidden non-harmonic periodic covariates, as shown in Figure 3.2b. Note that an unobserved external effect is defined as a *hidden covariate*. For this purpose, the block component matrices are defined as

$$\mathbf{A}_t^{\mathtt{HDR}} = 1,\ \mathbf{C}_t^{\mathtt{HDR}} = h(\mathcal{C}, t),\ \mathbf{Q}_t^{\mathtt{HDR}} = (\sigma^{\mathtt{HDR}})^2, \tag{3.22}$$

where observation matrix $\mathbf{C}_t^{\mathtt{HDR}}$ is replaced by the hidden response function $h(\mathcal{C}, t)$ and $\mathcal{C}$ is a set of control points being used for describing the shape of a pattern. In the context of BDLMs, a *control point* is defined by a time stamp and a value. The function $h(\mathcal{C}, t)$ consists in a cubic spline [92] capable of interpolating hidden covariate values at any time stamps.

Figure 3.6 presents an example of a *hidden response* function $h(\mathcal{C}, t)$. $h(\mathcal{C}, t)$ depends on the time stamp t as well as on a set of $\mathtt{N}$ *master control points*, $\mathcal{C} = \{(t_i, h_i), \forall i = 1 : \mathtt{N}\}$, where $h_i \in [-1, 1]$ is the *Normalized Hidden Covariate Value* (NHCV) and t_i is the time stamp corresponding to h_i. Note that the amplitude of the hidden covariates that influence the structural responses, is defined not by $h(\mathcal{C}, t)$ but by the dynamic regression coefficient $x_t^{\mathtt{DR}}$. The methodology can take advantage of the periodicity of the studied phenomenon to identify only (t_i, h_i) for master control points that are defined over the domain (1) having a duration corresponding to half a period, and (2) bounded at each end by symmetry planes. Note that if no symmetry planes exist, the same method applies except that the number of control points increases. The NHCVs h_{s1} and h_{s2} for time stamps corresponding to the symmetry planes, t_{s1} and t_{s2}, are fixed at either -1 or 1. Time stamps t_i for the master control points are uniformly spaced between t_{s1} and t_{s2}. Over one half-period before and after the symmetry planes, *slave control points* are defined to constrain the spline slope for the

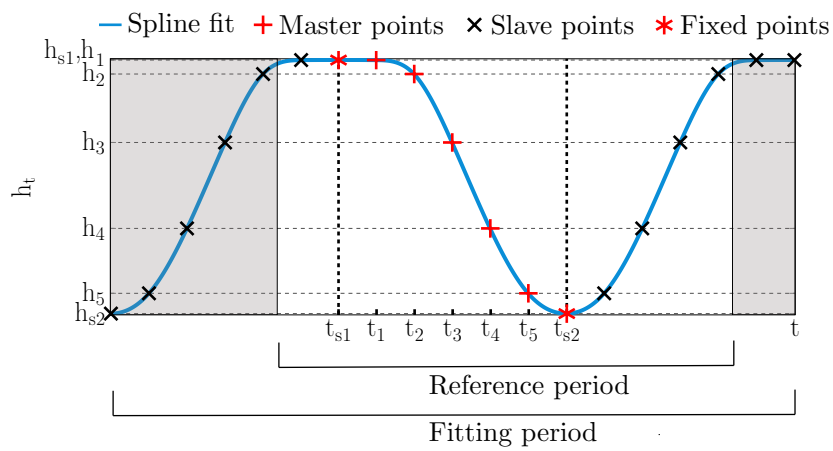


Figure 3.6 Example spline fitted using master and slave control points defined over three sub-segments that are separated by the vertical symmetry lines.

fixed points (t_{s1}, h_{s1}) and (t_{s2}, h_{s2}). Slave control points are replicates of the master points defined using the symmetry condition with respect to either t_{s1} or t_{s2}. Although the spline is fitted over the *fitting period* including the entire set of slave and master control points, only a portion having a length of one period is employed. This sub-selection is called *reference period*. An example of spline fitted using a set of five control points, $\mathcal{C} = \{(t_i, h_i), \forall i = 1 : 5\}$, is presented in Figure 3.6. Master control points are represented by plus signs (+), slave points by crosses (x), and fixed points by asterisks ($*$). Vertical dashed lines represent symmetry planes with respect to time. In practice, the NHCVs h_i are unknown and need to be estimated indirectly from observations of a structure's behaviour. Once NHCVs have been estimated using data, the hidden response $h(\mathcal{C}, t)$ is generalized for any time stamps t by extracting the spline value corresponding to any day of the year within the reference period.

Although this method has been shown to be capable of modeling non-harmonic periodic patterns, its main limitation is that complex periodic patterns typically require a large number of NHCVs making the approach computationally demanding. The reason behind this computational demand is that the NHCVs need to be estimated from the observations with an optimization algorithm such as MLE method.

3.6 Kernel Regression

Kernel regression methods have gained much attention in recent years, mainly due to the high performance that they provide in a variety of tasks [93, 94]. Kernel regression is a non-parametric approach that uses a known function, that is, the kernel function to fit nonlinear patterns in the data [95]. The use of kernel enables to interpolate between a set

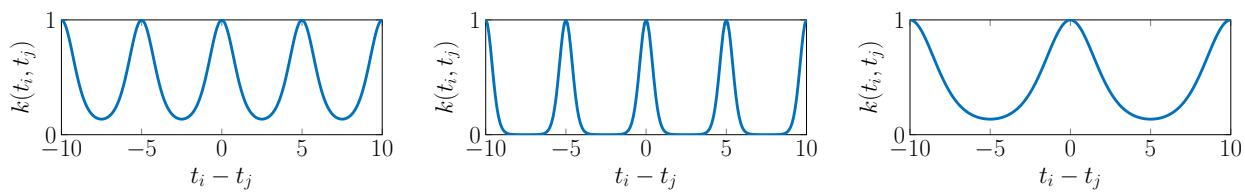
of control points as well as to extrapolate beyond them, which is well suited for handling non-uniformly spaced observations over time.

3.6.1 Periodic Kernel Regression

Kernel regression allows modeling periodic phenomena while overcoming the limitations of the methods presented in §3.2.2 and Section 3.5. The *kernel* is employed to measure the similarity between pairs of covariates. The goal here is to model the periodic phenomena in time series, therefore the periodic kernel [96] is formulated as

$$k(t_i, t_j) = \exp\left[-\frac{2}{\ell^2}\sin\left(\pi\frac{t_i - t_j}{\mathtt{P}}\right)^2\right], \tag{3.23}$$

where the covariate is the time t. The kernel output $k(t_i, t_j) \in (0, 1)$ measures the similarity between two time stamps t_i and t_j as a function of the distance between these, as well as a function of two parameters; the period and kernel lengthscale, $\boldsymbol{\theta} = \{\mathtt{P}, \ell\}$. Figure 3.7 presents three examples of periodic kernels for different sets of parameters $\boldsymbol{\theta}$.



(a) $\boldsymbol{\theta} = \{\mathtt{P} = 5, \ell = 1\}$ (b) $\boldsymbol{\theta} = \{\mathtt{P} = 5, \ell = 0.5\}$ (c) $\boldsymbol{\theta} = \{\mathtt{P} = 10, \ell = 1\}$

Figure 3.7 Examples of periodic kernels.

With kernel regression assuming that there is a set of observations, $\mathcal{D} = \{(t_i, y_i), \forall i = 1 : \mathtt{D}\}$, consisting in $\mathtt{D}$ pairs of observed system responses y_i, each associated with its time of occurrence t_i. The regression model is built using a set of control points defined using $\mathbf{t}^{\mathtt{KR}} \in \mathbb{R}^{\mathtt{N}}$, a vector of time covariates associated with a vector of control point's (CP's) values, $\mathbf{x} \in \mathbb{R}^{\mathtt{N}}$. The observed system responses are modeled following

$$y_i = \mathbf{x}^{\intercal}\frac{k(t_i, \mathbf{t}^{\mathtt{KR}})}{\sum_t k(t, \mathbf{t}^{\mathtt{KR}})} + v_i, \quad \underbrace{v : V \sim \mathcal{N}(v; 0, \sigma_v^2),}_{\text{Observation errors}} \tag{3.24}$$

where the predictive capacity of the model comes from the product of the CP's values $\mathbf{x}$ and the normalized kernel values which measure the similarity between the time t_i and those stored for the control points in $\mathbf{t}^{\mathtt{KR}}$. The main challenge here is to estimate $\mathbf{x}$ using the set of

observation $\mathcal{D}$. The likelihood describing the conditional probability of $\mathcal{D}$ given $\mathbf{x}$ is

$$f(\mathcal{D}|\mathbf{x}) = \prod_{i=1}^{\mathtt{D}} \mathcal{N}\left(y_i; \mathbf{x}^{\intercal} \frac{k(t_i, \mathbf{t}^{\mathtt{KR}})}{\sum_t k(t, \mathbf{t}^{\mathtt{KR}})}, \sigma_v^2\right). \tag{3.25}$$

If one employs a MLE approach to estimate the optimal values of the control points $\mathbf{x}^*$, the problem consists in maximizing the log-likelihood following

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \ \ln f(\mathcal{D}|\mathbf{x}). \tag{3.26}$$

Figure 3.8 presents an example of application of the kernel regression for modeling a non-harmonic periodic pattern. A set of $\mathtt{D} = 15$ simulated observations represented by plus signs (+) are generated by adding normal-distributed observation noise on the ground truth signal presented by the dashed line. Then, an optimization algorithm is employed to identify the optimal values $\mathbf{x}^*$ for a set $\mathtt{N} = 10$ control points represented by asterisks ($*$). This example

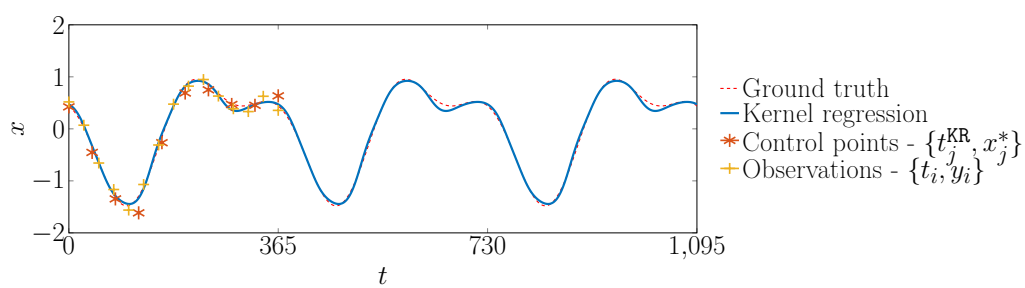


Figure 3.8 Examples of application of kernel regression.

illustrates the capacity of the kernel-based method for interpolating the system responses within control points as well as for extrapolating beyond them.

The main limitation of this approach is that the model cannot evolve over time. Moreover, like the dynamic regression method presented in §3.2.2, the estimation of $\mathbf{x}^*$ relies on a maximization algorithm which makes it computationally inefficient when applied to complex patterns.

3.6.2 Periodic Kernel Regression with BDLMs

This section proposes the new method which consists in coupling the kernel regression method with the hidden dynamic regression method presented in Section 3.5. For the new approach, we assume that there is a vector of $\mathtt{N}+1$ hidden state variables $\mathbf{x}_t^{\mathtt{KR}} = [x_{t,0}^{\mathtt{KR}}\ x_{t,1}^{\mathtt{KR}} \ldots x_{t,\mathtt{N}}^{\mathtt{KR}}]^{\intercal}$, where the first hidden state variable $x_{t,0}^{\mathtt{KR}}$, represents the kernel pattern, and the remaining

hidden state variables describe the CP's values. $\mathbf{x}_t^{\mathtt{KR}}$ is estimated using the filtering procedure presented in Section 3.2. The transition matrix is defined as

$$\mathbf{A}_t^{\mathtt{KR}} = \begin{bmatrix} \kappa & \tilde{\boldsymbol{k}}^{\mathtt{KR}}(t, \mathbf{t}^{\mathtt{KR}}) \\ \mathbf{0} & \mathbf{I}_{\mathtt{N}} \end{bmatrix}, \tag{3.27}$$

where $\kappa \in \{0, 1\}$ is the kernel regression coefficient and $\tilde{\boldsymbol{k}}^{\mathtt{KR}}(t, \mathbf{t}^{\mathtt{KR}})$ corresponds to the *normalized kernel*, $k(t, \mathbf{t}^{\mathtt{KR}}) / \sum_t k(t, \mathbf{t}^{\mathtt{KR}})$, as presented in Equation 3.23. $\tilde{\boldsymbol{k}}^{\mathtt{KR}}(t, \mathbf{t}^{\mathtt{KR}})$ is parameterized by the kernel lengthscale $\ell^{\mathtt{KR}}$, its period $p^{\mathtt{KR}}$, and a vector of $\mathtt{N}$ time stamps $\mathbf{t}^{\mathtt{KR}} = [t_1^{\mathtt{KR}} \dots t_{\mathtt{N}}^{\mathtt{KR}}]$, so that

$$\tilde{\boldsymbol{k}}^{\mathtt{KR}}(t, \mathbf{t}^{\mathtt{KR}}) \equiv \tilde{\boldsymbol{k}}^{\mathtt{KR}}(t, \mathbf{t}^{\mathtt{KR}}, \ell^{\mathtt{KR}}, p^{\mathtt{KR}}) \equiv \tilde{\boldsymbol{k}}_t^{\mathtt{KR}}, \tag{3.28}$$

where each time stamp $t_i^{\mathtt{KR}}, \forall i = 1 : \mathtt{N}$, corresponds to a hidden state variable $x_{t,i}^{\mathtt{KR}}$ associated with a CP's value. $x_{t,0}^{\mathtt{KR}}$ describes the kernel pattern value at time t which is obtained by multiplying the remaining $\mathtt{N}$ hidden state variables $[x_{t,1}^{\mathtt{KR}} \dots x_{t,\mathtt{N}}^{\mathtt{KR}}]^\intercal$ and the normalized kernel values $\tilde{\boldsymbol{k}}_t^{\mathtt{KR}}$. The coefficient κ describes the autocorrelation of the hidden state variables associated with the kernel pattern between the successive steps. Setting $\kappa = 0$ indicates that there is no autocorrelation between them and that the kernel regression models directly the periodic pattern. If this autocorrelation exists, κ is set to 1. In this case, the kernel regression models the stationary difference of the periodic pattern values from the previous to current time steps. In the context of this dissertation, κ is set to 0 for the case studies employed the kernel regression approach.

The $\mathtt{N} \times \mathtt{N}$ identity matrix forming the bottom right corner of $\mathbf{A}_t^{\mathtt{KR}}$ indicates that each of the hidden state variables $[x_{t,1}^{\mathtt{KR}} \dots x_{t,\mathtt{N}}^{\mathtt{KR}}]^\intercal$ evolves over time following a random walk model [79]. The temporal evolution of these hidden state variables is controlled by the process noise covariance matrix

$$\mathbf{Q}_t^{\mathtt{KR}} = \begin{bmatrix} (\sigma_{w,0}^{\mathtt{KR}})^2 & \mathbf{0} \\ \mathbf{0} & (\sigma_{w,1}^{\mathtt{KR}})^2 \cdot \mathbf{I}_{\mathtt{N}} \end{bmatrix}, \tag{3.29}$$

where $\sigma_{w,1}^{\mathtt{KR}}$ controls the increase in the variance of the hidden state variables associated with the CP's values between successive time steps, and $\sigma_{w,0}^{\mathtt{KR}}$ controls the time-independent process noise in the hidden state variable associated with the kernel pattern. This process noise allows describing random, unpredictable changes in the periodic phenomena between successive time steps. There are four possible cases:

1. $\sigma_{w,0}^{\mathtt{KR}} = 0,\ \sigma_{w,1}^{\mathtt{KR}} = 0$: the kernel pattern is stationary and the kernel regression can exactly model the true process.
2. $\sigma_{w,0}^{\mathtt{KR}} > 0,\ \sigma_{w,1}^{\mathtt{KR}} = 0$: the kernel pattern is stationary, yet the kernel regression is an

approximation of the true process being modeled.

3. $\sigma_{w,0}^{\texttt{KR}} = 0,\ \sigma_{w,1}^{\texttt{KR}} > 0$: the kernel pattern is non-stationary so the hidden state variables associated with the CP's values evolve over time, and the kernel regression can exactly model the true process.
4. $\sigma_{w,0}^{\texttt{KR}} > 0,\ \sigma_{w,1}^{\texttt{KR}} > 0$: the kernel pattern is non-stationary and the kernel regression cannot exactly model the true process.

As mentioned, above only the hidden state variable associated with the kernel pattern contributes directly to the observation so the observation matrix is

$$\mathbf{C}_t^{\texttt{KR}} = [1\ 0 \ldots 0]. \tag{3.30}$$

All CP's values are considered as hidden state variables, thus the only parameters to be estimated using the optimization algorithm are $\boldsymbol{\theta}^{\texttt{KR}} = [\sigma_{w,0}^{\texttt{KR}}\ \sigma_{w,1}^{\texttt{KR}}\ \ell^{\texttt{KR}}\ \texttt{P}^{\texttt{KR}}]^{\intercal}$.

An example using kernel regression with BDLMs on a non-harmonic periodic pattern is presented in Figure 3.9, where ground truth is presented by the dashed line. The control points and kernel pattern are presented by the asterisk ($*$) and circle ($\circ$). Figure 3.9a illustrates a target periodic pattern with a period of 365 days and a time step of 1 day. This example employs a vector of 11 hidden state variables associated with the kernel pattern and CP's values for modeling the periodic pattern. The new approach allows estimating the CP's values over time, thus each time step has its own CP's values that are employed for constructing its kernel pattern value (see Figure 3.9b). For an intuitive visualization, only 4 out of 365 time steps are shown in this figure. Figure 3.9c illustrates the left orthographic projection of the CP's values presented in Figure 3.9b. The kernel pattern value at a given time stamp t_i, is estimated using its CP's values and the periodic kernel formulation in Equation 3.23. Figure 3.9d shows 4 estimated values of the kernel pattern corresponding to 4 time steps in Figure 3.9b. Figure 3.9d illustrates a superposition of the complete kernel pattern and ground truth. The standard deviation of the hidden state variables associated with the CP's values $\sigma_1^{\texttt{KR}}$, is set to zero, thus these 10 hidden state variables are constant over time (Figure 3.9b). Note that Figure 3.9 only shows the mean values of the hidden state variables associated with the kernel pattern and CP's values over time.

3.7 Applications

This section presents applications of the methods proposed for modeling periodic external effects as well as for detecting anomalies for several datasets collected on different structures

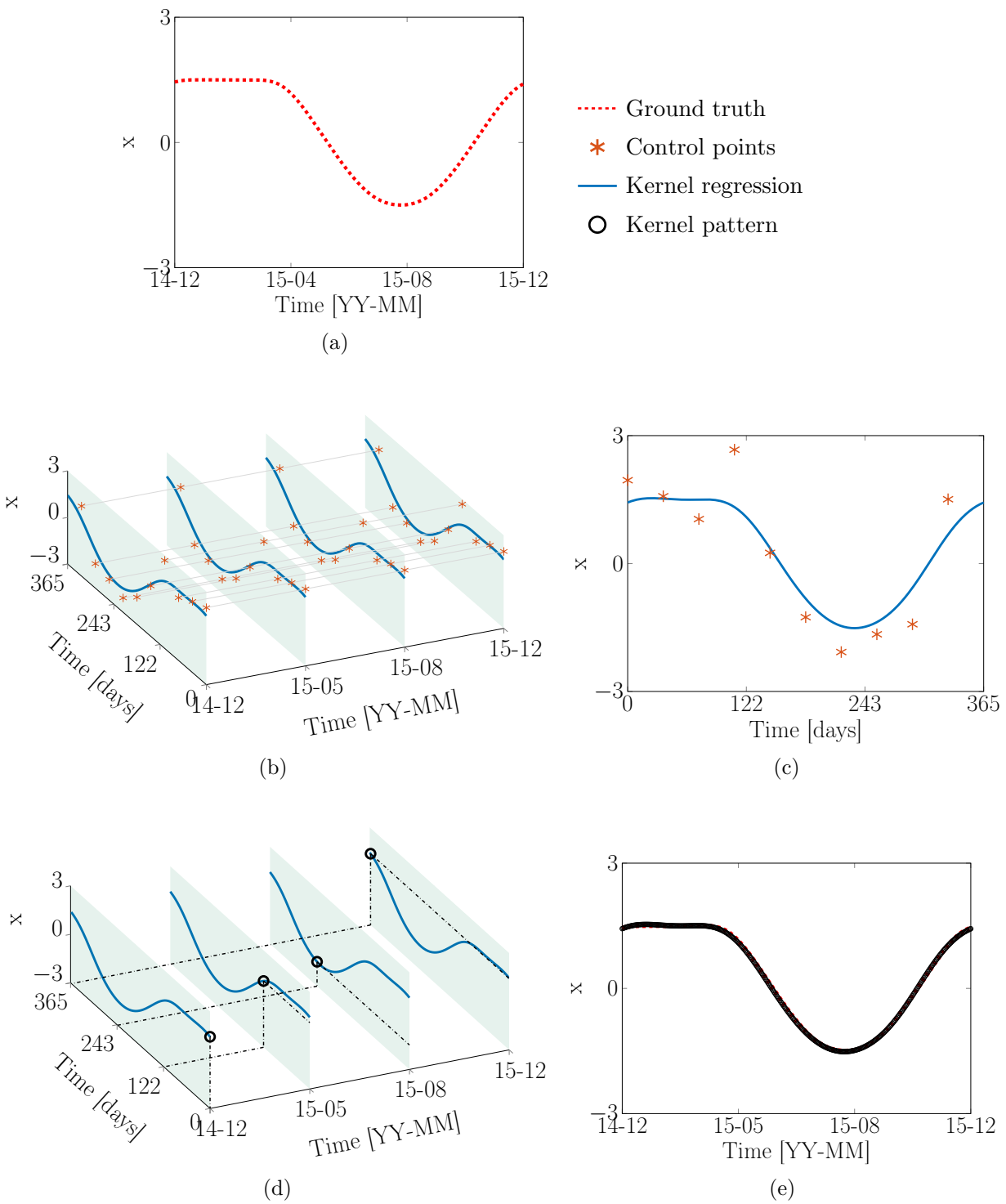


Figure 3.9 An example of using Kernel regression with BDLMs. (a) Periodic pattern with a period of 365 days; (b) Evolution of 10 hidden state variables associated with the control point's values for 4 time steps; (c) Left orthographic projection of 10 hidden state variables associated with the control point's values; (d) Estimated values of kernel pattern for 4 time steps; (e) Kernel pattern.

such as dams and bridges.

3.7.1 Comparison of Approaches for Modeling a Simple Non-Harmonic Pattern

The objective of this case study is to compare the kernel regression (§3.6.2), hidden dynamic regression (Section 3.5), and Fourier form (§3.2.2) approaches for modeling external effects exhibiting a simple non-harmonic, yet periodic pattern.

Data Description

The horizontal dam displacement data on a dam is measured using an inverted pendulum over a period of five years with a total of 1721 data points. The sensor studied is located on the west bank of the dam as shown in Figure 3.10. The engineers responsible for the dam

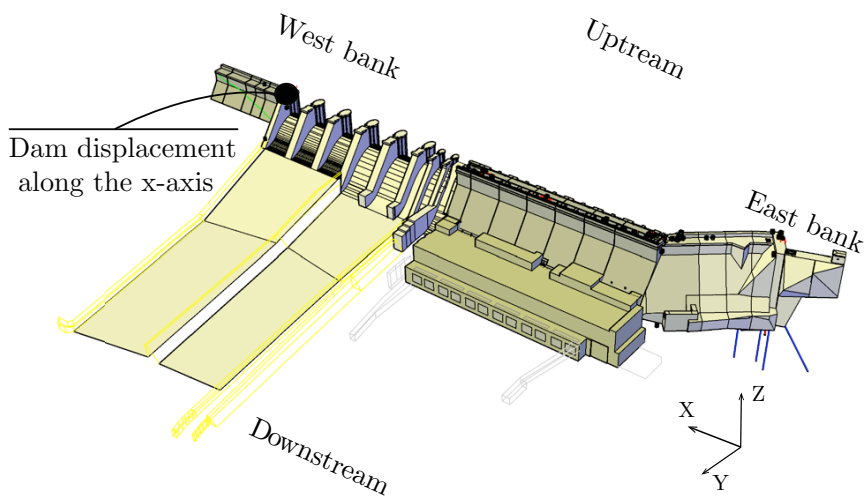


Figure 3.10 Location plan of the sensor employed to monitor the dam behaviour.

instrumentation have estimated the observation error standard deviation to be $\sigma_v = 0.3\,\text{mm}$. Figure 3.11 shows the complete dataset. In addition to the linear trend, the data displays a yearly seasonal pattern where displacements are maximal during winter months and minimal during summer. The key aspect here is that the seasonal effect is non-harmonic, yet periodic; the evolution of displacement during the winter is more stable than during the summer. The hypothesis for this behaviour is that the structure's response depends not only on the air temperature but also on the water temperature that is known to follow a non-harmonic periodic pattern where in winter months, the temperature stabilizes despite the air temperature dropping below -20°C. To measure model predictions, the dataset are divided into a *training*

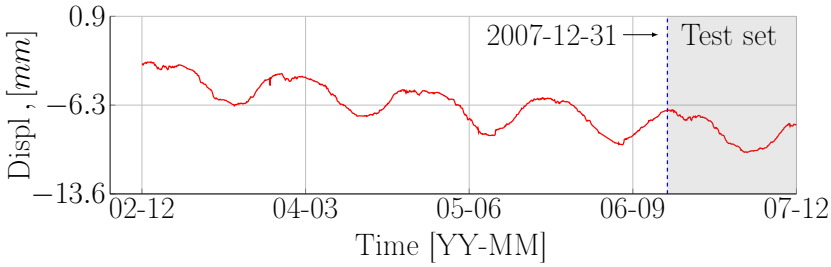


Figure 3.11 Illustration of the raw displacement dataset.

set of 4 years (1383 data points) and a *test set* of 1 year (338 data points). The data in the test set is used to measure how well the model performs at making forecasts in this test set.

Architecture for Model #1, #2, and #3

This case study compares three models using the kernel regression (#1), hidden dynamic regression (#2), and Fourier form (#3) approaches for describing the non-harmonic periodic pattern. In model #1, the observations are decomposed into a vector of hidden state variables including a baseline component (B), a local trend (LT), a kernel regression (KR) component with a period of 365.24 days, and an autoregressive (AR) component. The baseline component and local trend are used to model the behaviour of the displacement over time. The kernel regression is employed to describe the external effect. The autoregressive component is used to capture the time-dependent model errors. The vector of hidden state variables is defined following

$$\mathbf{x}_t = \left[x^{\mathtt{B}}\ x^{\mathtt{LT}}\ x_0^{\mathtt{KR}}\ x_1^{\mathtt{KR}} \ldots x_6^{\mathtt{KR}}\ x^{\mathtt{AR}}\right]_t^{\mathsf{T}}, \tag{3.31}$$

where $x_{t,0}^{\mathtt{KR}}$ is the kernel pattern and $x_{t,1:6}^{\mathtt{KR}}$ are associated with 6 CP's values. Instead of using the kernel regression component, the vector of hidden state variables for Model #2 employs the hidden-dynamic-regression component (HDR) with 5 control points for modeling the external effect in which the reference period of hidden covariates is 365.24 days and the two fixed points corresponding to symmetry planes are located at the 50th end 232.5^{th} day. The vector of hidden state variables for Model #2 is written as

$$\mathbf{x}_t = \left[x^{\mathtt{B}}\ x^{\mathtt{LT}}\ x^{\mathtt{HDR}}\ x^{\mathtt{AR}}\right]_t^{\mathsf{T}}. \tag{3.32}$$

In Model #3, the vector of hidden state variables is the same as the models #1 and #2, except that the external effect is modeled by a superposition of two Fourier-form components with a period of $\mathtt{P}_1 = 365.24$ days and $\mathtt{P}_2 = 182.62$ days. The vector of hidden state variables

for model #3 is given by

$$\mathbf{x}_t = \left[x^{\mathtt{B}}\ x^{\mathtt{LT}}\ x^{\mathtt{F_1P_1}}\ x^{\mathtt{F_2P_1}}\ x^{\mathtt{F_1P_2}}\ x^{\mathtt{F_2P_2}}\ x^{\mathtt{AR}}\right]_t^{\mathsf{T}}. \tag{3.33}$$

The vector of unknown model parameters for three models are defined as

$$\begin{aligned} \text{Model \#1} \quad \boldsymbol{\theta}^{M_1} &= [\ell^{\mathtt{KR}}\ \phi^{\mathtt{AR}}\ \sigma^{\mathtt{AR}}]^{\mathsf{T}} \\ \text{Model \#2} \quad \boldsymbol{\theta}^{M_2} &= [h_1^{\mathtt{HDR}}\ \ldots\ h_5^{\mathtt{HDR}}\ \phi^{\mathtt{AR}}\ \sigma^{\mathtt{AR}}]^{\mathsf{T}} \\ \text{Model \#3} \quad \boldsymbol{\theta}^{M_3} &= [\phi^{\mathtt{AR}}\ \sigma^{\mathtt{AR}}]^{\mathsf{T}}, \end{aligned} \tag{3.34}$$

where $\ell^{\mathtt{KR}}$ is the kernel length, $\phi^{\mathtt{AR}}$ is the autoregression coefficient, $\sigma^{\mathtt{AR}}$ is the autoregression standard deviation, and $h_i^{\mathtt{HDR}}, \forall i \in \{1, \cdots, 5\}$ is the normalized hidden covariate value. The standard deviation, kernel lengthscale, and kernel period are positive real numbers $\mathbb{R}^+$. The autoregression coefficient is constrained to the interval $[0, 1]$ because the autoregressive component is assumed to be stationary [97]. As presented in Section 3.5, the normalized hidden covariate value varies in range from -1 to 1, that is, $h_i^{\mathtt{HDR}} \in [-1, 1], \forall i \in \{1, \cdots, 5\}$. For an efficient optimization, the model parameters are transformed in the unbounded space [98]. The natural logarithm function is applied to the standard deviation, kernel lengthscale, and kernel period. The logistic sigmoid function is employed for the autoregression coefficient. The details of these transformation functions are presented in Section 4.5. The complete model matrices $\{\mathbf{A}_t, \mathbf{C}_t, \mathbf{Q}_t, \mathbf{R}_t\}$ for three models are presented in Appendices A.1, A.2, and A.3. $\boldsymbol{\theta}^{M_1}, \boldsymbol{\theta}^{M_2}$, and $\boldsymbol{\theta}^{M_3}$ are learned from data using MLE approach presented in Section 4.3. The initial parameter values for the vectors of model parameters in the original space are tuned using engineering heuristics as well as prior data analysis such that

$$\begin{aligned} \boldsymbol{\theta}_0^{M_1} &= [0.5\ 0.95\ 0.16]^{\mathsf{T}} \\ \boldsymbol{\theta}_0^{M_2} &= [-0.95\ -0.8\ -0.3\ 0.3\ 0.7\ 0.95\ 0.16]^{\mathsf{T}} \\ \boldsymbol{\theta}_0^{M_3} &= [0.95\ 0.16]^{\mathsf{T}}, \end{aligned} \tag{3.35}$$

where the ordering of each model parameter is the same as in Equation 3.34.

Comparative Results

The optimal vector of model parameters for each model is

$$\begin{aligned}
\boldsymbol{\theta}^{M_1,*} &= [0.582\ 0.991\ 0.02]^\intercal \\
\boldsymbol{\theta}^{M_2,*} &= [-1\ \ -0.906\ \ -0.443\ 0.278\ 0.782\ 0.992\ 0.03]^\intercal \\
\boldsymbol{\theta}^{M_3,*} &= [0.99\ 0.03]^\intercal\,,
\end{aligned} \tag{3.36}$$

where again the ordering of each model parameter remains identical as in Equation 3.34. Figure 3.12 presents the displacement forecast of three models in the test set. The solid line

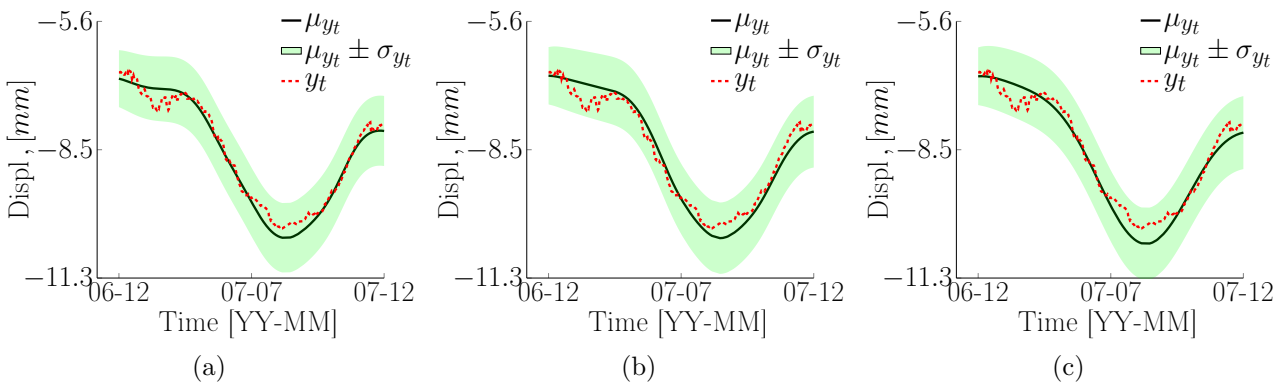


Figure 3.12 Illustration of displacement forecasts in the test set. (a) Model #1: Kernel regression; (b) Model #2: hidden dynamic regression; (c) Model #3: Fourier form.

presents the predicted mean values of displacement μ_{y_t}. The uncertainty bounds $\mu_{y_t} \pm \sigma_{y_t}$ are presented by the shaded regions. Note that the uncertainty includes the errors from both the model and observations. The test data are presented by the dashed line. A general remark is that the uncertainty bounds obtained from three models include the observations in the test set. The *Mean Absolute Error* (MAE), *Root Mean Squared Error* [99], and *Log Predictive Density* (LPD) [100] evaluated on the test dataset are employed to measure the forecast accuracy. The mathematical formulations for these metrics are detailed in Appendix C. Table 3.1 presents the metric values evaluated on the test set as well as the training time for each model. Model #1 using the kernel regression outperforms Model #2 and Model #3 that employs the hidden dynamic regression and Fourier form. More specifically, it yields more accurate forecast values while requiring less training time than the other models. The training time of Model #2 is approximated to 10 times higher than Model #1 and Model #3. This is because it has a larger number of model parameters than the others.

Table 3.1 Comparison of three approaches of the forecast capacity for modeling an external effect. Model #1: kernel regression; Model #2: hidden dynamic regression; Model #3: Fourier form; RMSE: root-mean-square error; MAE: mean absolute error; LPD: Log Predictive Density.

Model	Test set (338 data points)			Training time (s)
	RMSE	MAE	LPD	
#1	**0.18**	**0.14**	**−8.26**	**30**
#2	0.23	0.20	−38.91	280
#3	0.23	0.19	−44.17	37

3.7.2 Modeling a Complex Non-Harmonic Periodic Pattern

The goal of this case study illustrates the potential of the kernel regression approach presented in §3.6.2 for modeling a complex non-harmonic periodic pattern. For such a pattern, the Fourier form approach struggles to identify the required number of periodic components while the hidden-dynamic-regression method is more computationally demanding regarding the optimization of model parameters.

Data Description

The traffic loading data are collected on the Tamar Bridge from September 01 to October 21, 2007 with a total of 2409 data points [101]. Figure 3.13 presents the entire dataset. The traffic loading data are recorded from a toll booth where the vehicles are counted and classified by weight classes. The data are collected with a uniform time step of 30 minutes and the units are kilotons (kTs). A constant baseline and a periodic pattern with a period of 7 days can be observed from the raw data. The traffic loading on weekends is much lighter than those on weekdays. For most of the day, the traffic load presents a high volume between 8 am and 4 pm, and it then drops after 20 pm. To examine the predictive performance, the dataset is divided into a *training set* (1649 data points) and a *test set* (760 data points). The unknown model parameters are learned using the training set and the predictive performance is then evaluated using the test set. The test set is presented by the shaded region in Figure 3.13.

Architecture for Model #4

Model #4 for interpreting the traffic load data consists in a vector of hidden state variables that includes a baseline (`B`) component for the average traffic load, a kernel regression (`KR`) component with 101 hidden state variables to describe the periodic pattern, and an

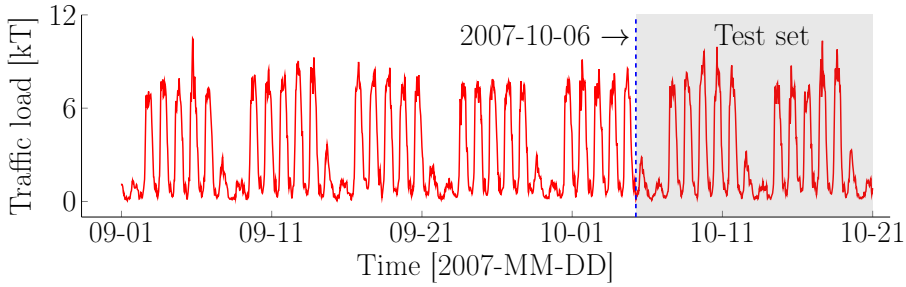


Figure 3.13 Traffic load on the Tamar Bridge in the United Kingdom.

autoregressive (AR) component to capture the time-dependent model errors. The vector of hidden states is given by

$$\mathbf{x}_t \;=\; [x^{\mathtt{B}}\; x_0^{\mathtt{KR}}\; x_1^{\mathtt{KR}} \ldots x_{100}^{\mathtt{KR}}\; x^{\mathtt{AR}}]_t^{\mathsf{T}}\,. \tag{3.37}$$

Model #4 involves a vector of unknown model parameters $\boldsymbol{\theta}$ that are defined as

$$\boldsymbol{\theta} = \left[\sigma_w^{\mathtt{B}}\; \sigma_{w,0}^{\mathtt{KR}}\; \sigma_{w,1}^{\mathtt{KR}}\; \ell^{\mathtt{KR}}\; \mathtt{P}^{\mathtt{KR}}\; \phi^{\mathtt{AR}}\; \sigma_w^{\mathtt{AR}}\; \sigma_v\right]^{\mathsf{T}}, \tag{3.38}$$

where $\sigma_w^{\mathtt{B}} \in (0, +\infty)$ is the baseline standard deviation, $\sigma_{w,0}^{\mathtt{KR}} \in (0, +\infty)$ is the standard deviation of the hidden state variable associated with the kernel pattern, $\sigma_{w,1}^{\mathtt{KR}} \in (0, +\infty)$ is the standard deviation for the hidden state variables associated with the control point's values, $\ell^{\mathtt{KR}} \in (0, +\infty)$ is the kernel lengthscale, $\mathtt{P}^{\mathtt{KR}} \in (0, +\infty)$ is the kernel period, $\phi^{\mathtt{AR}} \in [0, 1]$ is the autoregression coefficient, $\sigma^{\mathtt{AR}}$ is the autoregression standard deviation, and $\sigma_v \in (0, +\infty)$ is the observation noise standard deviation. For transforming the model parameters to an unbounded space, the natural logarithm is applied to the standard deviations, kernel lengthscale, and kernel period. The logistic sigmoid function is used for the autoregression coefficient. The complete model matrices are detailed in Appendix A.4. The initial parameter values in the original space are selected using expert judgment and experience as well as prior data analysis,

$$\boldsymbol{\theta}_0 = \left[\underbrace{10^{-6}}_{\sigma_w^{\mathtt{B}}}\; \underbrace{0.29}_{\sigma_{w,0}^{\mathtt{KR}}}\; \underbrace{0.029}_{\sigma_{w,1}^{\mathtt{KR}}}\; \underbrace{0.5}_{\ell^{\mathtt{KR}}}\; \underbrace{7}_{\mathtt{P}^{\mathtt{KR}}}\; \underbrace{0.75}_{\phi^{\mathtt{AR}}}\; \underbrace{0.075}_{\sigma_w^{\mathtt{AR}}}\; \underbrace{1.47}_{\sigma_v}\right]^{\mathsf{T}}. \tag{3.39}$$

Parameter & Hidden State Estimation

The optimal model parameters obtained using the MLE technique presented in Section 4.3 are

$$\boldsymbol{\theta}^* = \left[\underbrace{10^{-6}}_{\sigma_w^{\mathtt{B}}} \; \underbrace{3.3 \times 10^{-5}}_{\sigma_{w,0}^{\mathtt{KR}}} \; \underbrace{10^{-5}}_{\sigma_{w,1}^{\mathtt{KR}}} \; \underbrace{0.0511}_{\ell^{\mathtt{KR}}} \; \underbrace{7}_{\mathtt{p}^{\mathtt{KR}}} \; \underbrace{0.78}_{\phi^{\mathtt{AR}}} \; \underbrace{0.34}_{\sigma_w^{\mathtt{AR}}} \; \underbrace{1.2 \times 10^{-5}}_{\sigma_v} \right]^{\mathsf{T}} . \tag{3.40}$$

It is noted that $\sigma_{w,0}^{\mathtt{KR}}$ and $\sigma_{w,1}^{\mathtt{KR}}$ are close to zero, thus the kernel pattern for this case study is stationary. Figure 3.14 presents the hidden state variables and predicted means for the traffic load estimated using the Kalman smoother [28] for both the training and test set. $\mu_{t|t}$ and $\sigma_{t|t}$ are the mean value and standard deviation at time t for the hidden state variables. The mean value $\mu_{t|t}$ and its uncertainty bound $\mu_{t|t} \pm \sigma_{t|t}$, are presented by the solid line and shaded region. The estimates for the training set and test set are delimited by the dashed line. The traffic load data is presented by the dash-dotted line. Figure 3.14c shows that the autoregressive component, $x_t^{\mathtt{AR}}$ is stationary. If it would not be the case, the non-stationarity would indicate that either the component choice, or the optimal parameter identified are inadequate. Figure 3.14b shows that the kernel pattern is stationary and all cycles of 7 days have an identical pattern. Figure 3.14d displays that the estimates of the traffic load in the test set are close to the corresponding observations. It can be seen that the model is not capable of predicting the peaks on the test set where the traffic loading presents a high volume during the rush hours. This can be explained by the high variability associated with these rush hours. The proposed method is intended to capture the periodic phenomena and not the non-periodic changes occurring from day to day.

The MAE, RMSE, and LPD are employed to measure the forecast accuracy by comparing the estimates with their corresponding traffic-load data in the test set. The mathematical formulations for the MAE, RMSE, and LPD are presented in Appendix C. These metrics are evaluated for the forecast periods of 1, 3, 7, and 14 days in the test set. The results are summarized in Table 3.2. The MAE for each forecast period is small in comparison with the traffic-loading amplitude that varies in range from 0.07 to 10.3 kT. The uncertainty bounds typically widen as the forecast horizon increases, leading to a decrease in the LPD and an increase in the MAE as well as RMSE.

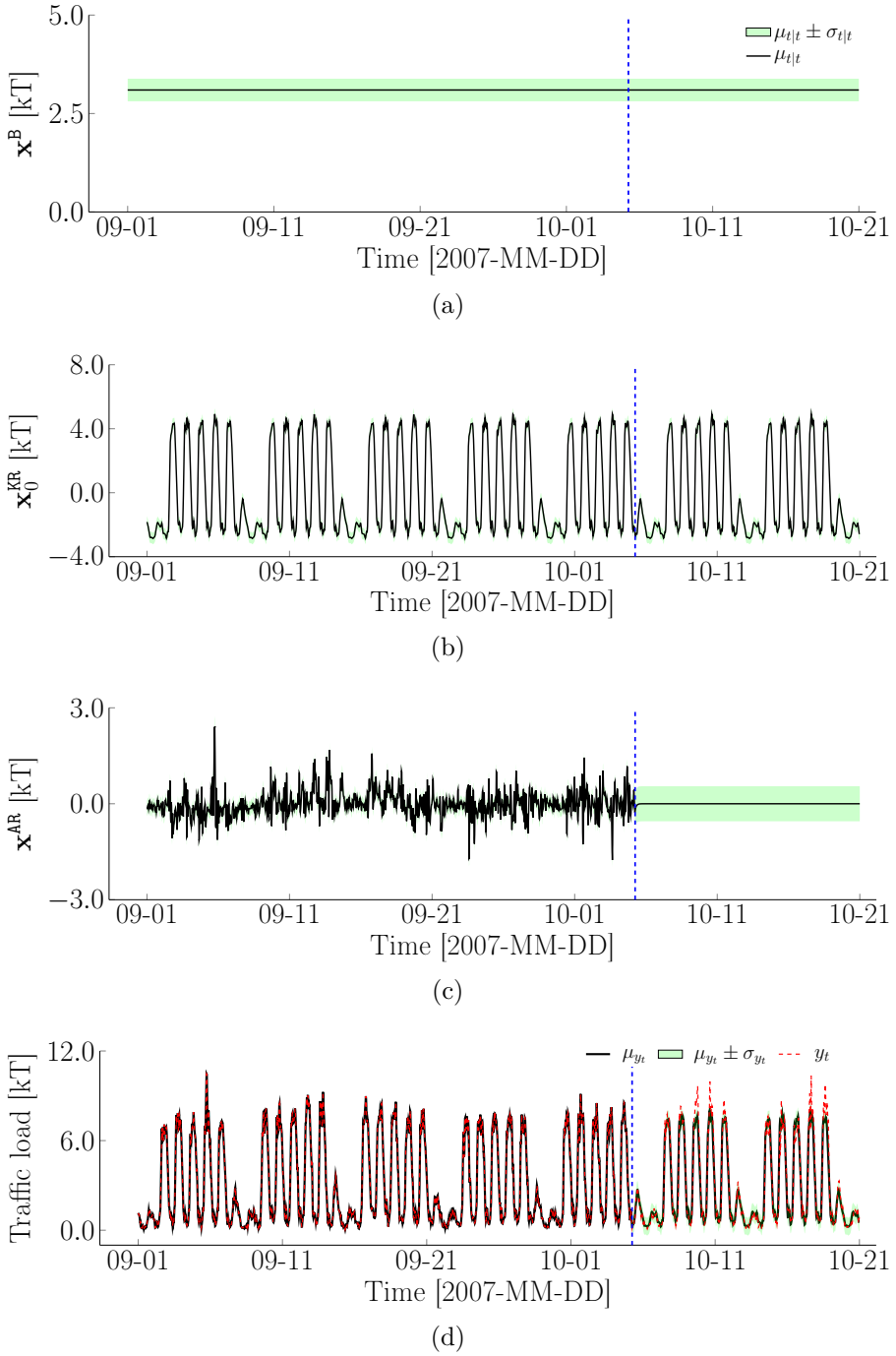


Figure 3.14 Illustration of the estimation of hidden state variables for the traffic load data: (a) Baseline component, $x_t^{\mathtt{B}}$; (b) Kernel pattern, $x_{t,0}^{\mathtt{KR}}$; (c) Autoregressive component, $x_t^{\mathtt{AR}}$; (d) Traffic load.

Table 3.2 Evaluation of forecast accuracy with respect to the different forecast periods for the traffic load data. RMSE: root-mean-square error; MAE: mean absolute error; LPD: log predictive density.

	Forecast period			
Metric	1 (day) 48 pts	3 (days) 144 pts	7 (days) 336 pts	14 (days) 760 pts
MAE	0.15	0.20	0.37	0.38
RMSE	0.21	0.28	0.55	0.58
LPD	−10.90	−61.68	−267.55	−656.30

3.7.3 Offline Anomaly Detection

This case study illustrates the potential of the method proposed for the anomaly detection method presented in Section 3.4 and applied on the horizontal displacement data collected on a Dam in Canada.

Data Description

Figure 3.15 presents the horizontal displacement data along the X-direction (see Figure 3.10) recorded over the period of 13 years and 1 month (8364 data points). The observation error

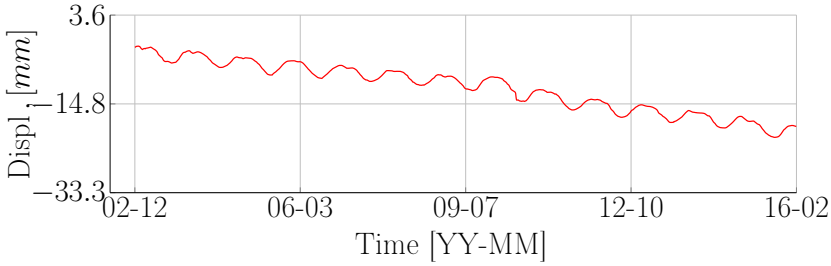


Figure 3.15 X-direction displacement data collected over the period of 13 years and 1 month.

standard deviation $\sigma_v = 0.3\,\text{mm}$ was provided by the instrumentation engineers. Based on the raw data, one can observe a linear trend and a seasonal pattern with a period of one year. The seasonal pattern reaches its maximum during winter and minimum during summer and is non-harmonic because of the lack of symmetry with respect to the horizontal

axis. This behaviour can be explained by the dependence of the displacement on the water temperature [35], as mentioned in §3.7.1. Figure 3.16 presents the time-step length for the entire dataset. Time-step length varies in range from 1 hour to 36 days in which the two most

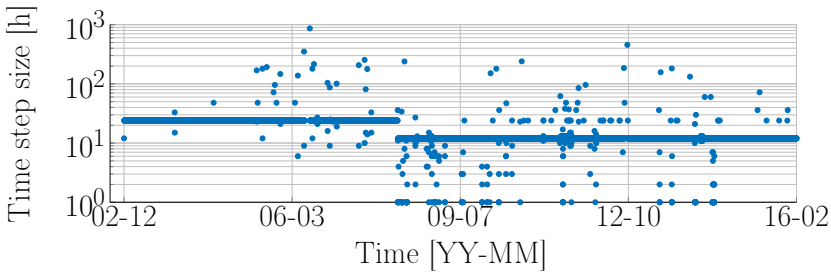


Figure 3.16 Time-step size is presented in a log scale.

frequent time steps are 12 and 24 hours. To adapt with the non-uniformity of time steps, the parameters need to be defined as a function of the time-step length where the *reference time step* is selected as the most frequent one [77].

Architecture for Model #5

Model #5 consists, for both case studies, in a vector of hidden state variables that includes a baseline (`B`) component, a local trend (`LT`), a local acceleration (`LA`), a kernel-regression (`KR`) component with a period of 365.24 days, and an autoregressive (`AR`) component. The structural behaviour over time is described by the baseline. The local trend is employed to model the rate of changes in the baseline component. The local acceleration is used to model the rate of changes in the local trend. The kernel regression component involves 11 hidden state variable and describes the non-harmonic periodic pattern. The autoregressive component is used to capture the time-dependent model errors. The vector of hidden state variables is written as

$$\mathbf{x}_t = \left[x^{\mathtt{B}}\ x^{\mathtt{LT}}\ x^{\mathtt{LA}}\ x_0^{\mathtt{KR}}\ x_1^{\mathtt{KR}} \ldots x_{10}^{\mathtt{KR}}\ x^{\mathtt{AR}}\right]_t^{\mathsf{T}}. \tag{3.41}$$

Because the main interest here is to detect anomalies in time series, two model classes representing respectively a state $s_t \in \{1\text{: } \textit{normal}, 2\text{: } \textit{abnormal}\}$ are built for Model #5. These two model classes use the same vector of hidden state variables presented in Equation 3.41, except that the local acceleration for the normal model-class is set to zero. This is done by assigning a value of zero to the line and row corresponding to the local acceleration component in the transition matrix $\mathbf{A}_t$ and model error covariance matrix $\mathbf{Q}_t$.

The transition probability matrix is

$$\mathbf{Z} = \begin{bmatrix} Z^{11} & Z^{12} \\ Z^{21} & Z^{22} \end{bmatrix}, \tag{3.42}$$

where $Z^{ij} = p(s_t = j | s_{t-1} = i)$ with $i, j = 1, 2$ is the prior probability of transitioning from a state i at the time $t-1$ to a state j at the time t. To be valid, this transition matrix must satisfy $\sum_j Z^{ij} = 1$ so that

$$\mathbf{Z} = \begin{bmatrix} Z^{11} & 1 - Z^{11} \\ 1 - Z^{22} & Z^{22} \end{bmatrix}. \tag{3.43}$$

Given this constraint, only transition probabilities Z^{ii} need to be defined as unknown parameters to be learned from data. The unknown model parameters are defined as

$$\boldsymbol{\theta} = \left[\ell^{\mathtt{KR}} \ \phi^{\mathtt{AR}} \ \sigma_w^{\mathtt{AR}} \ Z^{11} \ \sigma_w^{\mathtt{LA}} \ Z^{22} \right]^{\mathsf{T}}, \tag{3.44}$$

where $\ell^{\mathtt{KR}} \in (0, +\infty)$ is the kernel lengthscale, $\phi^{\mathtt{AR}} \in [0, 1]$ is the autoregression coefficient, $\sigma_w^{\mathtt{AR}} \in (0, +\infty)$ is the autoregression standard deviation, $Z^{ii} \in [0, 1]$ is the transition probability, and $\sigma_w^{\mathtt{LA}} \in (0, +\infty)$ is the local acceleration standard deviation. For the transformation function, the natural logarithm function is used for the standard deviations and kernel lengthscale. The logistic sigmoid function is employed for the autoregression coefficient and transition probabilities. The full model matrices employed in this case study is presented in Appendix A.5.

The model parameters are estimated using the MLE approach shown in Section 4.3. For this approach, using engineering knowledge for the definition of initial values for both model parameters and hidden state variables ensures an effective performance in terms of the capacity to detect anomalies and in terms of computational time for learning model parameters. For this purpose, the parameters related to the state transition such as Z^{11}, Z^{22}, and $\sigma^{\mathtt{LA}}$ need to be chosen with care. This case study assumes that the occurrence of an anomaly is rare so that the possibility of switching from a normal state to an abnormal state is lower than the probability of a switch in the opposite direction. Also, the uncertainty in the local acceleration $\sigma^{\mathtt{LA}}$ in the baseline behaviour for the abnormal model must be greater than for the normal model. Initial parameter values estimated based on the above-mentioned engineering heuristics are

$$\boldsymbol{\theta}^0 = \left[\underbrace{0.5}_{\ell^{\mathtt{KR}}} \ \underbrace{0.95}_{\phi^{\mathtt{AR}}} \ \underbrace{0.095}_{\sigma_w^{\mathtt{AR}}} \ \underbrace{0.9999}_{Z^{11}} \ \underbrace{4.8 \times 10^{-4}}_{\sigma_w^{\mathtt{LA}}} \ \underbrace{0.95}_{Z^{22}} \right]^{\mathsf{T}}. \tag{3.45}$$

The initial mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$ for hidden state variables are estimated using the multi-pass technique presented in §4.3.3 using a period of 5 years (1694 data points) with $\mathtt{N} = 5$ iterations. This period is selected because (i) it provides a sufficiently accurate estimation for the initial values and (ii) it requires less computational time than when using a longer period. The optimization procedure employs the entire dataset (8364 data points) for estimating the parameter values $\boldsymbol{\theta}^*$.

Results

The computational time required for parameter calibration is approximately 16 minutes for a computer with 32 Gb of Random Access Memory (RAM) and an Intel i7 processor. The optimal vector of model parameter values identified are

$$\boldsymbol{\theta}^* = \left[\underbrace{0.953}_{\ell^{\mathtt{KR}}} \; \underbrace{0.996}_{\phi^{\mathtt{AR}}} \; \underbrace{0.018}_{\sigma_w^{\mathtt{AR}}} \; \underbrace{1}_{Z^{11}} \; \underbrace{4.5 \times 10^{-3}}_{\sigma_w^{\mathtt{LA}}} \; \underbrace{0.997}_{Z^{22}} \right]^{\mathsf{T}} . \tag{3.46}$$

Combining BDLMs in Section 3.2 with SKF in Section 3.3 serves two purposes: (i) it enables the detection of anomalies without requiring labeled training data, and (ii) it decomposes the observations into its reversible and irreversible effects. Figure 3.17 presents the probabilities of the abnormal state estimated at each time step. The proposed method identifies that there

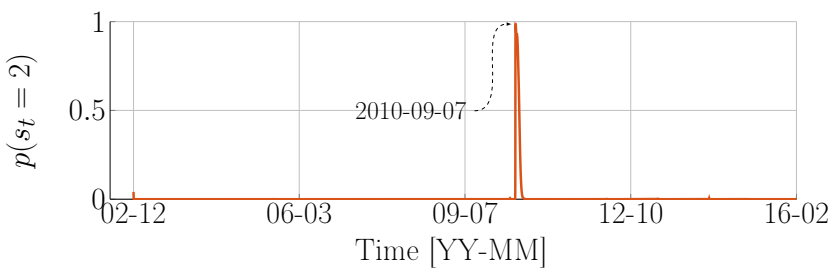
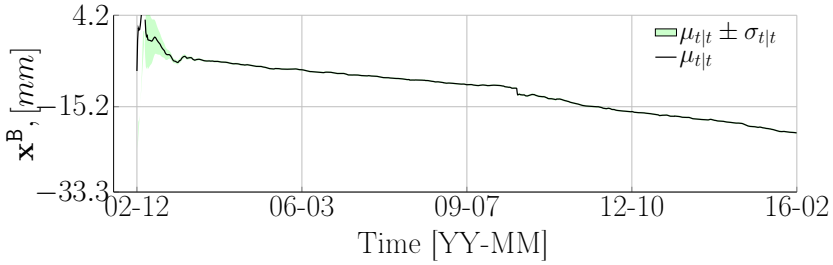
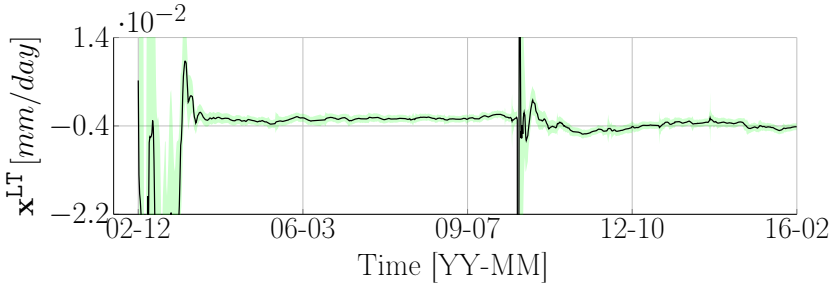


Figure 3.17 Probabilities of the two states are evaluated using SKF algorithm for the entire dataset.

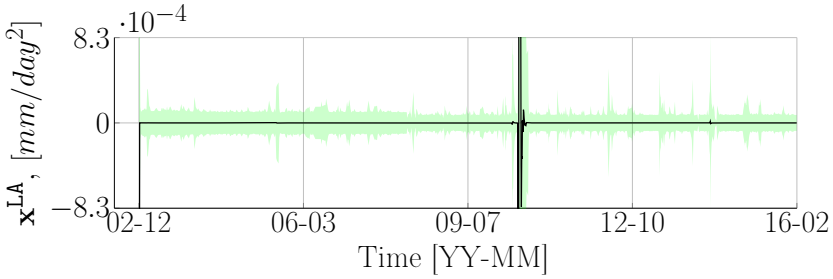
is an abnormal event occurring on July 09, 2010. This anomaly was caused by refection work that took place on the dam in early July. After the work was completed, the model identifies that the dam behaviour returns to a normal one. This example of application demonstrates how anomalies can be detected without triggering any false alarm that would jeopardize the applicability of the approach. Figure 3.18 presents the hidden state variables estimated for the entire dataset. The solid line represents the mean values $\mu_{t|t}$ and its uncertainty bounds $\mu_{t|t} \pm \sigma_{t|t}$ are represented by the shaded region. Figure 3.18a, b, and c show a sudden change



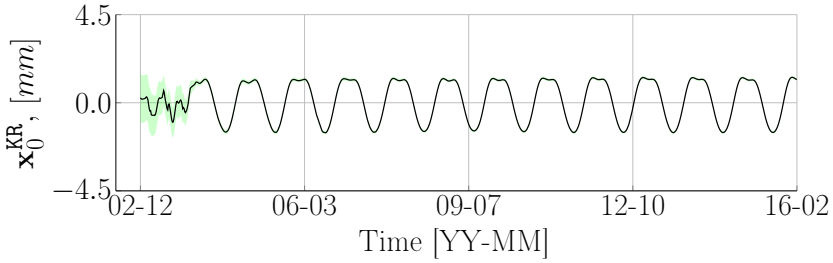
(a) Baseline component



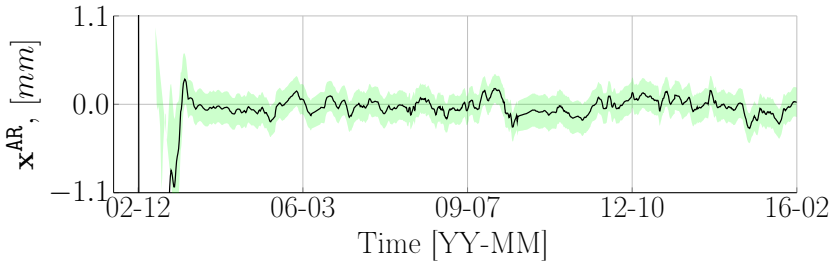
(b) Local trend



(c) Local acceleration



(d) kernel pattern



(e) Autoregressive component

Figure 3.18 Expected values $\mu_{t|t}$ and uncertainty bound $\mu_{t|t} \pm \sigma_{t|t}$ for hidden state variables of a combination of the normal and abnormal models are evaluated using SKF algorithm.

in the baseline, local trend and local acceleration at the moment when the anomaly occurred. These three figures show how the baseline behaviour of the structure can be isolated from the effect of external temperature. The external temperature is modeled by a kernel regression component in Figure 3.18d. Figure 3.18e shows that the autoregressive component follows a stationary process. Figures 3.18a, b, c, d, and e show that the large uncertainty in the hidden state variables during the initial period is due to the imperfect initial conditions. The uncertainty then vanishes as more and more data are observed.

3.8 Conclusion

This chapter proposes the methodologies (i) for modeling periodic external effects and (ii) for detecting anomalies in the structural responses. The hidden dynamic regression and kernel regression approaches are developed for the first task. Both methods have the ability to describe a non-harmonic, yet periodic pattern. The applications on the structural datasets show that the kernel regression method yields more accurate forecasts while requiring less training time than the hidden dynamic regression and Fourier form approaches. It is also capable of modeling complex non-harmonic periodic patterns. On the other hand, the hidden dynamic regression shows its ability to describe non-harmonic periodic patterns, yet it is much more computationally demanding than the kernel regression, especially when it comes to complex non-harmonic periodic patterns such as traffic load.

A common downside of both approaches is that a good initial guess for the pattern's period parameter is needed to ensure an efficient optimization. The reason is that the likelihood for the period parameter is strongly peaked so that starting from a wrong value can lead to a slow convergence. A prior data analysis that combines a periodogram study with a visualization of the time series can be used as a heuristic for identifying good starting values for the period prior analysis.

The key aspects of the anomaly detection methodology are that (i) it considers the prior probability of anomalies, the anomaly kinematic model and the probability to transition from a normal state to an abnormal state, (ii) does not require labeled training data with normal and abnormal conditions, and (iii) increases the robustness towards false alarms in real operation conditions. The method has shown to be capable to detect changes in the dam behaviour, hearby caused by the refection work. It also provided the specific information about the dam behaviour over time. The main limitations of the method proposed are that (i) model parameters are assumed to be constant over time, (ii) the influence of different anomaly types remains to be tested, and (iii) the estimation of model parameters can only be done offline. This last limitation will be addressed in Chapter 5.

CHAPTER 4 BATCH LEARNING

4.1 Introduction

In BDLMs, the vector of unknown model parameters $\boldsymbol{\theta}_t$ is employed to define the model matrices $\{\mathbf{A}_t, \mathbf{C}_t, \mathbf{Q}_t, \mathbf{R}_t, \mathbf{Z}_t\}$ presented in Chapter 3. The values taken by $\boldsymbol{\theta}_t$ are different from one dataset to another, so there is a need for an automated optimization procedure that allows learning $\boldsymbol{\theta}_t$ from the data itself. This chapter presents the adaptation of existing optimization algorithms available in the literature to BDLMs. In the field of machine learning, the task of learning model parameters can be regrouped in two common methods: *Maximum Likelihood Estimation* (MLE) and *Bayesian Estimation* (BE) [28]. The MLE consists in finding a single vector of model parameters that maximizes the log-likelihood function. Gradient-based optimization algorithms such as the *gradient ascent* and *stochastic gradient ascent* [58] can be employed for carrying out this task. Instead of only providing point estimates like the MLE approach, BE methods such as *Hamiltonian Monte Carlo* (HMC) and *Laplace Approximation* (LAP) allow approximating the posterior PDF of model parameters. In the context of this chapter, both MLE and BE assume that $\boldsymbol{\theta}_t$ is constant over time and is learned from a fixed-size training dataset, that is, $\boldsymbol{\theta}_t \equiv \boldsymbol{\theta}$. This learning technique is called *batch learning* [58]. The main contributions of this chapter are:

- adapting gradient-based optimization algorithms to BDLMs and comparing their performance (§4.3.5).
- adapting the HMC and LAP for quantifying the model parameter uncertainty to BDLMs (§4.4.4).
- proposing a method for propagating model parameter uncertainty to the hidden state variables in the context of BDLMs (§4.4.3).
- validating the proposed approaches on the simulated and real datasets (Section 4.6).

The chapter is organized as follows. Section 4.2 introduces the mathematical formulation of the marginal likelihood being used for BDLMs. Section 4.3 surveys existing gradient-based optimization algorithms and presents the framework proposed for integrating them into BDLMs. Section 4.4 provides the details for the Bayesian estimation methods. Section 4.5 presents the transformation functions being used to transform the model parameters from a bounded space to an unbounded one. Section 4.6 presents the applications of the proposed approaches to both the simulated and real datasets.

4.2 Likelihood

The *likelihood* is defined as the *joint prior probability of observations*, that is, the plausibility of the available observations $\mathbf{y}_{1:\mathtt{T}}$ given the model parameter vector $\boldsymbol{\theta}$. Assuming that the observations are conditionally independent from each other, the joint likelihood function is defined as the product of the marginal likelihoods,

$$\begin{aligned} p(\mathbf{y}_{1:\mathtt{T}}|\boldsymbol{\theta}) &= \prod_{t=1}^{\mathtt{T}} p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\boldsymbol{\theta}). \\ &= \prod_{t=1}^{\mathtt{T}}\prod_{j=1}^{\mathtt{S}}\prod_{i=1}^{\mathtt{S}} p(\mathbf{y}_t|s_t=j,s_{t-1}=i,\mathbf{y}_{1:t-1},\boldsymbol{\theta})\cdot p(s_t=j|s_{t-1}=i)\ldots \\ &\qquad \ldots\cdot p(s_{t-1}=i|\mathbf{y}_{1:t-1},\boldsymbol{\theta}), \\ &= \prod_{t=1}^{\mathtt{T}}\prod_{j=1}^{\mathtt{S}}\prod_{i=1}^{\mathtt{S}} \mathcal{L}_{t|t}^{i(j)}\cdot Z^{i(j)}\cdot \pi_{t-1|t-1}^{i}, \end{aligned} \tag{4.1}$$

where $s_t \in \{1,2,\ldots,\mathtt{S}\}$ is the Markov-switching variable presented in Section 3.3, $\mathcal{L}_{t|t}^{i(j)}$ is defined in Equation 3.10, $Z^{i(j)}$ is the transition probability, and $\pi_{t-1|t-1}^{i}$ is the previous state probability. To avoid either underflow or overflow issue, the joint likelihood function is transformed into the natural logarithm space, and Equation 4.1 then becomes

$$\ln p(\mathbf{y}_{1:\mathtt{T}}|\boldsymbol{\theta}) = \sum_{t=1}^{\mathtt{T}} \ln\left[\sum_{j=1}^{\mathtt{S}}\sum_{i=1}^{\mathtt{S}} \mathcal{L}_{t|t}^{i(j)}\cdot Z^{i(j)}\cdot \pi_{t-1|t-1}^{i}\right], \tag{4.2}$$

where $\ln p(\mathbf{y}_{1:\mathtt{T}}|\boldsymbol{\theta})$ is the joint log-likelihood.

4.3 Maximum Likelihood Estimation

For BDLMs, the MLE approach consists in finding an optimal vector of model parameters, $\boldsymbol{\theta}^*$ by maximizing the log-likelihood presented in Equation 4.2,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \ \ln p(\mathbf{y}_{1:\mathtt{T}}|\boldsymbol{\theta}). \tag{4.3}$$

The optimization task can be done using gradient-based algorithms. This section first covers the mathematical formulation for different gradient-ascent variants. It then details how they are adapted to BDLMs. Note that the term "gradient ascent" is not commonly found in the literature, which most often refers to "gradient descent". This is because the gradient-based optimization is commonly used for minimizing an objective function, for example, a cost

function. However, both gradient descent and ascent are in practice identical, where one can be transformed into the other by taking the negative of the objective function.

4.3.1 Batch Gradient Ascent

Batch Gradient Ascent (BGA) [58], or simply *gradient ascent*, is a gradient-based optimization method used to maximize an objective function such as log-likelihood function. It iteratively updates model parameters with a step, $\Delta_{\boldsymbol{\theta}}^{n}$, in the direction calculated using of the gradient for the entire training dataset $\mathbf{y}_{1:\mathrm{T}}$. The model parameter update is done following

$$\begin{cases} \boldsymbol{\theta}^{n} &= \boldsymbol{\theta}^{n-1} + \Delta_{\boldsymbol{\theta}}^{n} \\ \Delta_{\boldsymbol{\theta}}^{n} &= \eta \nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{1:\mathrm{T}}), \end{cases} \tag{4.4}$$

where n corresponds to the optimization loop, η is the learning rate, ∇ is the gradient operator, and the objective function $\mathcal{F}(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{1:\mathrm{T}})$ depends on the vector of model parameters at the optimization loop $n-1$ and on the training dataset. The BGA algorithm is computationally demanding when dealing with a large dataset, because it computes the gradient using a large amount of observations.

4.3.2 Stochastic Gradient Ascent

Stochastic Gradient Ascent (SGA) [58] is an extension of the BGA algorithm that allows speeding up the optimization procedure. The SGA consists in updating model parameters

$$\begin{cases} \boldsymbol{\theta}^{n} &= \boldsymbol{\theta}^{n-1} + \Delta_{\boldsymbol{\theta}}^{n} \\ \Delta_{\boldsymbol{\theta}}^{n} &= \eta \nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{t}), \end{cases} \tag{4.5}$$

where the gradient is computed using a single observation. An *epoch* is completed when Equation 4.5 has been applied to every observation in the dataset. In common cases, multiple epochs are required for ensuring the convergence. Figure 4.1 shows the behaviour of the log-likelihood function of the BGA and SGA algorithms with respect to the number of epochs. Because only one observation at a time is used for computing the vector of gradients $\nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{t})$, the SGA algorithm shows a high variability in the updates of model parameters, which explains the fluctuations in the log-likelihood illustrated in Figure 4.1b. On the other hand, the log-likehood for the BGA algorithm keeps increasing until converging, as shown in Figure 4.1a.

To improve the stability of the SGA algorithm, it is common to employ *Mini-Batch*

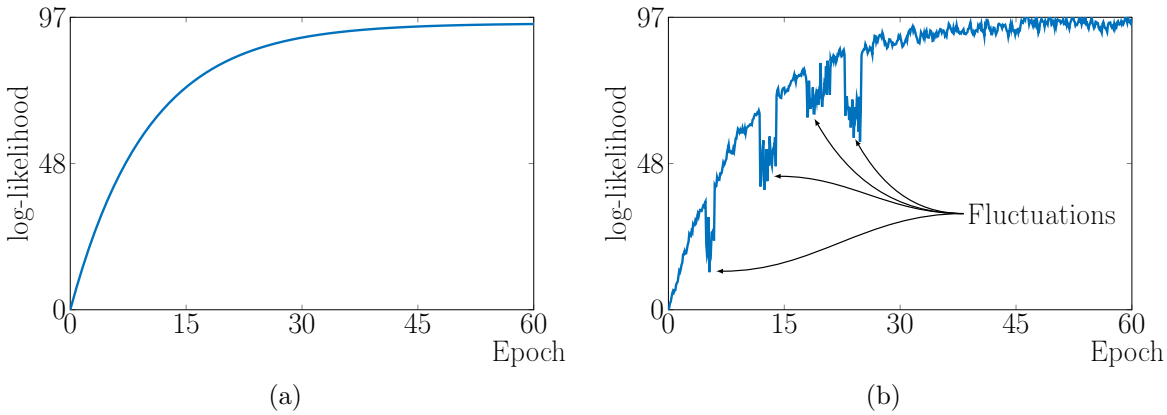


Figure 4.1 The log-likelihood of (a) the batch gradient ascent and (b) stochastic gradient ascent algorithm during training.

Gradient Ascent (MGA) that computes the gradient using mini-batches of data $\mathbf{y}_{t:t+l_{\mathtt{MB}}}$. The model parameter update then follows

$$\begin{cases} \boldsymbol{\theta}^n &= \boldsymbol{\theta}^{n-1} + \Delta_{\boldsymbol{\theta}}^n \\ \Delta_{\boldsymbol{\theta}}^n &= \eta \nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{t:t+l_{\mathtt{MB}}}), \end{cases} \tag{4.6}$$

where $l_{\mathtt{MB}}$ is the length of the mini-batch. The MGA algorithm allows (i) reducing the variability in the updates of model parameters and (ii) providing a speed up for the optimization procedure. A gradual decrease in the learning rate over time for the MGA algorithm is needed in order to achieve the same performance as the BCA algorithm [44].

4.3.3 Common Challenges

Despite the widespread application of the gradient ascent variants to optimization problems, several key challenges [44] remain to be addressed in order to ensure an efficient performance. The first limitation is that its performance relies on how the learning rate is tuned. More specifically, a small learning rate can slow down the learning speed while a large learning rate might overshoot the maximum, as shown in Figure 4.2. In common cases, the same learning rate for all model parameters can be obtained using expert judgment and experience as well as prior data analysis. However, this tuning method becomes inefficient in the case where the objective function is sensitive to some search directions of model parameters as defined by their gradients, yet insensitive to the others. For this case, it is required to update the model parameters with different values of the learning rate.

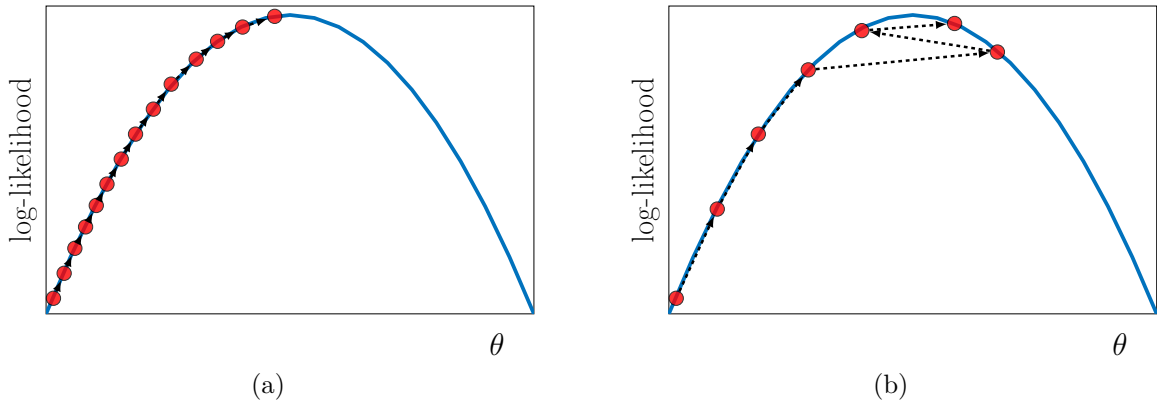


Figure 4.2 Illustration of the impact of the learning rate on the model parameter optimization. (a) a small learning rate; (b) a large learning rate. Each circle presents an update step. The ordering update is presented by the arrows.

The second limitation is that, in the field of machine learning, the objective functions are usually non-convex because of the presence of multiple local maxima, as shown in Figure 4.3a. A local maximum is defined as a point where the first derivative is equal to zero, the

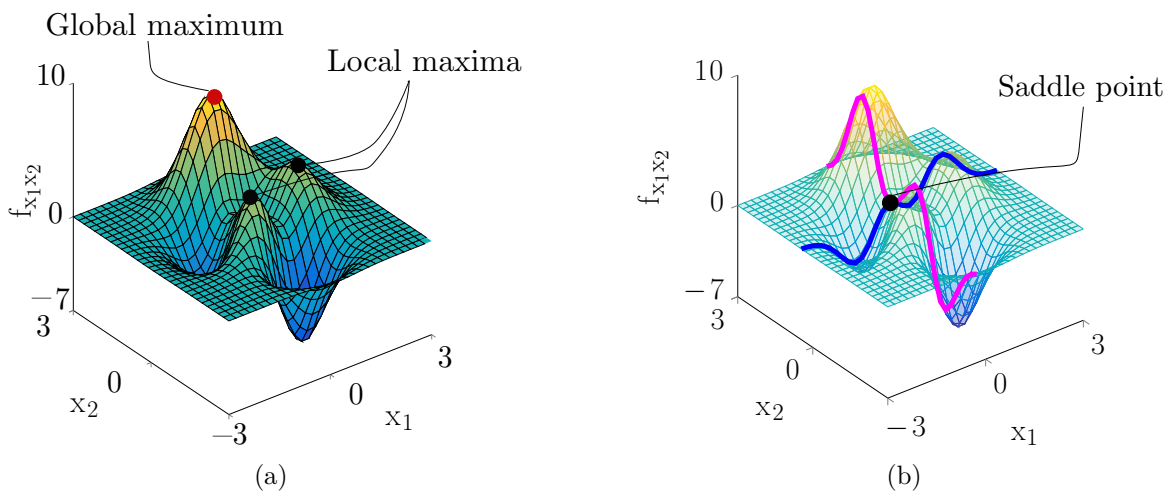


Figure 4.3 Examples of (a) the local maxima and (b) the saddle points.

second derivative is negative, yet it is not the highest value of the objective function. Escaping from these local maxima using gradient ascent is theoretically impossible because the model parameters can no longer be updated due to the zero-value gradient. Another difficulty is the saddle points [102], that is, the location where the objective function has a local minimum

in one dimension but a local maximum in another dimension, as illustrated in Figure 4.3b. These saddle points make it difficult for optimization algorithms to escape from them because the gradient of the objective function around them are close to zero.

The third limitation is that the BGA algorithm is sensitive to the initial values of model parameters. The initial values have a significant impact on the convergence speed of the optimization algorithm. They can also determine whether it converges to a global or local maximum or it gets stuck in a saddle point. In some cases, poor initial values may also cause a numerical instability. In practice, the optimization algorithm should be run several times with some random sets of initial values of model parameters.

All the aforementioned limitations make gradient-based optimization algorithms difficult to fully automate. These algorithms are thus typically coupled with the other algorithms such as *momentum* and *adaptive moment estimation* which are described in detail in the following sections.

4.3.4 Optimization Algorithm

This section reviews the mathematical formulation along with the advantages as well as limitations of five optimization algorithms. The discrepancy between these five algorithms lies on the computation of the step update $\Delta_{\boldsymbol{\theta}}^{n}$ defined in Equation 4.4−4.6.

Newton-Raphson (NR)

The Newton-Raphson (NR) algorithm [98] performs the update of model parameters using the most recent gradient and Hessian. The learning rate is approximated using the inverse of negative Hessian matrix of the objective function with respect to the model parameters. $\Delta_{\boldsymbol{\theta}}^{n}$ is calculated following

$$\Delta_{\boldsymbol{\theta}}^{n} = \left[-\nabla_{\boldsymbol{\theta}}^{2}\mathcal{F}(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{1:\mathrm{T}})\right]^{-1} \cdot \nabla_{\boldsymbol{\theta}}\mathcal{F}(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{1:\mathrm{T}}), \tag{4.7}$$

where $\nabla_{\boldsymbol{\theta}}^{2}\mathcal{F}(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{1:\mathrm{T}})$ is the Hessian matrix of the objective function with respect to the vector of model parameters and the training dataset. The use of the Hessian matrix allows estimating the curvature of the objective function at a given $\boldsymbol{\theta}$. The computational time for evaluating the full Hessian matrix is not negligible when dealing with a large number of model parameters. To reduce the computational cost, it is possible to employ only its main diagonal terms. A key limitation is that the NR algorithm performs poorly in the presence of saddle points and local minima [44].

Momentum (MMT)

For the case where the objective function exhibits pathological curvature [103] (e.g., valleys, ravines, and trenches) or where there is a high variability in the gradient. Both BGA and MGA algorithms gradually oscillate towards the maximum, as presented by the solid line in Figure 4.4. Consequently, these oscillations result in a slow convergence. A momentum

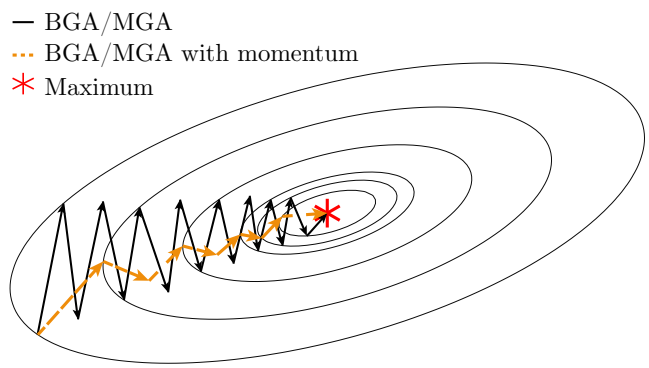


Figure 4.4 Illustration of how the momentum algorithm addresses the limitations of classic gradient ascent variants. Each arrow presents an update step made by the optimization algorithm.

term [104] that accumulates an exponentially decaying average of the past gradients, helps the gradient ascent algorithms to dampen these oscillations. As a result, the MGA and BGA algorithms with momentum lead to a faster convergence than the one without momentum, as shown by the dashed line in Figure 4.4. $\Delta_{\boldsymbol{\theta}}^{n}$ for the MMT algorithm is given by

$$\begin{cases} \Delta_{\boldsymbol{\theta}}^{n} &= \mathbf{r}^{n} \\ \mathbf{r}^{n} &= \beta \mathbf{r}^{n-1} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{F}\left(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{1:\mathrm{T}}\right), \end{cases} \tag{4.8}$$

where $\mathbf{r}$ is the vector of momentum and $\beta \in [0, 1]$ is a hyperparameter that defines the contribution of the previous momentum. In practice, the possible values of β typically are $0.5, 0.9$ and 0.99 [44]. The advantage of the MMT algorithm is that it makes a small update in model parameters, yet it provides an efficient and fast learning process, especially in the presence of saddle points and local maxima. A drawback of the MMT algorithm is that it ignores the bias of the gradients, leading to inaccurate updates for model parameters.

Root Mean Square Propagation (RMSProp)

RMSProp [105] allows each model parameter to have its own adaptive learning rate. Like the MMT algorithm, RMSProp also dampens the oscillations, yet this time it uses an exponentially

decaying average of past squared gradients. The update step for model parameters is defined as

$$\begin{cases} \Delta_{\boldsymbol{\theta}}^{n} &= \left[\eta\left(\sqrt{\hat{\boldsymbol{\nu}}^{n}}+\boldsymbol{\epsilon}\right)^{-1}\right]\odot\nabla_{\boldsymbol{\theta}}\mathcal{F}\left(\boldsymbol{\theta}^{n-1},\mathbf{y}_{1:\mathrm{T}}\right) \\ \boldsymbol{\nu}^{n} &= \beta_1\boldsymbol{\nu}^{n-1}+\left[1-\beta_1\right]\left[\nabla_{\boldsymbol{\theta}}\mathcal{F}\left(\boldsymbol{\theta}^{n-1},\mathbf{y}_{1:\mathrm{T}}\right)\right]^2, \end{cases} \tag{4.9}$$

where $\boldsymbol{\nu}^n$ is the exponential average of the past squared gradients, $\boldsymbol{\epsilon}$ is a vector containing small constants to ensure that $(\boldsymbol{\nu}^n+\boldsymbol{\epsilon})\neq 0$, $\beta_1\in(0,1)$ is a hyperparameter that controls the length scale of the moving average of the past squared gradients, and $\odot$ is the element-wise operator. In practice, β_1 and $\boldsymbol{\epsilon}$ are typically set to 0.9 and 10^{-8}. RMSProp performs well for non-convex optimization problems in which there might be local minima and saddle points [44]. Its main limitation is that it does not take into account the bias of the squared gradients.

Adaptive Moment Estimation (ADAM)

Like RMSProp, ADAM [106] is another approach that provides the adaptive learning rate for each model parameter. This adaptive learning rate is evaluated based on the exponentially decaying average of past squared gradients as well as of past gradients. The ADAM algorithm is a combination of the MMT algorithm with the RMSProp algorithm where their limitations are addressed introducing a bias correction to the gradient and squared gradients. For the ADAM algorithm, the step update, $\Delta_{\boldsymbol{\theta}}^n$, is computed following

$$\begin{cases} \Delta_{\boldsymbol{\theta}}^{n} &= \eta\hat{\mathbf{s}}^{n}\odot\left(\sqrt{\hat{\boldsymbol{\nu}}^{n}}+\boldsymbol{\epsilon}\right)^{-1} \\ \hat{\boldsymbol{\nu}}^{n} &= \frac{\boldsymbol{\nu}}{1-(\beta_1)^n} \\ \hat{\mathbf{s}}^{n} &= \frac{\mathbf{s}}{1-(\beta_2)^n} \\ \boldsymbol{\nu}^{n} &= \beta_1\boldsymbol{\nu}^{n-1}+\left[1-\beta_1\right]\left[\nabla_{\boldsymbol{\theta}}\mathcal{F}\left(\boldsymbol{\theta}^{n-1},\mathbf{y}_{1:\mathrm{T}}\right)\right]^2 \\ \mathbf{s}^{n} &= \beta_2\mathbf{s}^{n-1}+(1-\beta_2)\nabla_{\boldsymbol{\theta}}\mathcal{F}\left(\boldsymbol{\theta}^{n-1},\mathbf{y}_{1:\mathrm{T}}\right), \end{cases} \tag{4.10}$$

where $\mathbf{s}^n$ is the exponential average of the previous gradients, $\hat{\mathbf{s}}^n$ is the bias correction of the exponential average of previous gradients, $\hat{\boldsymbol{\nu}}^{n-1}$ is the bias correction of the exponential average of the previous squared gradients, β_1 is defined in Equation 4.9, and $\beta_2\in(0,1)$ is a hyperparameter that controls the length scale of the moving average of the previous gradients. β_2 is typically set to 0.999.

Adaptive Momentum (AMMT)

AMMT is a combination of the NR algorithm with the MMT algorithm. The AMMT algorithm provides the adaptive learning rate for each model parameter while overcoming

the limitations of the NR and MMT algorithms. The AMMT algorithm accumulates an exponentially decaying average of the past gradients and diagonal Hessian terms [107]. The update step for model parameters $\Delta_{\boldsymbol{\theta}}^{n}$ is given by

$$\begin{cases} \Delta_{\boldsymbol{\theta}}^{n} &= \left(\bar{\mathbf{h}}^{n}\right)^{-1} \cdot \mathbf{s}^{n} \\ \mathbf{s}^{n} &= \beta \mathbf{s}^{n-1} + (1-\beta)\nabla_{\boldsymbol{\theta}}\mathcal{F}\left(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{1:\mathrm{T}}\right) \\ \bar{\mathbf{h}}^{n} &= \beta \bar{\mathbf{h}}^{n-1} + (1-\beta)\nabla_{\boldsymbol{\theta}}^{2}\mathcal{F}\left(\boldsymbol{\theta}^{n-1}, \mathbf{y}_{1:\mathrm{T}}\right), \end{cases} \tag{4.11}$$

where $\bar{\mathbf{h}}^{n}$ is the exponential average of the previous diagonal Hessian terms, $\mathbf{s}^{n}$ is defined in Equation 4.10, and β is defined in Equation 4.8.

A general remark is that there is no absolute rule for selecting the optimization algorithm. Instead, the choice of the optimization algorithm depends on (i) the dataset, (ii) the model architecture, and (iii) the user's preference. Furthermore, the performance of an optimization algorithm relies on the hyperparameter tuning.

4.3.5 Practical Implementation for BDLMs

To provide an efficient and fast learning process for BDLMs, the BGA and MGA algorithms need to be designed in such as a way that they can meet some specific requirements for BDLMs. The section proposes practical implementations of both BGA and MGA algorithms for optimizing the vector of model parameters $\boldsymbol{\theta} = [\theta_1\ \theta_2\ \ldots\ \theta_{\mathrm{p}}]^{\intercal}$ for BDLMs. The first part presents a general implementation for both algorithms. The following parts provide the specifications of the implementation for each algorithm in order to maximize their performance in BDLMs. Note here that the objective function is the log-likelihood function presented in Equation 4.2. For simplicity purpose, the log-likelihood function is denoted as $\mathcal{F}(\boldsymbol{\theta}, \mathbf{y}_{1:\mathrm{T}})$ that depends on the model parameters $\boldsymbol{\theta}$ and training observations $\mathbf{y}_{1:\mathrm{T}}$.

General Implementation

In BDLMs, the analytic formulation of the log-likelihood function is typically complex to obtain. Therefore, its first and second derivatives are evaluated using the finite difference method,

$$\begin{aligned} g_p &= \frac{\mathcal{F}(\boldsymbol{\theta}(p) + \delta_\theta, \mathbf{y}_{1:\mathrm{T}}) - \mathcal{F}(\boldsymbol{\theta}(p) - \delta_\theta, \mathbf{y}_{1:\mathrm{T}})}{2\delta_\theta} \\ h_p &= \frac{\mathcal{F}(\boldsymbol{\theta}(p) + \delta_\theta, \mathbf{y}_{1:\mathrm{T}}) - \mathcal{F}(\boldsymbol{\theta}(p), \mathbf{y}_{1:\mathrm{T}}) + \mathcal{F}(\boldsymbol{\theta}(p) - \delta_\theta, \mathbf{y}_{1:\mathrm{T}})}{\delta_\theta^2}, \end{aligned} \tag{4.12}$$

where g_p is the first derivative, h_p is the second derivative, and δ_θ is a small change in $\boldsymbol{\theta}(p)$. The details of their implementation are presented in Algorithm 2 in Appendix B.1. Note that these numerical approximations can introduce errors into first and second derivatives. For an efficient and fast optimization, the five optimization algorithms presented in §4.3.4 can be employed to perform the updates of model parameters. Algorithm 3 in Appendix B.2 presents a simple version of the implementation for the model parameter updates.

For the convergence criterion, a metric needs to be computed after each epoch. The optimization algorithm can use either the log-likelihood value (L_{tr}) of the training set or the log-likelihood value (L_v) of the validation set. Figure 4.5 illustrates how the data are used to evaluate each metric as well as the first and second derivatives for updating model parameters. Note that the validation set is not used to optimize model parameters. In common cases,

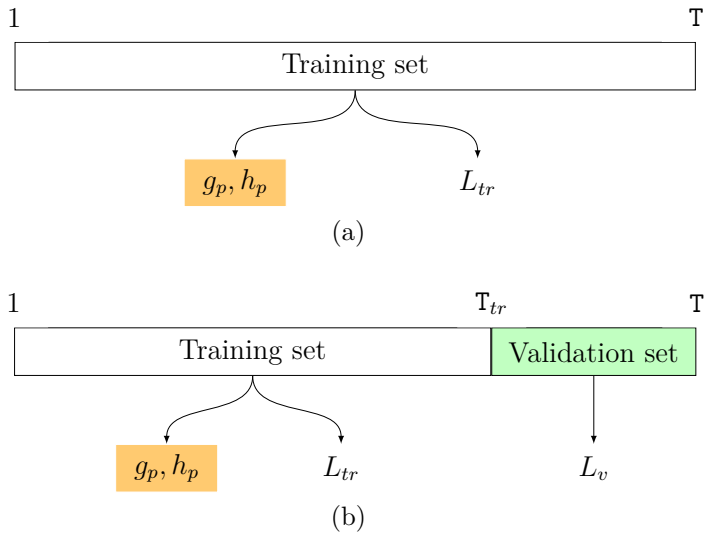


Figure 4.5 Illustration of two metrics employed in BDLMs to evaluate the performance of the optimization algorithm. (a) training log-likelihood value (L_{tr}); (b) validation log-likelihood value (L_v).

the validation log-likelihood value is used as a metric when a small dataset is employed to update model parameters (e.g., a mini-batch of data in the MGA algorithm). Such a case is prone to overfitting when the performance of the model on training set is better than on the validation set. It means that the model does not generalize well from the training set to the validation set. Figure 4.6 shows an example of this overfitting issue. In this example, the training log-likelihood keeps increasing while the validation log-likelihood starts decreasing after the 37th epoch. Here, the best-case scenario is to stop the optimization algorithm at the epoch 37. Note that the validation log-likelihood is not an accurate metric when it comes to

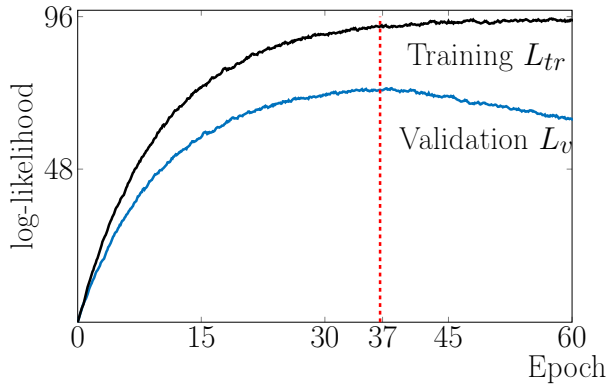


Figure 4.6 Illustration of the log-likelihood function of the training set and validation set.

anomaly detection because the occurrence of the abnormal events is not frequent in a time series. For example, the validation set might include these events, yet the training set does not, meaning that the training and validation sets come from different distributions. This phenomenon is called *data shift* [108]. Such a case, it is not well suited to measure the model performance on the validation set whose the distribution is different from the one in the training set. The implementation of two metrics is presented in Algorithm 4 in Appendix B.3.

BGA Implementation

Each model parameter in BDLMs has its own contribution to the log-likelihood value. Therfore, the log-likelihood might be more sensitive to some model parameters than others. Only updating the model parameter having the most relevant contribution to the log-likelihood function at each epoch may lead to the maximum faster than updating all model parameters at once. This update technique is a variant of the *coordinate ascent* technique [109, 110]. A limitation of the coordinate ascent is that it does not perform well if there are dependencies between model parameters. Another limitation is that the computational cost increases rapidly with the number of model parameters such as in the case of a neural network which contains millions of model parameters to be optimized. In the context of BDLMs, however, the number of unknown model parameters commonly varies from 1 to 20, which makes this technique computationally efficient. For these reasons, updating each model parameter once at a time is proposed for the BGA algorithm in BDLMs.

The challenge now is how to select the model parameter to perform the updates for the next epoch because there is neither general rule nor prior knowledge to determine which model parameter is more relevant than others. Randomly shuffling the model parameter

for each epoch can be used for this task, yet it does not take into account the information from data as well as model architecture, thus making it unreliable. Another way to address this challenge is to define a distance measure between the current and previous values of the training log-likelihood, L_{tr}^{n} and L_{tr}^{n-1}, for each model parameter as the learning progresses,

$$\mathbf{d}(p) = L_{tr}^{n} - L_{tr}^{n-1}, \tag{4.13}$$

where p corresponds to the pth model parameter of $\boldsymbol{\theta}$. Once $\mathbf{d}$ is evaluated, the index p of the next model parameter to be updated is sampled from a discrete distribution with a probability proportional to

$$\mathbf{p}_d = \frac{\mathbf{d}}{\sum_{p=1}^{\mathtt{P}} \mathbf{d}(p)}. \tag{4.14}$$

The algorithm for sampling the index p is presented Algorithm 5 in Appendix B.4. Because the log-likelihood function contains both information from data and model architecture, it can theoretically provide a better performance than the random selection technique. Note here that this selection procedure can be biased due to a high variability in the data. The five optimization algorithms presented in §4.3.4 can be coupled with the BGA algorithm.

Like any other optimization algorithms, a convergence criterion needs to be defined. Because the model parameters are updated once at a time, a converged vector $\mathbf{c}$ is defined following

$$\mathbf{c}(p) = \begin{cases} 1 & \text{if } \; L_{tr}^{n} > L_{tr}^{n-1} \text{ and } \left| \dfrac{L_{tr}^{n} - L_{tr}^{n-1}}{L_{tr}^{n-1}} \right| < tol \\ 0 & \text{otherwise,} \end{cases} \tag{4.15}$$

where L_{tr}^{n} and L_{tr}^{n-1} are the current and previous values of the training log-likelihood, and tol is a convergence tolerance that needs to be tuned before optimizing. The optimization procedure has converged when either all elements of $\mathbf{c}$ are equal to one or the number of epoch reaches the allowed maximum. A version of the convergence check can be implemented as described in Algorithm 6 in Appendix B.5. The training log-likelihood L_{tr} (see Figure 4.5) is typically used as the metric for the BGA algorithm. The first reason is that, as discussed in the previous section, it is better suited than the validation log-likelihood in the case of anomaly detection, where the BGA algorithm is a common choice for optimizing its model parameters. Second, it allows reducing the computational cost, thus leading a fast optimization. Third, using the entire dataset to optimize model parameters is less sensitive to overfitting than using a small portion of the dataset like the MGA algorithm. Algorithm 8 presented in Appendix B.6 summarizes the main steps of the BGA algorithm for BDLMs.

MGA Algorithm

As presented in §4.3.2, the MGA algorithm is used for improving the optimization speed when dealing with a large dataset by separating the training data into mini-batches in order to update model parameters. There are two main features in BDLMs for which the MGA algorithm needs to be adapted: (i) the selection of data for the mini-batch and (ii) the use of mini-batches of data for optimizing model parameters. In general cases, to avoid biasing the optimization algorithm, the classic MGA algorithm shuffles the training data for the mini-batches at every update. This requirement becomes problematic for BDLMs that specializes in interpreting sequential data for which the order of observations matters. To tackle this problem, the starting time $t_s \in [1 : \mathtt{T} - l_{\mathtt{MB}}]$ for each mini-batch is randomly selected. Each mini-batch of data is then built by taking the observations with respect to their ordering in the time window from t_s to $t_s + l_{\mathtt{MB}}$, where $l_{\mathtt{MB}}$ is the length of mini-batch. An epoch is completed when the model parameters are updated with $\mathtt{N}_{\mathtt{maxM}} = \lfloor \mathtt{T}/l_{\mathtt{MB}} \rceil$ times, where $\lfloor \cdot \rceil$ represents the nearest integer. To prevent overfitting, the validation log-likelihood is used as a metric to measure the performance of the optimization procedure. Figure 4.7 illustrates how the data are selected for each mini-batch.

The key challenges with the use of mini-batches of data are that (i) the selection technique for the model parameter for the next update in the BGA algorithm is no longer applied to the MGA algorithm because the mini-batch of data is not the same at every update, so the contribution of each model parameter cannot be evaluated, (ii) its first and second derivatives are much noisier than when using the full batch of data, and (iii) it can be prone to overfitting because of small datasets. For addressing the first challenge, the multiple possibilities of the model parameter update are proposed for the MGA algorithm in BDLMs. It means that each model parameter has the same potential to be updated for the next epoch. More specifically, a matrix in which each column represents a possible update for the model parameters is defined following

$$\mathbf{M}_{\boldsymbol{\theta}} = \begin{bmatrix} \theta_1 + \Delta_{\theta_1} & \theta_1 & \dots & \theta_1 & \theta_1 + \Delta_{\theta_1} \\ \theta_2 & \theta_2 + \Delta_{\theta_2} & \dots & \theta_2 & \theta_2 + \Delta_{\theta_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \theta_{\mathtt{p}} & \theta_{\mathtt{p}} & \dots & \theta_{\mathtt{p}} + \Delta_{\theta_{\mathtt{p}}} & \theta_{\mathtt{p}} + \Delta_{\theta_{\mathtt{p}}} \end{bmatrix}, \tag{4.16}$$

where Δ_θ is the step update. Note that the last column of $\mathbf{M}_{\boldsymbol{\theta}}$ presents a possibility of updating all model parameters at once. Because Δ_θ for each model parameter has been computed, this possibility is obtained without requiring additional computations and it may lead to a faster convergence. Each column of $\mathbf{M}_{\boldsymbol{\theta}}$ is associated with a value of the validation

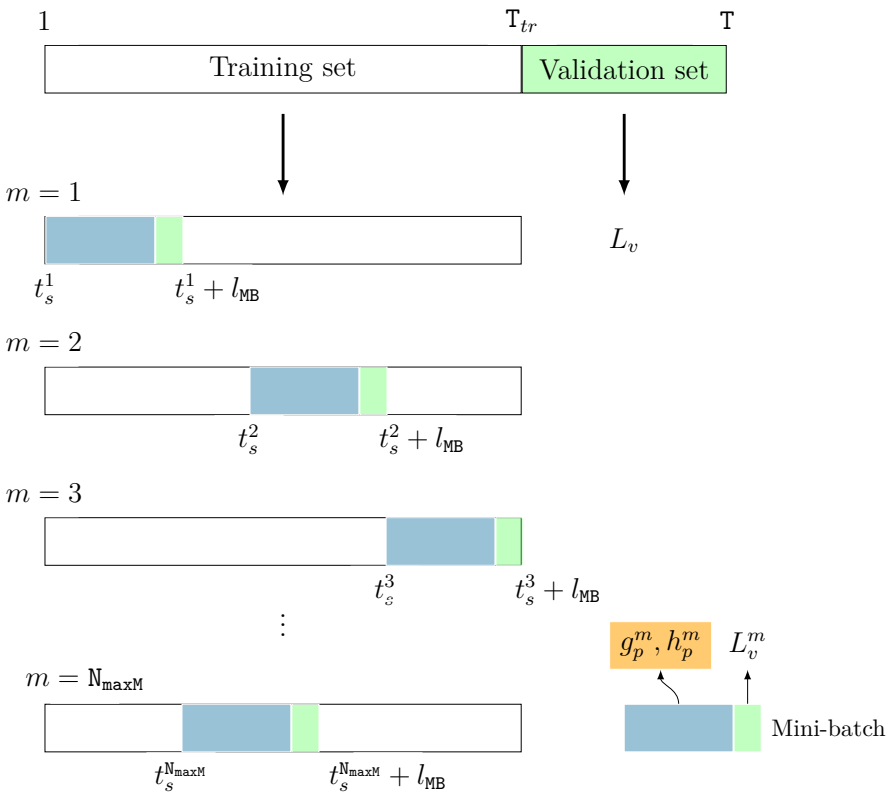


Figure 4.7 Illustration of the selection of the mini-batch of data for the mini-batch gradient ascent algorithm. g_p: the first derivative of $\mathcal{F}(\boldsymbol{\theta}, \mathbf{y}_{1:\mathtt{T}})$ with respect to $\boldsymbol{\theta}(p)$; h_p: the second derivative of $\mathcal{F}(\boldsymbol{\theta}, \mathbf{y}_{1:\mathtt{T}})$ with respect to $\boldsymbol{\theta}(p)$; L_v: the validation log-likelihood.

log-likehood computed using Algorithm 4 so that a vector of validation log-likelihood values is given by

$$\mathbf{L}_v^m = \left[L_{v,1}^m \; L_{v,2}^m \; \ldots \; L_{v,\mathtt{p}}^m\right]^\intercal, \tag{4.17}$$

where m is the mini-batch loop. The new vector of the model parameters corresponds to the column of $\mathbf{M}_{\boldsymbol{\theta}}$ having the highest validation log-likelihood value of $\mathbf{L}_v^m$. This update technique can lead to the optimal maximum faster than the previous one because (i) it considers different update configurations of model parameters that are updated either one model parameter at a time or all model parameters at once and (ii) the computational time for a mini-batch of data is small. In the current implementation, $\mathbf{M}_{\boldsymbol{\theta}}$ is not include all possibilities of model parameter update because it is computationally demanding. Further improvements can focus on increasing the size of $\mathbf{M}_{\boldsymbol{\theta}}$ by well selecting different combinations of model parameters to be updated in such a way that the model parameters relating to the same hidden state variable should be updated at once.

To tackle the second challenge, the update of model parameters is performed using one of four options following: MMT, RMSProp, ADAM, and AMMT algorithms. The convergence is reached when one of the following conditions is satisfied

$$\left[\begin{array}{lcl} L_v^n & < & tol \cdot L_v^{n-1} \\ n & = & \mathtt{N}_{\mathtt{maxEpoch}}, \end{array}\right. \tag{4.18}$$

where tol is convergence tolerance and $\mathtt{N}_{\mathtt{maxEpoch}}$ is the maximal number of epochs. Algorithm 7 presented in Appendix B.5 presents the convergence check after each epoch for the MCA algorithm.

Despite all advantages offered by the MGA algorithm, the practical application shows that it does not provide a stable optimization in the case of anomaly detection. This can be explained by the infrequent occurrence of abnormal events. It means that some of the mini-batches may include them but not to the others, leading to a high variability when updating model parameters. Therefore, the BGA algorithm is more suited than the MGA algorithm to such a case. The details of the MGA algorithm proposed for the BDLMs are presented in Algorithm 9 in Appendix B.7.

Initialization Strategies

As mentioned in §4.3.3, the performance of both BGA and MGA algorithms depends on the initial parameter values $\boldsymbol{\theta}^0$. In the context of BDLMs, they are additionally dependent on the initial mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$ for the hidden state variables. Poor guesses for the

initial values of either model parameters or hidden state variables can lead to suboptimal local maxima. In the case of anomaly detection, it can trigger false alarms or prevent the detection of anomalies. For addressing the first limitation, several runs with different sets of initial values of $\boldsymbol{\theta}$ should be tested to ensure proper initial values. The *multi-pass* technique is proposed to tackle the second limitation. The multi-pass consists in recursively employing the Switch Kalman Smoother (SKS) [86] for estimating $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$ for hidden state variables. During training, the model is first built using the vector of initial model parameter

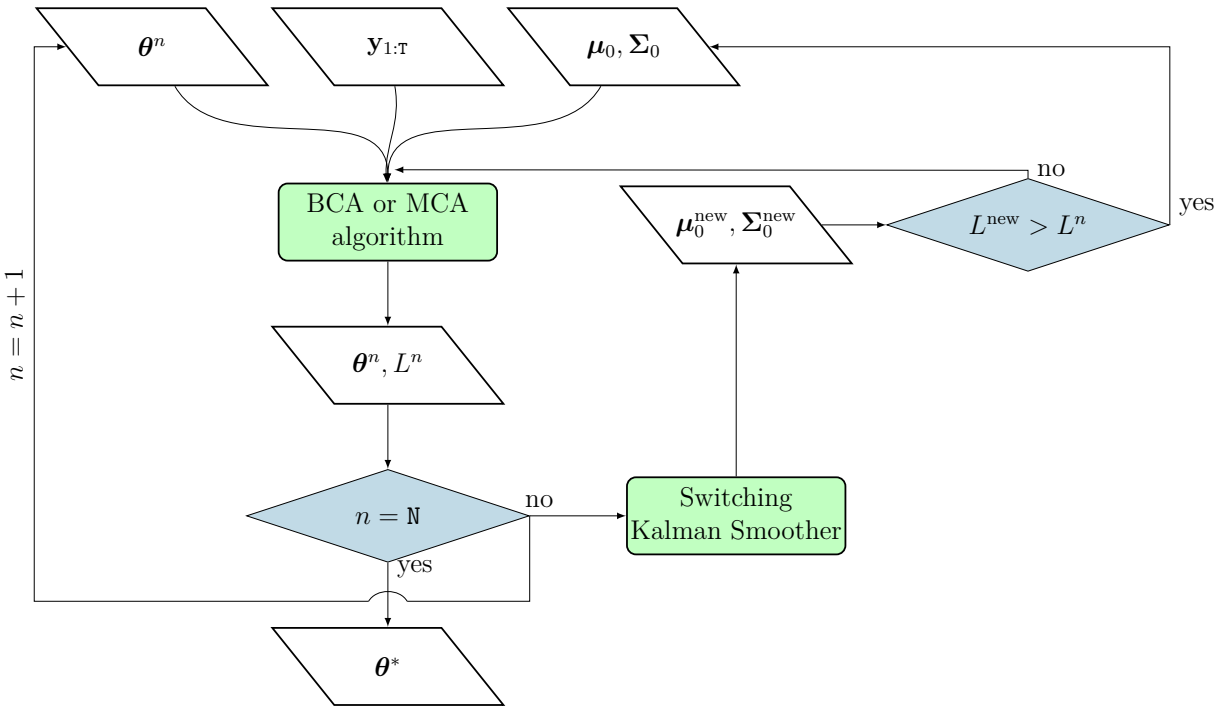


Figure 4.8 Illustration of the batch optimization procedure using the multi-pass.

values $\boldsymbol{\theta}^n$, initial values for hidden state variables $\{\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\}$, and the training data $\mathbf{y}_{1:\mathtt{T}}$. The optimization algorithm is employed for optimizing the vector of model parameters $\boldsymbol{\theta}^n$ and for evaluating the corresponding log-likelihood L^n. After each iteration, $\boldsymbol{\mu}_0^{\text{new}}$ and $\boldsymbol{\Sigma}_0^{\text{new}}$ are estimated using SKS. To be accepted as new initial values, the log-likelihood evaluated using $\{\boldsymbol{\mu}_0^{\text{new}}, \boldsymbol{\Sigma}_0^{\text{new}}\}$ needs to be greater than L^n. This procedure is repeated until the convergence criterion, $n = \mathtt{N}$, are met. The final output of the procedure is the vector of optimal parameters $\boldsymbol{\theta}^*$. Note that $\mathtt{N}$ needs to be defined before the optimization. To reduce the computation cost, the amount of data employed for estimating the initial values $\boldsymbol{\mu}_0^{\text{new}}$ and $\boldsymbol{\Sigma}_0^{\text{new}}$ can be smaller than the training data employed for the optimization procedure.

4.4 Bayesian Estimation

In the context of BDLMs, the uncertainty in the model parameters is a key aspect for quantifying the confidence over the estimation of hidden state variables. The reason behind this uncertainty is the limited amount of data for learning model parameters. The uncertainty in model parameters can be incorporated in a decision-making process in order to provide risk-aware decisions. For instance, the MLE approach presented in Section 4.3 only provides point estimates for model parameters which does not take into account the model parameter uncertainty. To overcome this limitation, the section presents two methods for approximating the posterior PDF of model parameters: *Hamiltonian Monte Carlo* [111–113] and *Laplace approximation* [98]. A *Gaussian mixture approximation* [87] is then employed for propagating the model parameter uncertainty towards the hidden state variables.

4.4.1 Hamiltonian Monte Carlo (HMC)

The posterior PDF for the vector of model parameters defined as

$$\begin{aligned} p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathsf{T}}) &= \frac{p(\mathbf{y}_{1:\mathsf{T}}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta})}{p(\mathbf{y}_{1:\mathsf{T}})} \\ &\propto p(\mathbf{y}_{1:\mathsf{T}}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}), \end{aligned} \tag{4.19}$$

where $p(\boldsymbol{\theta})$ is the *prior* PDF, $p(\mathbf{y}_{1:\mathsf{T}}|\boldsymbol{\theta})$ is the marginal likelihood, $p(\mathbf{y}_{1:\mathsf{T}})$ is a normalization constant, and $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \cdots \theta_{\mathsf{P}}]^{\mathsf{T}}$ is a vector of model parameters to be estimated. The prior probability represents the knowledge available for model parameter values before the data have been collected. The marginal likelihood formulation for the BDLMs is shown in Equation 4.1. HMC is a Markov chain monte carlo method for approximating the posterior PDF $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathsf{T}})$ for model parameters $\boldsymbol{\theta}$, given the training data $\mathbf{y}_{1:\mathsf{T}}$. The particularity of the HMC algorithm is that an auxiliary momentum variable r_i is added to each parameter θ_i. The joint probability density for a parameter vector $\boldsymbol{\theta}$ and its momentum variable vector $\mathbf{r}$ is defined as

$$p(\boldsymbol{\theta}, \mathbf{r}|\mathbf{y}_{1:\mathsf{T}}) = p(\mathbf{r}|\boldsymbol{\theta}, \mathbf{y}_{1:\mathsf{T}}) \cdot p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathsf{T}}), \tag{4.20}$$

where $p(\mathbf{r}|\boldsymbol{\theta}, \mathbf{y}_{1:\mathsf{T}})$ is a conditional probability density of $\mathbf{r}$ given $\boldsymbol{\theta}$. The joint density $p(\boldsymbol{\theta}, \mathbf{r}|\mathbf{y}_{1:\mathsf{T}})$ is also called the *canonical distribution* that is independent from the choice of parameterization [114]. Hence, the joint probability density can be written in another form using an invariant *Hamiltonian* function $H(\boldsymbol{\theta}, \mathbf{r})$ as

$$p(\boldsymbol{\theta}, \mathbf{r}|\mathbf{y}_{1:\mathsf{T}}) = \exp\left[-H(\boldsymbol{\theta}, \mathbf{r})\right]. \tag{4.21}$$

The Hamiltonian function originally comes from the field of classical mechanics where it refers to the energy at specific points and is *conservative* over time. In most cases, $H(\boldsymbol{\theta}, \mathbf{r})$ is decomposed into two terms

$$H(\boldsymbol{\theta}, \mathbf{r}) = T(\boldsymbol{\theta}, \mathbf{r}) + V(\boldsymbol{\theta}),$$

where $T(\boldsymbol{\theta}, \mathbf{r})$ is the *kinetic* energy and $V(\boldsymbol{\theta})$ is the *potential* energy. In the case of HMC, $H(\boldsymbol{\theta}, \mathbf{r})$ can be obtained using the Equations 4.19 , 4.20, and 4.21 so that

$$\begin{aligned} H(\boldsymbol{\theta}, \mathbf{r}) &= \overbrace{-\ln\, p(\mathbf{r})}^{T(\mathbf{r})} \; \overbrace{-\ln\, p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathrm{T}})}^{V(\boldsymbol{\theta})} \\ &= -\ln\, p(\mathbf{r}) - \ln\, p(\mathbf{y}_{1:\mathrm{T}}|\boldsymbol{\theta}) - \ln p(\boldsymbol{\theta}) \end{aligned} \tag{4.22}$$

with the assumption that the momentum variables $\mathbf{r}$ do not depend on $\boldsymbol{\theta}$. In common cases, the kinetic energy is defined as

$$T(\mathbf{r}) = \frac{1}{2}\mathbf{r}^{\intercal}\mathcal{M}^{-1}\mathbf{r}, \tag{4.23}$$

where $\mathcal{M}$ is a symmetric and positive-definite *mass matrix*. $\mathcal{M}$ corresponds to the inverse covariance matrix of $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathrm{T}})$. Once the kinetic and potential energies are identified, the Hamiltonian's equations over time can be written as

$$\begin{aligned} \frac{d\mathbf{r}}{dt} &= -\nabla_{\boldsymbol{\theta}}\, V(\boldsymbol{\theta}) \\ \frac{d\boldsymbol{\theta}}{dt} &= \nabla_{\mathbf{r}}\, T(\mathbf{r}), \end{aligned} \tag{4.24}$$

where ∇ is the gradient operator. Equation 4.24 is employed to propose new samples in HMC. For practical implementations, Equation 4.24 can be approximated using the *leapfrog* method [111], that is, a *symplectic integrator*, that allows simulating the trajectories for an efficient exploration of the posterior density $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathrm{T}})$. Given a time discretization parameter l, the main steps in the leapfrog method are written as

$$\begin{aligned} \mathbf{r}^{l+1/2} &= \mathbf{r}^{l} &&- \tfrac{\xi}{2}\,\nabla_{\boldsymbol{\theta}}\, V(\boldsymbol{\theta}^{l}) \\ \boldsymbol{\theta}^{l+1} &= \boldsymbol{\theta}^{l} &&+ \xi\,\nabla_{\mathbf{r}}\, T(\mathbf{r}^{l+1/2}) \\ \mathbf{r}^{l+1} &= \mathbf{r}^{l+1/2} &&- \tfrac{\xi}{2}\,\nabla_{\boldsymbol{\theta}}\, V(\boldsymbol{\theta}^{l+1}), \end{aligned} \tag{4.25}$$

where ξ is defined as the step size. A half step for the momentum vector $\mathbf{r}^{l+1/2}$ is first evaluated. Then, a full step for $\boldsymbol{\theta}^{l+1}$ is updated using $\mathbf{r}^{l+1/2}$. Finally, the other half step for the momentum vector $\mathbf{r}^{l+1}$ is computed using $\boldsymbol{\theta}^{l+1}$. The iterative process is repeated a number of steps $\mathtt{L}_s$. A limitation of the leapfrog method is that it can introduce errors during

the discretization, leading to a bias. Therefore, an acceptance probability β is defined to ensure the validity of the Markov chain:

$$\begin{aligned} \beta &= \min\left\{1, \frac{\exp\left[-H(\boldsymbol{\theta}^{l+1}, \mathbf{r}^{l+1})\right]}{\exp\left[-H(\boldsymbol{\theta}^{l}, \mathbf{r}^{l})\right]}\right\} \\ &= \min\left\{1, \exp\left[-T(\mathbf{r}^{l+1}) - V(\boldsymbol{\theta}^{l+1}) + T(\mathbf{r}^{l}) + V(\boldsymbol{\theta}^{l})\right]\right\}. \end{aligned} \tag{4.26}$$

The key challenge in HMC is to tune the parameters such as the step size ξ and the number of steps $\mathtt{L}_s$ [114, 115]. A small step size provides a more accurate approximation and effective exploration, yet it is computationally more demanding. A large step-size leads to inaccurate simulations and yields low acceptance probabilities. Similarly, a small number of steps yields a high autocorrelation between the successive samples. A larger number of steps causes back loop trajectories [111, 115], leading to a poor exploration. Optimal values for ξ and $\mathtt{L}_s$ are tuned based on the acceptance probability in Equation 4.26 using either the Dual Averaging method [116] or the No-U-Turn Sampler method [115].

The convergence diagnostic statistic $\hat{\mathbf{R}}$ [117], that is, the *Estimated Potential Scale Reduction* (EPSR) is employed to measure the efficiency exploration in HMC. The idea behind is to interpret the stationarity of multiple, parallel Markov chains based on the quantity $\hat{\mathbf{R}}$. If $\hat{\mathbf{R}}$ is approximately 1, the estimates obtained from the Markov chains are deemed reliable.

4.4.2 Laplace Approximation (LAP)

The MLE approach presented in Section 4.3 does not take into account the uncertainty in the model parameter estimates. One solution to tackle this problem is to combine the MLE with the Laplace Approximation (LAP) [58]. The LAP consists in approximating the posterior PDF of parameters with a Gaussian PDF

$$p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathrm{T}}) \approx \mathcal{N}\left(\boldsymbol{\theta}; \boldsymbol{\theta}^*, \underbrace{\left(-\nabla^2_{\boldsymbol{\theta}}\mathcal{F}(\boldsymbol{\theta}^*, \mathbf{y}_{1:\mathrm{T}})\right)^{-1}}_{\mathrm{cov}(\boldsymbol{\theta}^*, \mathbf{y}_{1:\mathrm{T}})}\right), \tag{4.27}$$

where $\nabla^2_{\boldsymbol{\theta}}\mathcal{F}(\boldsymbol{\theta}^*, \mathbf{y}_{1:\mathrm{T}})$ is the Hessian of the log-likelihood function evaluated at the optimal model parameter values $\boldsymbol{\theta}^*$ for the entire training dataset.

4.4.3 Gaussian Mixture Approximation

This section proposes a method based on a *Gaussian Mixture Approximation* (GMAP) [89] for including the model parameter uncertainty in the estimation of the hidden state variables in BDLMs. After having estimated the posterior PDF for the model parameters $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathtt{T}})$, the posterior PDF for hidden state variables can be obtained by integrating over the vector of model parameters $\boldsymbol{\theta}$,

$$p(\mathbf{x}_t|\mathbf{y}_{1:\mathtt{T}}) = \int p(\mathbf{x}_t|\boldsymbol{\theta}, \mathbf{y}_{1:\mathtt{T}}) \cdot p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathtt{T}})\ d\boldsymbol{\theta}, \tag{4.28}$$

where $p(\mathbf{x}_t|\boldsymbol{\theta}, \mathbf{y}_{1:\mathtt{T}})$ is assumed to be a multivariate Gaussian PDF, as shown in Equation 3.1. With the HMC approach presented §4.4.1, each model parameter is now represented by a realization of the posterior PDF. On the other hand, the LAP approach presented in §4.4.2 allows generating samples for model parameters using a Gaussian distribution in Equation 4.27. Equation 4.28 can be approximated by a weighted sum of multivariate Gaussian PDFs, that is, a Gaussian mixture. Assuming that there are $\mathtt{K}$ samples from the model parameter posterior PDF approximated using either the MLE with the Laplace approximation, or the HMC method, the mixture density of hidden state variables is built from a linear combination of $\mathtt{K}$ Gaussian densities. The mathematical formulation for the mixture density is written as

$$\begin{aligned} p(\hat{\mathbf{x}}_t|\mathbf{y}_{1:\mathtt{T}}) &= \sum_{k=1}^{\mathtt{K}} p(\mathbf{x}_t^k|\boldsymbol{\theta}^k, \mathbf{y}_{1:t}) \cdot w^k \\ &= \sum_{k=1}^{\mathtt{K}} \mathcal{N}(\mathbf{x}_t^k; \boldsymbol{\mu}_{t|t}^k, \Sigma_{t|t}^k) \cdot w^k, \end{aligned} \tag{4.29}$$

where $\boldsymbol{\mu}_{t|t}^k$ and $\boldsymbol{\Sigma}_{t|t}^k$ are obtained using Equation 3.4 and w^k are the mixing weights. Note that the sum of all mixing weights are equal to one. Because, in common cases, these multivariate Gaussian PDFs are not far from each other, the Gaussian mixture can be approximate by a single multivariate Gaussian PDF [87, 89, 90], as illustrated in Figure 4.9. This Gaussian PDF is the closest one to the true mixture PDF. $p(\hat{\mathbf{x}}_t|\mathbf{y}_{1:\mathtt{T}})$ is approximated by a Gaussian PDF with mean $\hat{\boldsymbol{\mu}}_{t|t}$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_{t|t}$ that can be calculated following

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{t|t} &= \sum_{n=1}^{\mathtt{K}} \boldsymbol{\mu}_{t|t}^k \cdot w^k \\ \hat{\boldsymbol{\Sigma}}_{t|t} &= \sum_{k=1}^{\mathtt{K}} \boldsymbol{\Sigma}_{t|t}^k \cdot w^k + \sum_{k=1}^{\mathtt{K}} (\boldsymbol{\mu}_{t|t}^k - \hat{\boldsymbol{\mu}}_{t|t})(\boldsymbol{\mu}_{t|t}^k - \hat{\boldsymbol{\mu}}_{t|t})^{\mathtt{T}} \cdot w^k. \end{aligned} \tag{4.30}$$

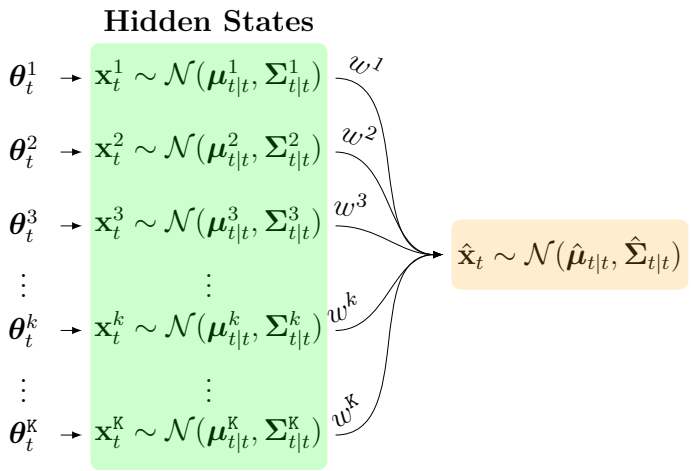


Figure 4.9 Illustration of the Gaussian mixture approximation for the hidden state variables.

Because samples $\boldsymbol{\theta}^k$ are realizations of the posterior PDF, w^k are all equals to $1/\mathtt{K}$ and Equation 4.30 simplifies to

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{t|t} &= \frac{1}{\mathtt{K}} \sum_{k=1}^{\mathtt{K}} \boldsymbol{\mu}_{t|t}^k \\ \hat{\boldsymbol{\Sigma}}_{t|t} &= \frac{1}{\mathtt{K}} \left[\sum_{k=1}^{\mathtt{K}} \boldsymbol{\Sigma}_{t|t}^k + \sum_{k=1}^{\mathtt{K}} (\boldsymbol{\mu}_{t|t}^k - \hat{\boldsymbol{\mu}}_{t|t})(\boldsymbol{\mu}_{t|t}^k - \hat{\boldsymbol{\mu}}_{t|t})^{\intercal} \right]. \end{aligned} \tag{4.31}$$

4.4.4 Framework Architecture for BDLMs

This section proposes the general framework architectures for the Laplace approximation as well as Hamiltonian Monte Carlo procedure. Both frameworks are specifically tailored for approximating the posterior PDF of model parameters and to estimate the hidden state variables.

Laplace Approximation Procedure (LAP-P)

The LAP-P consists in two main steps: *Posterior Density Approximation* (PDA) and *Uncertainty Marginalization* (UM). The PDA-step approximates the parameter posterior density $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathtt{T}})$ using the Laplace approximation presented in §4.4.2. The UM-step is related to the estimation of the expected values for hidden state variables $\hat{\boldsymbol{\mu}}_{t|t}$ and its covariance matrix $\hat{\boldsymbol{\Sigma}}_{t|t}$ based on Gaussian mixture approximation presented in Section 4.4.3 for BDLMs.

In the PDA-step, the optimal parameter vector $\boldsymbol{\theta}^*$ of a model is first learned from

a training set $\mathbf{y}_{1:\mathrm{T}}$ using either BGA or MGA algorithms presented in Section 4.3. The parameter posterior density $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathrm{T}})$ is then approximated using the Laplace approximation. Note that this density is a Gaussian density with mean $\boldsymbol{\theta}^*$ and covariance matrix $\mathrm{cov}(\boldsymbol{\theta}^*, \mathbf{y}_{1:\mathrm{T}})$ (Equation 4.27). The UM-step marginalizes the parameter uncertainty estimation using Equation 4.30 to estimate $\hat{\boldsymbol{\mu}}_{t|t}$ and $\hat{\boldsymbol{\Sigma}}_{t|t}$ at each time t. The LAP-P is summarized in Figure 4.10.

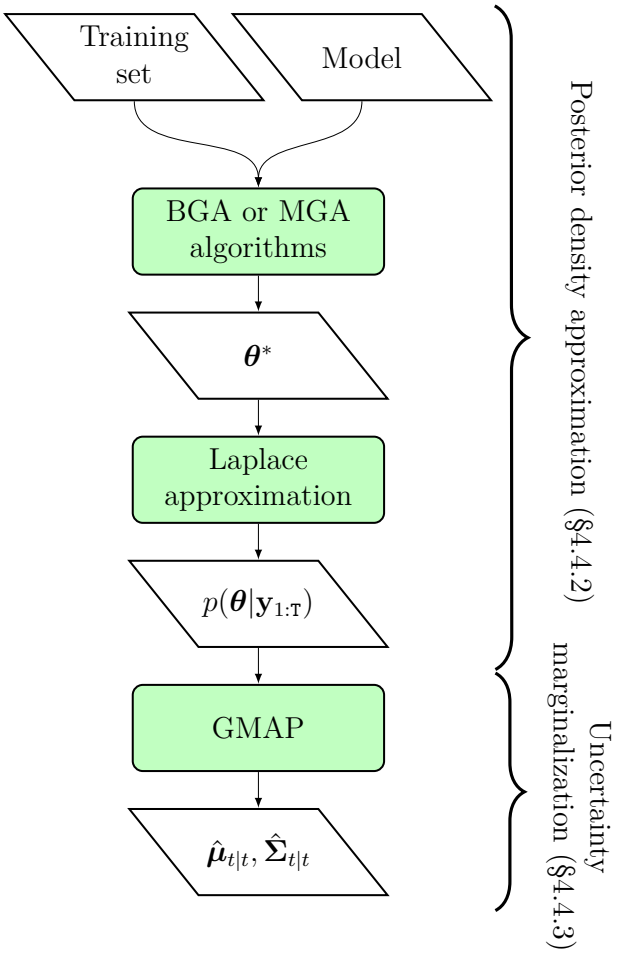


Figure 4.10 Illustration of the general procedure for approximating the posterior density of parameters $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathrm{T}})$ and the mean values of the hidden state variables and its covariance matrix at each time t, that is, $\{\hat{\boldsymbol{\mu}}_{t|t}, \hat{\boldsymbol{\Sigma}}_{t|t}\}$ using the combination of the Laplace approximation and Gaussian Mixture Approximation (GMAP). BGA: batch gradient ascent; MGA: mini-batch gradient ascent.

Hamiltonian Monte Carlo Procedure (HMC-P)

HMC-P has the same two steps as LAP-P, except that the PDA-step employs the HMC-based method presented in §4.4.1 for approximating the model parameter posterior density $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathtt{T}})$. The shematic architecture of a HMC-P is illustrated in Figure 4.11.

To ensure an efficient performance for the HMC-P, the parameters $\{\xi, \mathtt{L}_s\}$ for the leapfrog method presented in §4.4.1 need to be tuned in the PDA-step. For this purpose, the appropriate start point $\boldsymbol{\theta}^{\text{start}}$ for the parameters along with the HMC sampler are required. Once the leapfrog parameters are identified, the samples are then drawn from the constructed sampler for approximating $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathtt{T}})$.

4.5 Transformation of Model Parameter

In BDLMs, some model parameters are defined in bounded spaces where their values are restricted to specific intervals. For example, the model error standard deviation parameter σ in $\mathbf{Q}_t$ is a real positive number, that is, $\sigma \in (0, +\infty)$. During the optimization, a constraint needs to be defined for these model parameters in order to ensure that their values fall into their valid intervals. Yet, this constraint makes either the gradient ascent algorithms or the Bayesian estimation methods inefficient and slow [98]. For addressing this limitation, these model parameters are transformed into unbounded space, that is, $\theta \in (-\infty, +\infty)$. In BDLMs, the bounded model parameters are regrouped in two following categories,

$$\begin{aligned} &\text{category } \#1 \qquad \theta \in (0, +\infty) \\ &\text{category } \#2 \qquad \theta \in [a, b]\,, \end{aligned} \tag{4.32}$$

where a, b are the real numbers $\in \mathbb{R}$. The category #1 can be parameters such as the standard deviation, kernel lengthscale, and kernel period (§3.6.2). The category #2 represents, for example, the autocorrelation coefficient and the transition probability (§3.3.2), whose values lie in the interval $[0, 1]$. Either the base-10 logarithm ($\log_{10}$) or natural logarithm (ln) can be applied to the category #1 as a transformation function such that

$$\theta^{\mathtt{TR}} = \log_{10}(\theta) \quad \text{or} \quad \theta^{\mathtt{TR}} = \ln(\theta), \tag{4.33}$$

where $\theta^{\mathtt{TR}} \in (-\infty, +\infty)$ is the transformed model parameter. The reverse functions transforming the model parameter to the original space are given by

$$\theta = 10^{\theta^{\mathtt{TR}}} \quad \text{or} \quad \theta = \exp\left(\theta^{\mathtt{TR}}\right). \tag{4.34}$$

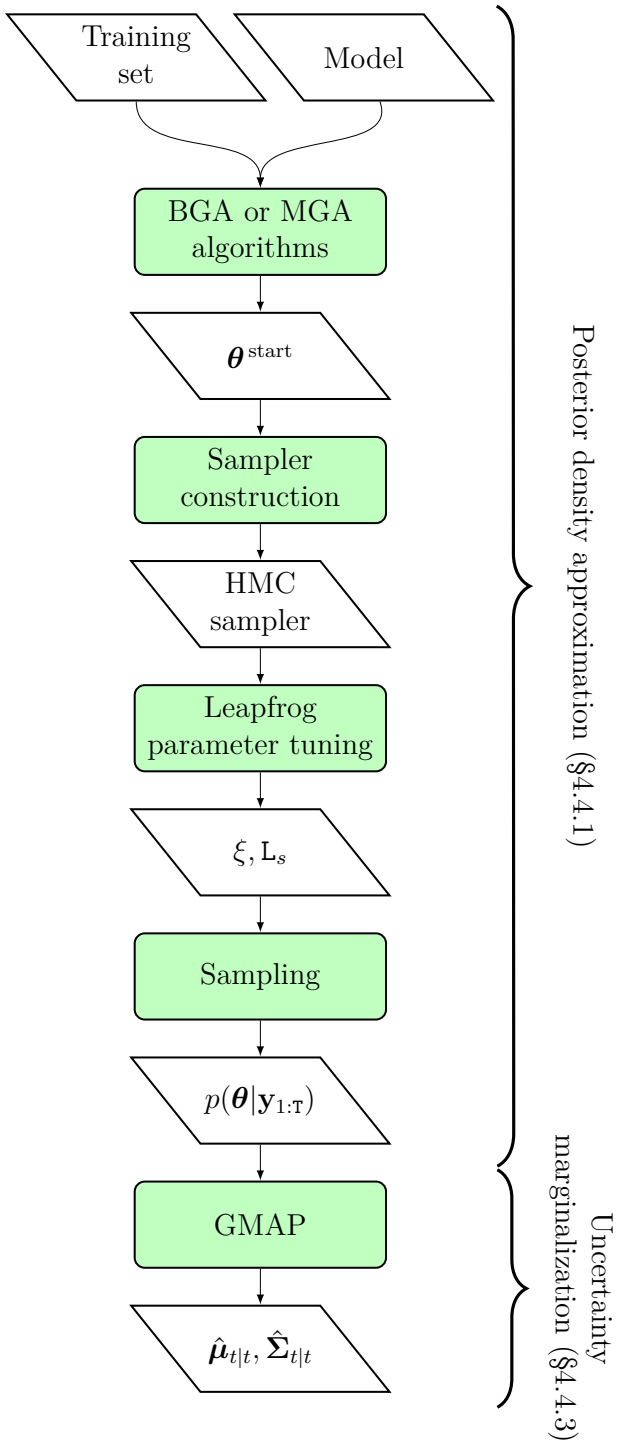


Figure 4.11 The two main steps for approximating the posterior density of parameters $p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathrm{T}})$ and the mean values of hidden state variables and its covariance matrix at each time t, that is, $\{\hat{\boldsymbol{\theta}}_{t|t}, \hat{\boldsymbol{\Sigma}}_{t|t}\}$ using the combination of the HMC method with Gaussian Mixture Approximation (GMAP). BGA: batch gradient ascent; MGA: mini-batch gradient ascent.

For the category #2, the logistic sigmoid function can be used as a transformation function, so that

$$\theta^{\mathtt{TR}} = -\ln\left(\frac{b-a}{\theta-a} - 1\right). \tag{4.35}$$

Its reverse function is defined following

$$\theta = \frac{b-a}{1+\exp(-\theta^{\mathtt{TR}})} + a. \tag{4.36}$$

Figure 4.12 presents three common transformation functions of model parameters in BDLMs. This example shows that the base-10 logarithm may provide a more intuitive interpretation

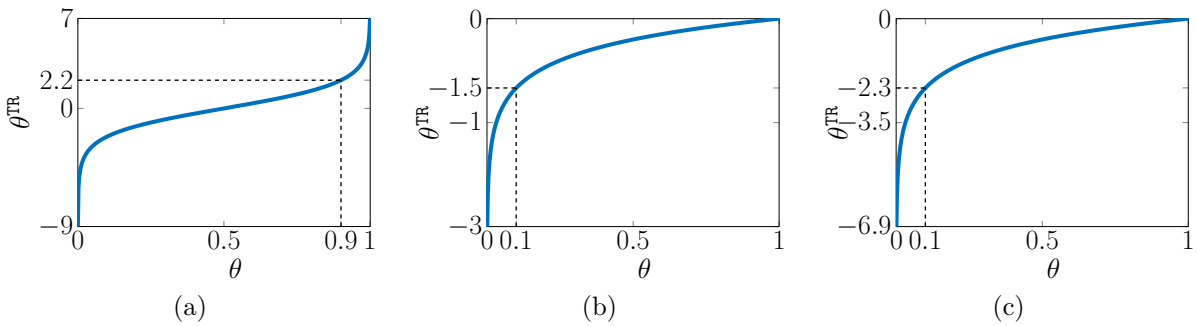


Figure 4.12 Illustration of transformation function of model parameters. (a) logistic sigmoid function with $a = 0$ and $b = 1$; (b) base-10 logarithm function; (c) nature logarithm function. TR stands for transformation.

than the natural logarithm in the case of the Bayesian estimation, where the prior PDFs of the model parameters of the category #1 are defined in the transformed space. This is because a value of -1 in the transformed space gives a value of 10^{-1} in the original space, as presented in Figure 4.12b.

4.6 Applications

This section presents three case studies. The first case study examines the performance of the gradient-based algorithms presented in Section 4.3 for optimizing the model parameters on a benchmark dataset. The second and third case studies compare the LAP with the HMC methods for approximating the posterior PDF of the model parameters and estimating the hidden state variables for both a benchmark and a real-world dataset.

4.6.1 Comparison of Optimization Algorithms

The goal of this experiment is to study the performance of the gradient-based optimization algorithms in finding the optimal model parameters. This case study is conducted on a simulated dataset where the optimized model parameters are compared with the true parameter values.

Simulated Data

The experiment is conducted on simulated data that are generated to be representative of the data recorded on a bridge. A temperature ($\mathtt{T}$) dataset is generated including a baseline ($\mathtt{B}$) component to represent the average temperature over time, the daily ($\mathtt{P}_1 = 1$ day) and seasonal ($\mathtt{P}_2 = 365.24$ days) sinusoidal cycles, an autoregressive ($\mathtt{AR}$) process to artificially introduce time-dependent model prediction errors, and observation errors. The mathematical formulation for the observation model describing this dataset is given by

$$y_t^{\mathtt{T}} = x_t^{\mathtt{B,T}} + x_t^{\mathtt{P_1,T}} + x_t^{\mathtt{P_2,T}} + x_t^{\mathtt{AR,T}} + v_t^{\mathtt{T}}, \tag{4.37}$$

where the formulation of each component is defined as

$$\begin{aligned}
x_t^{\mathtt{B}} &= 5\,^{\circ}C && && \text{average temperature} \\
x_t^{\mathtt{P_1,T}} &= 1 \cdot \sin\left[\tfrac{2\pi}{1}\left(t + \tfrac{8}{24}\right)\right] && && \text{daily cycle} \\
x_t^{\mathtt{P_2,T}} &= 20 \cdot \sin\left[\tfrac{2\pi}{365.24}\left(t + \tfrac{8}{12}\right)\right] && && \text{seasonal cycle} \\
x_t^{\mathtt{AR,T}} &= \underbrace{0.99}_{\phi^{\mathtt{AR,T}}} x_{t-1}^{\mathtt{AR,T}} + w_t^{\mathtt{AR,T}} && w_t^{\mathtt{AR,T}} \sim \mathcal{N}(0, \underbrace{0.5}_{\sigma^{\mathtt{AR,T}}}) && \text{AR(1) process} \\
v_t^{\mathtt{T}} &\sim \mathcal{N}(0, \underbrace{0.1}_{\sigma_v^{\mathtt{T}}}) && && \text{observation errors.}
\end{aligned} \tag{4.38}$$

A displacement dataset is generated by superposing the average displacement, the effect of temperature on the displacement, an autoregressive component, and observation errors. The mathematical formulation for the this dataset is defined as

$$y_t^{\mathtt{D}} = x^{\mathtt{B,D}} + x_t^{\mathtt{D|T,R}} + x_t^{\mathtt{AR,D}} + v_t^{\mathtt{D}}, \tag{4.39}$$

where the formulation of each component is defined as

$$
\begin{aligned}
x_t^{\mathtt{B,D}} &= 2\,[mm] && && \text{average displacement} \\
x_t^{\mathtt{D|T,R}} &= \underbrace{0.1}_{\beta_1^{\mathtt{R}}}\, x_t^{\mathtt{P_1,T}} - \underbrace{0.05}_{\beta_2^{\mathtt{R}}}\, x_t^{\mathtt{P_2,T}} - \underbrace{0.05}_{\beta_3^{\mathtt{R}}}\, x_t^{\mathtt{AR,T}} && && \text{temperature regression} \\
x_t^{\mathtt{AR,D}} &= \underbrace{0.98}_{\phi^{\mathtt{AR,D}}}\, x_{t-1}^{\mathtt{AR,D}} + w_t^{\mathtt{AR,D}} && w_t^{\mathtt{AR,D}} \sim \mathcal{N}(0, \underbrace{0.02}_{\sigma^{\mathtt{AR,D}}}) && \text{AR(1) process} \\
v_t^{\mathtt{D}} &\sim \mathcal{N}(0, \underbrace{0.05}_{\sigma_t^{\mathtt{D}}}) && && \text{observation errors.}
\end{aligned}
\tag{4.40}
$$

Both temperature and displacement datasets are generated with a uniform time-step length of 30 minutes. The total number of data points for each dataset is 52 608. Figure 4.13 presents the simulated data for the displacement and temperature.

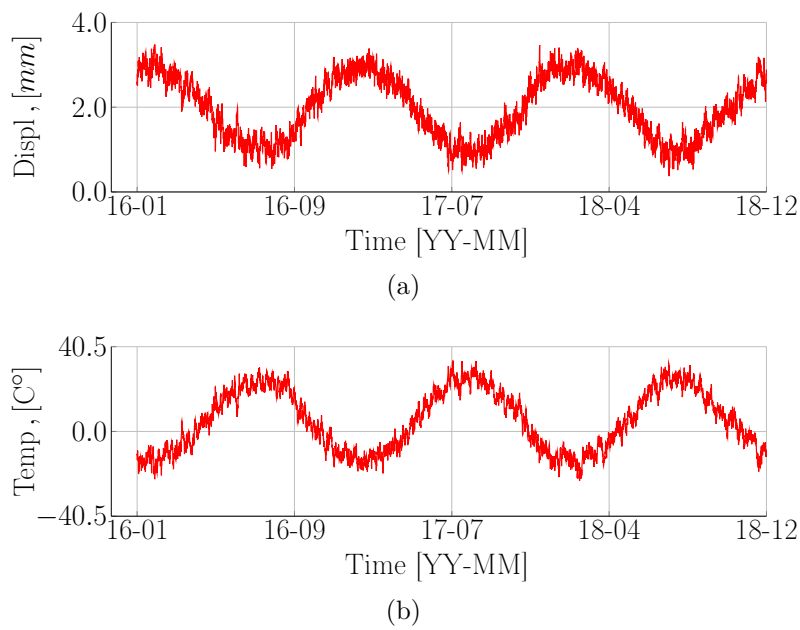


Figure 4.13 Illustration of the simulated datasets. (a) displacement; (b) temperature.

Architecture for Model #6

Model #6 is built (i) for the purpose of decomposing the displacement and temperature data into their hidden state variables and (ii) for describing the dependence of the displacement data on the temperature data. For this purpose, the vector of hidden state variables for the

temperature observations is defined as

$$\mathbf{x}_t^{\mathtt{T}} = \left[x^{\mathtt{B,T}}\ x^{\mathtt{F_1P_1,T}}\ x^{\mathtt{F_2P_1,T}}\ x^{\mathtt{F_1P_2,T}}\ x^{\mathtt{F_1P_2,T}}\ x^{\mathtt{AR,T}}\right]_t^{\mathtt{T}}, \tag{4.41}$$

where the baseline component $x_t^{\mathtt{B,T}}$ is used to model the average temperature, two Fourier-form components $[x^{\mathtt{F_1P_2,T}}\ x^{\mathtt{F_2P_2,T}}]_t^{\mathtt{T}}$ and $[x^{\mathtt{F_1P_1,T}}\ x^{\mathtt{F_2P_1,T}}]_t^{\mathtt{T}}$ are employed to describe the daily ($\mathtt{P_1} = 1$ day) and seasonal ($\mathtt{P_2} = 365.24$ days) cycles, and the autoregressive component $x_t^{\mathtt{AR,T}}$ is used to capture the time-dependent model errors. The displacement observations are decomposed into a vector of hidden state variables including a baseline component to model the average displacement and an autoregressive component to capture the time-dependent model errors. The vector of hidden state variables for the displacement data is given by

$$\mathbf{x}_t^{\mathtt{D}} = \left[x^{\mathtt{B,D}}\ x^{\mathtt{AR,D}}\right]_t^{\mathtt{T}}, \tag{4.42}$$

The dependency of the displacement on the temperature is described through the regression ($\mathtt{R}$) coefficients associated with the two Fourier-form components and the autoregressive component [77]. The vector of hidden state variables for Model #6 includes the vector of hidden state variables of the displacement and temperature data such that

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}^{\mathtt{D}} \\ \mathbf{x}^{\mathtt{T}} \end{bmatrix}_t. \tag{4.43}$$

Model #6 involves a vector of unknown model parameters that is given by

$$\boldsymbol{\theta} = \left[\phi^{\mathtt{AR,D}}\ \sigma^{\mathtt{AR,D}}\ \sigma_v^{\mathtt{D}}\ \beta_1^{\mathtt{R}}\ \beta_2^{\mathtt{R}}\ \beta_3^{\mathtt{R}}\ \phi^{\mathtt{AR,T}}\ \sigma^{\mathtt{AR,T}}\ \sigma_v^{\mathtt{T}}\right]^{\mathtt{T}}, \tag{4.44}$$

where $\phi^{\mathtt{AR,D}}$ is the autocorrelation coefficient for the displacement, $\sigma^{\mathtt{AR,D}}$ is autoregression standard deviation for the displacement, $\sigma_v^{\mathtt{D}}$ is the observation error standard deviation for the displacement, $\beta_1^{\mathtt{R}}$ is the regression coefficient for $x_t^{\mathtt{F_1P_1,T}}$, $\beta_2^{\mathtt{R}}$ is the regression coefficient for $x_t^{\mathtt{F_1P_2,T}}$, $\beta_3^{\mathtt{R}}$ is the regression coefficient for $x_t^{\mathtt{AR}}$, $\phi^{\mathtt{AR,T}}$ is the autocorrelation coefficient for the temperature, $\sigma^{\mathtt{AR,T}}$ is the autoregression standard deviation for the temperature, and $\sigma_v^{\mathtt{T}}$ is the observation error standard deviation for the temperature. These model parameters are learned from data using the MLE approach presented in Section 4.3. The optimization algorithms being used in the case study are BGA-NR and MGA-NR, ADAM, RMSProp, AMMT, and MMT, as presented in Chapter 4. The complete model matrices defining Model #6 are presented in Appendix A.6. Three runs with random initial values for the model parameters are conducted for each optimization algorithm. The maximal number of epochs

for each run is 30 and the number of data for a mini-batch is 3000 data points. The details of initial configurations are summarized in Table 4.1.

Table 4.1 Summary of the initial configurations for the model parameters.

No	Parameters	Bounds	Transformation functions	Random initial values			True values
				$\boldsymbol{\theta}_0^1$	$\boldsymbol{\theta}_0^2$	$\boldsymbol{\theta}_0^3$	
#1	$\phi^{\mathtt{AR,D}}$	$[0,1]$	logistic sigmoid	0.75	0.95	0.15	**0.98**
#2	$\sigma^{\mathtt{AR,D}}$	$(0,+\infty)$	natural logarithm	0.073	0.0095	0.015	**0.02**
#3	$\sigma_v^{\mathtt{D}}$	$(0,+\infty)$	natural logarithm	0.0073	0.013	0.13	**0.05**
#4	$\beta_1^{\mathtt{R}}$	$(-\infty,+\infty)$	-	0.01	-0.04	0.5	**0.1**
#5	$\beta_2^{\mathtt{R}}$	$(-\infty,+\infty)$	-	-0.1	0.1	0.01	$\mathbf{-0.05}$
#6	$\beta_3^{\mathtt{R}}$	$(-\infty,+\infty)$	-	0.01	0.1	0.1	$\mathbf{-0.05}$
#7	$\phi^{\mathtt{AR,T}}$	$[0,1]$	logistic sigmoid	0.75	0.5	0.85	**0.99**
#8	$\sigma^{\mathtt{AR,T}}$	$(0,+\infty)$	natural logarithm	1.433	0.05	0.085	**0.5**
#9	$\sigma_v^{\mathtt{T}}$	$(0,+\infty)$	natural logarithm	0.143	0.01	1	**0.1**

Optimization Results

Figure 4.14 presents the average log-likelihood evaluated for the validation set over three runs with respect to each optimization algorithm. The results show that the MGA- NR and AMMT outperforms the other algorithms, because the average validation log-likelihoods for BGA-NR and MGA-MMT are lower than for the other optimization algorithms. More specifically, it only took a few epochs for them to reach the validation log-likelihood value range that the remaining algorithms achieve in more than 10 epochs. During the 3rd run, MGA-NR suffered from numerical instabilities for a few epochs because of a high variability in the first and second derivatives. The MGA-AMMT yields a more stable performance than the MGA with NR. Note that both MGA-NR and AMMT do not require tuning the learning rate.

Table 4.2 presents the optimal vector of model parameters corresponding to the highest validation log-likehood value of three runs for each optimization algorithm. The vectors of model parameters obtained using the MGA-NR and AMMT are the closest to the true parameter values. Their corresponding validation log-likelihood values are higher than those obtained using the remaining optimization algorithms. The training time required for each

Table 4.2 Optimal vectors of model parameters for each optimization algorithm. BGA: batch gradient ascent; MGA: mini-batch gradient ascent; NR: Newton-Raphson; MMT: momentum; RMSProp: root mean square propagation; ADAM: adaptive moment estimation; AMMT: adaptive momentum; CPU: central processing unit.

No	Parameters	Optimization algorithm						True values
		BGA NR	MGA NR	MGA MMT	MGA RMSProp	MGA ADAM	MGA AMMT	
#1	$\phi^{\mathtt{AR,D}}$	0.962	0.978	0.771	0.980	0.981	0.981	**0.98**
#2	$\sigma^{\mathtt{AR,D}}$	0.04	0.02	0.07	0.02	0.02	0.02	**0.02**
#3	$\sigma_v^{\mathtt{D}}$	0.05	0.05	0.013	0.007	0.05	0.05	**0.05**
#4	$\beta_1^{\mathtt{R}}$	0.10	0.11	0.92	0.11	0.12	0.10	**0.10**
#5	$\beta_2^{\mathtt{R}}$	−0.07	−0.05	−0.05	−0.07	−0.05	−0.05	**−0.05**
#6	$\beta_3^{\mathtt{R}}$	−0.07	−0.05	−0.05	−0.04	−0.05	−0.05	**−0.05**
#7	$\phi^{\mathtt{AR,T}}$	0.989	0.989	0.94	0.995	0.993	0.991	**0.99**
#8	$\sigma^{\mathtt{AR,T}}$	0.87	0.49	0.53	0.49	0.49	0.49	**0.50**
#9	$\sigma_v^{\mathtt{T}}$	0.01	0.10	0.13	0.08	0.11	0.10	**0.10**
L_v		2889.7	<u>6471.1</u>	4292.9	6352.6	6443.5	<u>6470.6</u>	–
Training time (mins)		17.3	17.6	17.6	17.5	17.7	17.7	–
CPU (cores)		1	10	10	10	10	10	–

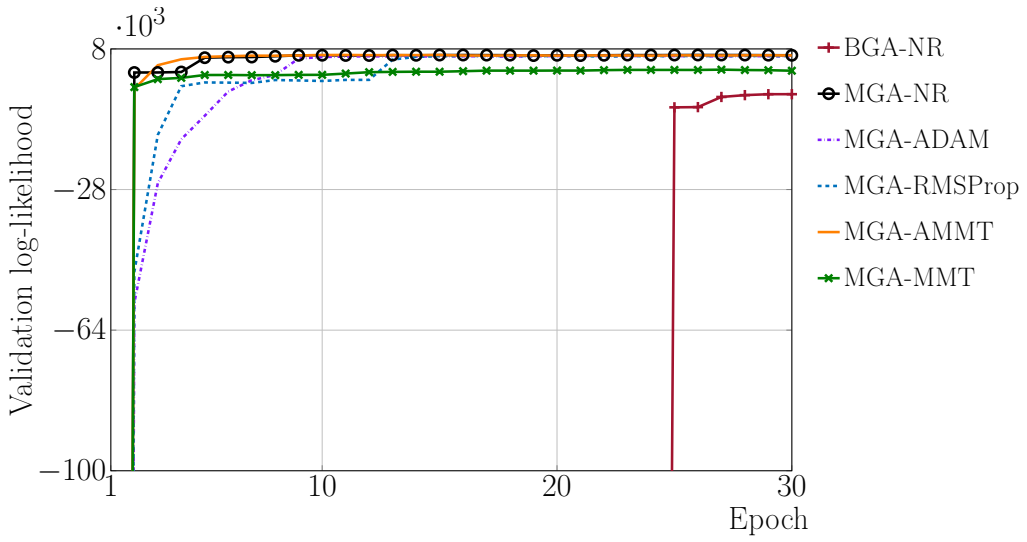


Figure 4.14 Illustration of the average validation-log-likelihood over three runs with respect to the optimization algorithm. BGA: batch gradient ascent; MGA: mini-batch gradient ascent; NR: Newton-Raphson; ADAM: adaptive moment estimation; RMSProp: root mean square propagation; MMT: momentum; AMMT: adaptive momentum.

optimization algorithm is approximated to 18 minutes, yet the BGA only uses one Central Processing Unit (CPU) core during the optimization while the MGA employs 10 CPU cores. This difference in the CPU usage is because the BGA can only perform a single possible model parameter update at each epoch, so it cannot leverage the parallelism of multiple CPUs. On the other hand, the MGA algorithm needs to evaluate multiple independent possibilities for updating the model parameters, as presented in §4.3.5 so that it takes advantage of the parallel computation of multiple CPU cores in order to reduce the computational time. Note that as shown in Figure 4.14, the BGA might need more than 30 epochs in order to achieve the same performance as the MGA, which requires more computational time. MGA algorithms except the MGA-MMT could have been stopped at the 15th epoch where the validation log-likelihood has reached a stationary stage, that is, the *learning plateau*. This would allow reducing the training time by a factor of two. Overall, this case study shows that the MGA algorithm is computationally more efficient than the BGA algorithms when dealing with a large dataset.

4.6.2 Comparison of LAP-P with HMC-P on a Simulated Dataset

The objective here is to compare the performance of the LAP-P with the HMC-P method using a simulated dataset for which the true values for the hidden state variables and the

model parameters are known.

Data Description

The experiment is conducted on simulated data that are generated to be representative of the data recorded on civil infrastructure such as a dam. For this purpose, a dataset of displacement measurements is generated including a baseline (B) to represent the structural behaviour over time, a seasonal sinusoidal cycle (P) to describe the thermal effect of environmental conditions on the displacement, an autoregressive (AR) process to artificially introduce time-dependent model prediction errors, and observation errors (v_t). The observation model is formulated following

$$y_t = x_t^{\mathtt{B}} + x_t^{\mathtt{P}} + x_t^{\mathtt{AR}} + v_t, \tag{4.45}$$

where each component is generated using the following formulation

$$\begin{aligned} x_t^{\mathtt{B}} &= 3 + w_t^{\mathtt{B}}, & w_t^{\mathtt{B}} &\sim \mathcal{N}(0, (\underbrace{10^{-5}}_{\sigma_w^{\mathtt{B}}})^2) \\ x_t^{\mathtt{P}} &= 4\sin\left[\tfrac{2\pi}{365.24}\cdot(t+15)\right] \\ x_t^{\mathtt{AR}} &= \underbrace{0.866}_{\phi^{\mathtt{AR}}}\cdot x_{t-1}^{\mathtt{AR}} + w_t^{\mathtt{AR}}, & w_t^{\mathtt{AR}} &\sim \mathcal{N}(0, (\underbrace{0.05}_{\sigma_w^{\mathtt{AR}}})^2) \\ v_t &\sim \mathcal{N}(0, (\underbrace{0.1}_{\sigma_v})^2). \end{aligned} \tag{4.46}$$

The four-year dataset (1461 observations) with a uniform time-step length of 24 hours is shown in Figure 4.15. For this case study, five tests on different training-set lengths (TSLs)

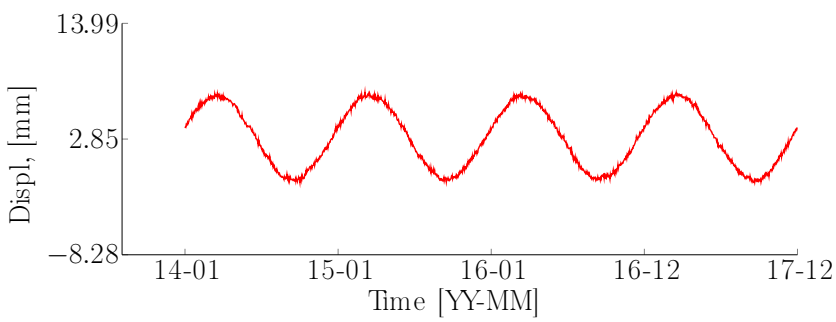


Figure 4.15 Illustration of 4 years of simulated data.

are conducted for approximating the posterior PDF of model parameters as well as for the estimation of the hidden state variables. The amount of data with respect to TSL is presented in Table 4.3.

Table 4.3 The amount of data points with respect to the training set length (TSL) for the simulated dataset.

TSL (days)	Data (points)
30	30
90	90
180	180
365	365
1095	1095

Architecture of Model #7

Model #7 is built for this case study. In this model, each observation is decomposed into a baseline (B) component to model the structural behaviour over time, a Fourier-form component (§3.2.2) with a period of $\mathtt{P} = 365.24$ days to model the environmental conditions, and an autoregressive (AR) component to describe model prediction errors. Hence, the vector of hidden state variables is defined as

$$\mathbf{x}_t = \left[x^{\mathtt{B}}\ x^{\mathtt{F1}}\ x^{\mathtt{F2}}\ x^{\mathtt{AR}}\right]_t^{\mathsf{T}}. \tag{4.47}$$

The model involves a vector of unknown parameters that are defined following

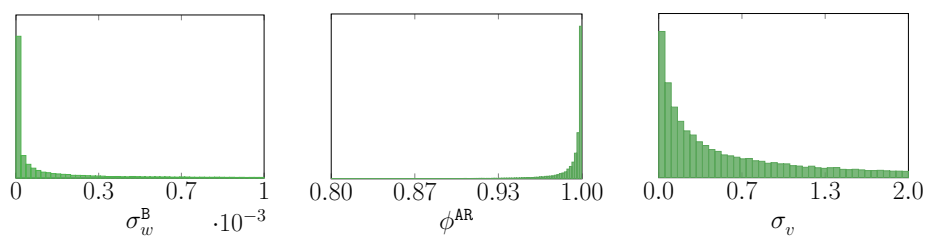
$$\boldsymbol{\theta} = \left[\sigma_w^{\mathtt{B}}\ \phi^{\mathtt{AR}}\ \sigma_w^{\mathtt{AR}}\ \sigma_v\right]^{\mathsf{T}}, \tag{4.48}$$

where $\sigma_w^{\mathtt{B}}$ is the baseline standard deviation, $\phi^{\mathtt{AR}}$ is the autocorrelation coefficient, $\sigma_w^{\mathtt{AR}}$ is the autoregression standard deviation, and σ_v is the observation error standard deviation. $\sigma_w^{\mathtt{B}}, \sigma_w^{\mathtt{AR}}$, and σ_v are positive real numbers $\in \mathbb{R}^+$, whereas $\phi^{\mathtt{AR}}$ is defined in a range between 0 and 1. The full model matrices can be found in Appendix A.7. The parameters being estimated are transformed into unbounded spaces for an efficient estimation (see Section 4.5). For this purpose, the base-10 logarithm and logistic sigmoid functions are applied to the standard deviations and the autocorrelation coefficient. The initial parameter values in the original space for the model are

$$\boldsymbol{\theta}_0 = \left[\underbrace{10^{-4}}_{\sigma_w^{\mathtt{B}}}\ \underbrace{0.7}_{\phi^{\mathtt{AR}}}\ \underbrace{0.01}_{\sigma_w^{\mathtt{AR}}}\ \underbrace{0.026}_{\sigma_v}\right]^{\mathsf{T}}. \tag{4.49}$$

In the BDLM framework, the baseline standard deviation $\sigma_w^{\mathtt{B}}$ is expected to be small because

the error between the model prediction and the observation at each time step will be captured in the autoregressive component $\mathbf{x}^{\mathtt{AR}}$. Hence, it yields high autocorrelations in the model prediction errors between time steps. The autocorrelation coefficient $\phi^{\mathtt{AR}}$ is assumed to be close to one. This prior knowledge defines the prior PDF for $\sigma_w^{\mathtt{B}}$ and $\phi^{\mathtt{AR}}$ in order to ensure a reliable estimation [118, 119]. The remaining model parameters are expected to be close to one. The prior distributions in the transformed space ($\mathtt{TR}$) associated with each parameter in Equation 4.48 are set as follows: $f(\sigma_w^{\mathtt{B,TR}}) = \mathcal{N}(-4, 2^2)$, $f(\phi^{\mathtt{AR,TR}}) = \mathcal{N}(1.5, 0.5^2)$, $f(\sigma_w^{\mathtt{AR,TR}}) = \mathcal{N}(0, 1^2)$, and $f(\sigma_v^{\mathtt{TR}}) = \mathcal{N}(0, 1^2)$. Figure 4.16 illustrates the different prior distributions corresponding to three parameters $\{\sigma_w^{\mathtt{B}}, \phi^{\mathtt{AR}}, \sigma_v\}$ represented in the original space. Because the LAP-P and



(a) $\sigma_w^{\mathtt{B}}$ (b) $\phi^{\mathtt{AR}}$ (c) σ_v

Figure 4.16 Illustration of prior distribution choices for model parameters in the original space.

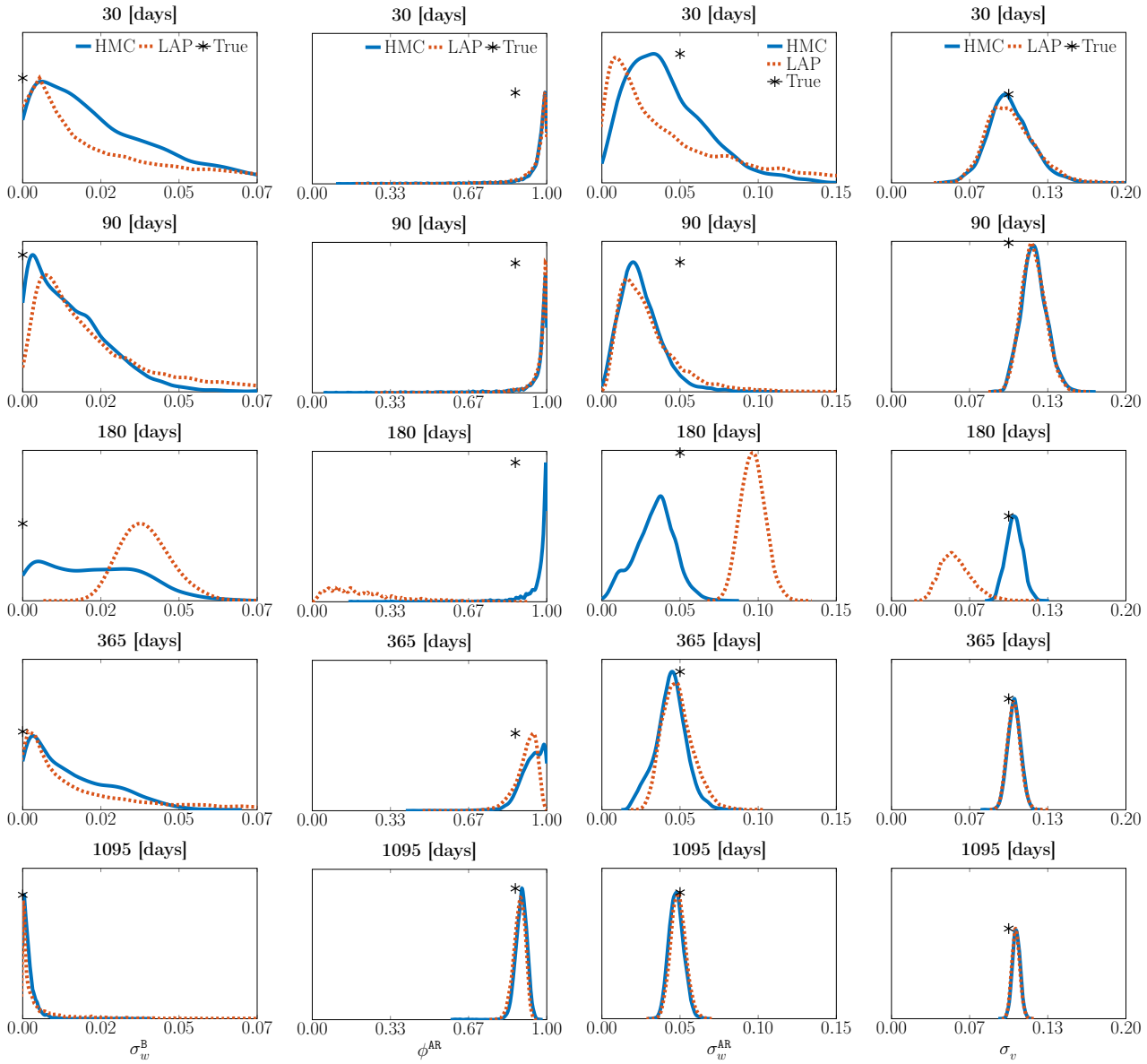
HMC-P are different from each other, some setup options for the parameter estimation must be tuned separately. In the LAP-P, the initial parameter values in the original space are equal to $\boldsymbol{\theta}^0$. The HMC-P employs 4 parallel Markov-chains ($\mathtt{C}_c$) for each training-set length, where each Markov chain has its own initial parameter values. These initial parameter values are directly defined in the transformed space as follows

$$\begin{aligned} \boldsymbol{\theta}_0^{\mathtt{C}_1} &= \boldsymbol{\theta}^{\text{start}} \\ \left\{\boldsymbol{\theta}_0^{\mathtt{C}_2}, \boldsymbol{\theta}_0^{\mathtt{C}_3}, \boldsymbol{\theta}_0^{\mathtt{C}_4}\right\} &= \mathcal{N}\left(\boldsymbol{\theta}^{\text{start}}, \text{diag}([1\ 1\ 1]^{\mathtt{T}})\right), \end{aligned} \tag{4.50}$$

where $\boldsymbol{\theta}^{\text{start}}$ is a vector of pre-estimated parameters as presented in §4.4.4. The mass matrix $\mathcal{M}$ in Equation 4.23 is chosen as the diagonal Hessian matrix of $-\ln p(\boldsymbol{\theta}|\mathbf{y}_{1:\mathtt{T}})$. Here, the stopping criterion is when the EPSR metric $\hat{\mathbf{R}}$ is less than 1.01 (see §4.4.1).

Results

Figure 4.17 shows the kernel density estimate [120] of the posterior PDFs for each model parameter according to each TSL. The dashed line and the solid line represent respectively



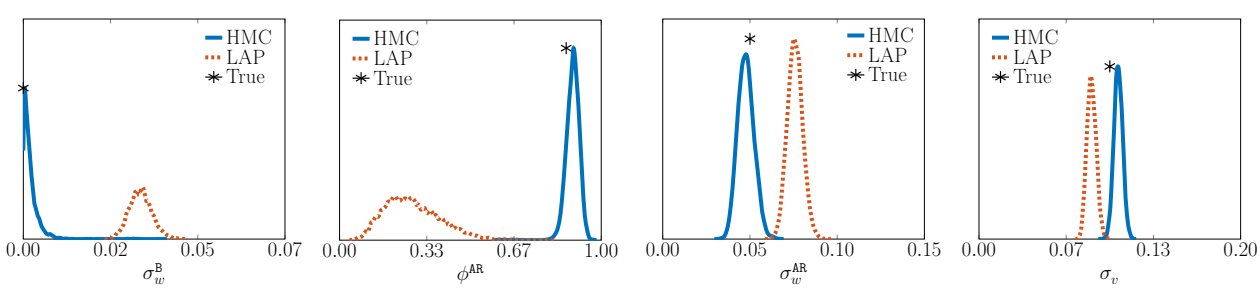
(a) $\sigma_w^{\mathtt{B}}$ (b) $\phi^{\mathtt{AR}}$ (c) $\sigma_w^{\mathtt{AR}}$ (d) σ_v

Figure 4.17 Each column illustrates the kernel density estimate of the posterior PDFs for each model parameter $p(\theta_i|\mathbf{y}_{1:\mathtt{T}})$ in the original space with respect to the training-set length of simulated data.

the PDFs obtained using LAP-P and HMC-P. The *true parameter values* ($\check{\boldsymbol{\theta}}$) are presented by the asterisks.

The results show that the posterior PDFs obtained using both procedures concentrate around $\check{\boldsymbol{\theta}}$ as the training dataset length increases. However, for the TSL of 180 days, the PDFs approximated using LAP-P are shifted away from those approximated using HMC-P and $\check{\boldsymbol{\theta}}$. This behaviour can be explained by the sensitivity toward the initial parameter values in LAP-P, leading to a local maximum. Furthermore, Figure 4.17a shows that the posterior PDF for $\sigma_w^{\mathtt{B}}$ approximated using HMC-P has a larger posterior mass in the tail than the other shorter TSLs. This behaviour is contrary to the general intuition that the more data are available, the more posterior mass concentrates around $\check{\boldsymbol{\theta}}$. This behaviour justifies that the extraction of information from data depends not only on the size of data, but also on the interaction of the prior with the likelihood function and the model parameter being estimated as noted by Gelman et al [119]. In the TSL of 1095 days, the posterior PDFs for model parameters obtained with both procedures are identical, yet the posterior PDF for $\sigma_w^{\mathtt{B}}$ obtained using LAP-P exhibits heavy tails. Overall, HMC-P shows a superior capacity at approximating the posterior PDF for model parameters over LAP-P for less than a year TSLs.

Biased estimation with LAP-P for the TLS of 180 days leads to the question of the robustness of LAP-P with respect to the choice of initial parameter values. To answer this question, an additional test using a different set of initial parameter values, is carried out with the TLS of 1095 days. Figure 4.18 presents the kernel density estimate of the posterior PDFs for each model parameter. Instead of concentrating around $\check{\boldsymbol{\theta}}$, the PDFs obtained using



(a) $\sigma_w^{\mathtt{B}}$ (b) $\phi^{\mathtt{AR}}$ (c) $\sigma_w^{\mathtt{AR}}$ (d) σ_v

Figure 4.18 Illustration of the kernel density estimate of the posterior PDFs for each model parameter in the original space with the training set of 1095 days.

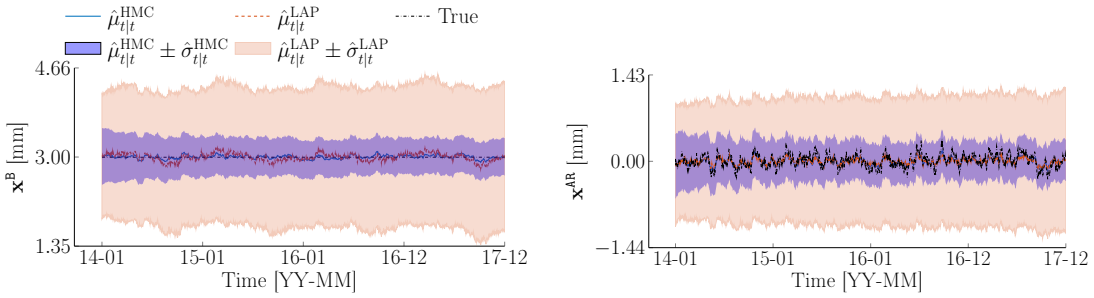
LAP-P are far from them. Note that in the previous test, the LAP-P performed well for the approximation of the posterior PDFs of model parameters, where these PDFs concentrated around $\check{\boldsymbol{\theta}}$ (see Figure 4.17). It illustrates the effect of poor initial parameter values on the

approximation of the posterior PDF for model parameters. Hence, a careful tuning of the initial parameter values in LAP-P is essential for an accurate estimation. For this purpose, different sets of initial parameter values should be tested during training.

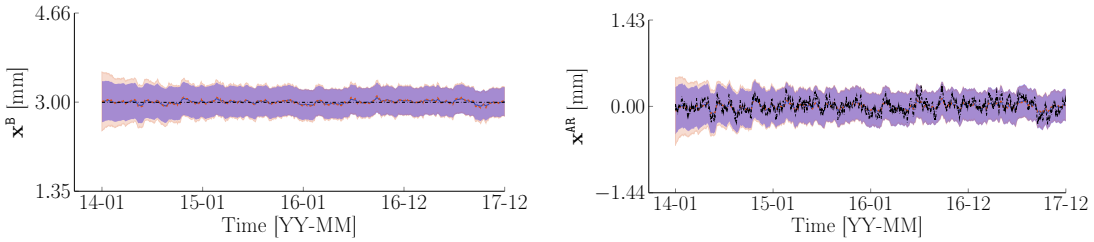
For both procedures, the hidden state variables are estimated using 1000 samples from the joint parameter posterior PDF. This number of samples provides a sufficient accuracy for estimating the hidden state variables, because the same results are found for the larger training set length. Figure 4.19 presents the hidden state variables estimated using the Kalman smoother [28] for the different training set lengths. Only the baseline ($\mathbf{x}^{\mathtt{B}}$) and autoregressive ($\mathbf{x}^{\mathtt{AR}}$) components are presented for this case study. The mean values and its standard deviation at time t are $\mu_{t|t}^{(\cdot)}$ and $\sigma_{t|t}^{(\cdot)}$, where the superscript $(\cdot)$ is associated with either LAP-P or HMC-P being employed for the estimation task. The expected value $\hat{\mu}_{t|t}^{\mathrm{LAP}}$ and its uncertainty bound $\hat{\mu}_{t|t}^{\mathrm{LAP}} \pm \hat{\sigma}_{t|t}^{\mathrm{LAP}}$ at time t are represented by the dashed line and the shaded region delimited by the solid line. Meanwhile, $\hat{\mu}_{t|t}^{\mathrm{HMC}}$ and $\hat{\mu}_{t|t}^{\mathrm{HMC}} \pm \hat{\sigma}_{t|t}^{\mathrm{HMC}}$ are represented by the solid line and the shaded region. The true hidden state variables ($\check{\mathbf{x}}_t$) are presented by the dash-dot line.

Like the posterior PDFs for model parameters, the estimation of the hidden state variables keeps improving as the amount of the training data increases. The mean values for the hidden state variables tend to $\check{\mathbf{x}}_t$ and their uncertainty bounds narrow down. More importantly, these uncertainty bounds include with $\check{\mathbf{x}}_t$ in almost all TLSs, except 180 days. Figure 4.19c shows that the estimation of the hidden state variables using LAP-P for the TSL of 180 days is affected by the biased posterior PDF. The baseline and autoregressive components obtained using LAP-P are not well separated even though their uncertainty bounds are smaller than those obtained using HMC-P. Meanwhile, the uncertainty bounds for the baseline and autoregressive components estimated using HMC-P are larger than those from the shorter TSL of 90 days because of a poor approximation of the posterior PDF for model parameters.

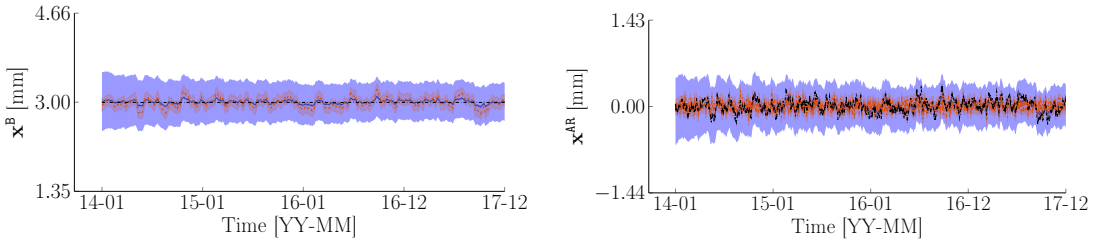
The uncertainty bounds of the baseline component obtained using LAP-P are larger than those obtained from HMC-P in almost all TLSs excepted for 180 days. The discrepancy between both procedures is clearly observable in the TSLs of 30 and 1095 days, as illustrated in Figures 4.19a and e. The heavy tailed posterior PDF for $\sigma_w^{\mathtt{B}}$ obtained using LAP-P (Figure 4.17a) is the cause for this behaviour. For the TSL of 365 days, the uncertainty bounds of the autoregressive component estimated using LAP-P are unexpectedly smaller than those estimated using HMC-P. This is explained by a more precise approximation with LAP-P than with HMC-P for the posterior PDFs for $\phi^{\mathtt{AR}}$ and $\sigma_w^{\mathtt{AR}}$ (see Figures 4.17b and c). Figure 4.20 presents the computational time of both procedures for approximating the posterior PDF for this case study. It shows that the HMC-P requires significantly more computational resources



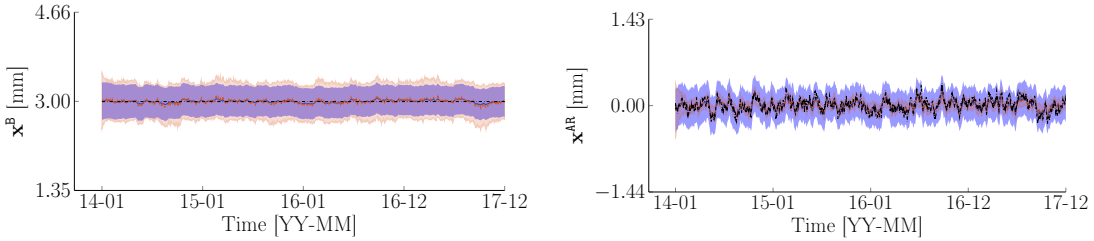
(a) Training set of 30 days (30 data points)



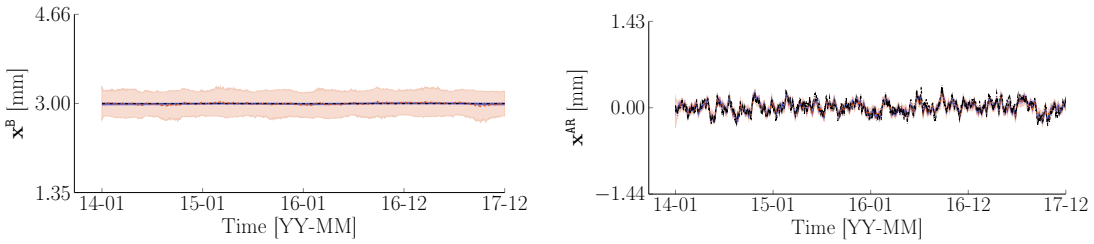
(b) Training set of 90 days (90 data points)



(c) Training set of 180 days (180 data points)



(d) Training set of 365 days (365 data points)



(e) Training set of 1095 days (1095 data points)

Figure 4.19 Expected value $\hat{\mu}$ and standard deviation $\hat{\sigma}$ for baseline (left) and autoregressive (right) components using the Laplace approximation procedure (LAP-P) and the Hamiltonian Monte Carlo procedure (HMC-P) with respect to the training-set length of the simulated data.

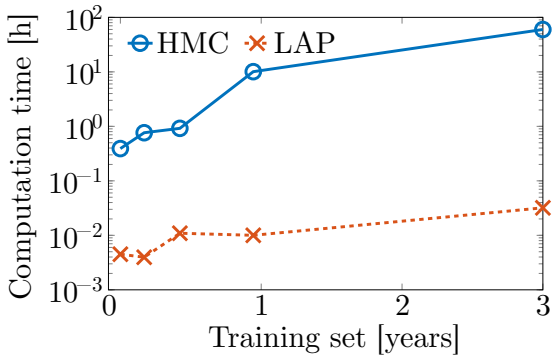


Figure 4.20 Computation time of LAP-P and HMC-P for approximating the parameter's posterior PDF in the simulated dataset. LAP: Laplace Approximation; HMC: Hamiltonian Monte Carlo.

than the LAP-P.

These results illustrate a potential impact of the model parameter uncertainty on the estimation of the hidden state variables. Also, the mean values and their uncertainty bounds for the hidden state variables obtained using HMC-P are more reliable than those obtained using LAP-P because of the lack of sensitivity with respect to the initial parameter values in LAP-P.

4.6.3 Comparison of LAP-P with HMC-P on a Real Dataset

In this case study, the comparison between the LAP-P and the HMC-P method is illustrated using the horizontal displacement data collected on a dam located in Canada.

Data Description

The horizontal displacement data along the X-direction (see Figure 3.10) are collected over the period of 4 years from 2010 to 2014 with a total of 2679 data points. The entire dataset is shown in Figure 4.15. A descending trend and a periodic pattern with a period of one year can be observed from the raw data. The periodic pattern reaches its maximum during winter and minimum during summer. Such a behaviour is attributed to the effect of temperature. Figure 4.22 shows that the data are collected with a non-uniform time-step length. The time-step length varies in the range from 1 to 216 hours, where the most frequent time-step is 12 hours. A reference time-step length [77] corresponding to the most frequent time-step according to the studied training-set is selected. As with the case involving simulated data, five tests conducted on different TSLs have been carried out using both the LAP-P and HMC-P for

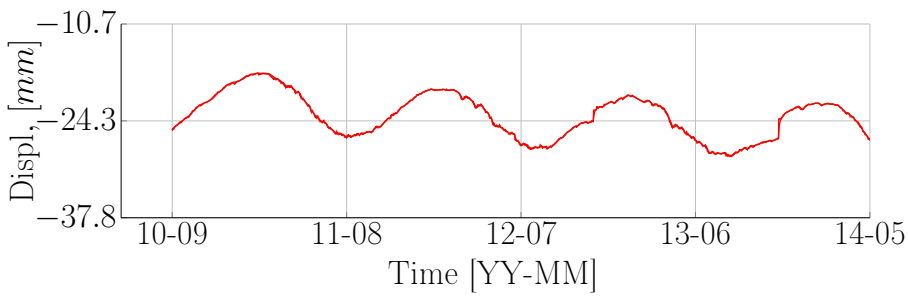


Figure 4.21 Raw displacement data.

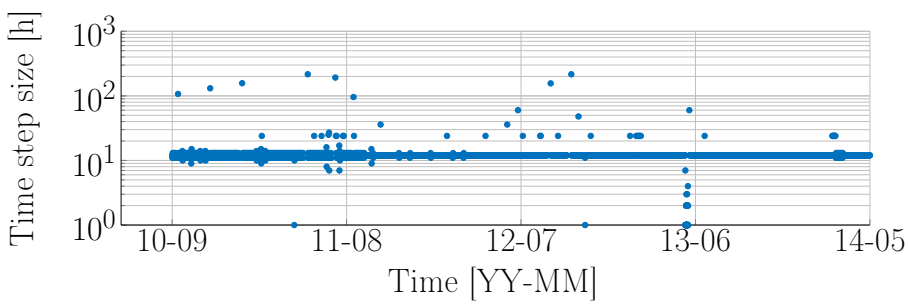


Figure 4.22 Time step size

this case study. The amount of data with respect to TSL is presented in Table 4.4.

Architecture for Model #8

Similarly to Model #7 detailed in §4.6.2, Model #8 is constructed using the same vector of hidden state variables, with an additional local trend (LT). Because of the descending trend behaviour observed from the raw data (see Figure 4.21), the local trend is needed to model the rate of change in the baseline component. Therefore, the vector of hidden state variables is written as

$$\mathbf{x}_t = \left[x^{\mathtt{B}}\ x^{\mathtt{LT}}\ x^{\mathtt{F_1,P}}\ x^{\mathtt{F_2,P}}\ x^{\mathtt{AR}}\right]_t^{\mathsf{T}}. \tag{4.51}$$

The parameter vector $\boldsymbol{\theta}$ corresponding to the model is defined following

$$\boldsymbol{\theta} = \left[\sigma_w^{\mathtt{LT}}\ \phi^{\mathtt{AR}}\ \sigma_w^{\mathtt{AR}}\ \sigma_v\right]^{\mathsf{T}}, \tag{4.52}$$

where $\sigma_w^{\mathtt{LT}}$ is the local trend standard deviation and the remaining parameters are the same as defined in Equation 4.48. The complete matrices for the Model #8 are detailed in Appendix

Table 4.4 The amount of data points with respect to the training set length (TSL) for the real dataset.

TSL (days)	Data (points)
30	51
90	161
180	328
365	651
1095	2142

A.8. The initial parameter values in the original space for the model are

$$\boldsymbol{\theta}_0 = \left[\underbrace{10^{-4}}_{\sigma_w^{\mathtt{LT}}} \; \underbrace{0.8}_{\phi^{\mathtt{AR}}} \; \underbrace{0.02}_{\sigma_w^{\mathtt{AR}}} \; \underbrace{0.03}_{\sigma_v}\right]^{\mathsf{T}}. \tag{4.53}$$

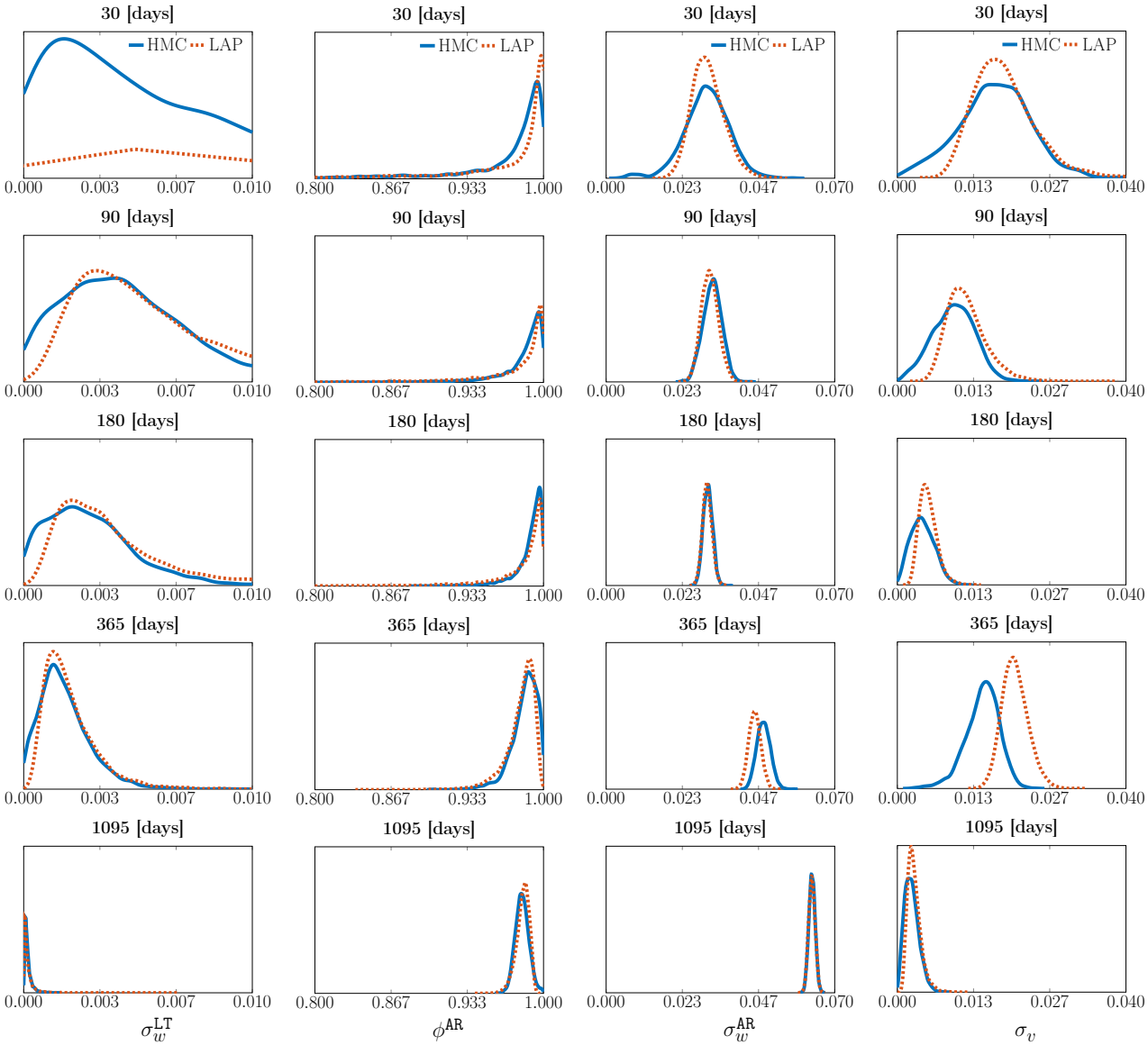
The other settings such as the transformation functions and the prior PDFs related to the LAP-P and HMC-P remain identical as for Model #7.

Results

Figures 4.23 and 4.24 show the posterior PDFs for each model parameter as well as the hidden state variables estimated using Kalman smoother. The convention for these figures remains identical as for the simulated case presented in §4.6.2. The dashed and solid lines represent the kernel density estimate for the parameter's posterior PDF obtained using the LAP-P and HMC-P. For hidden state variables, the dashed line and the shaded region delimited by the solid line represent the mean values $\hat{\mu}_{t|t}^{\mathrm{LAP}}$ and its uncertainty bounds $\hat{\mu}_{t|t}^{\mathrm{LAP}} \pm \hat{\sigma}_{t|t}^{\mathrm{LAP}}$ at time t, whereas $\hat{\mu}_{t|t}^{\mathrm{HMC}}$ and $\hat{\mu}_{t|t}^{\mathrm{HMC}} \pm \hat{\sigma}_{t|t}^{\mathrm{HMC}}$ are illustrated by the solid line and the shaded region.

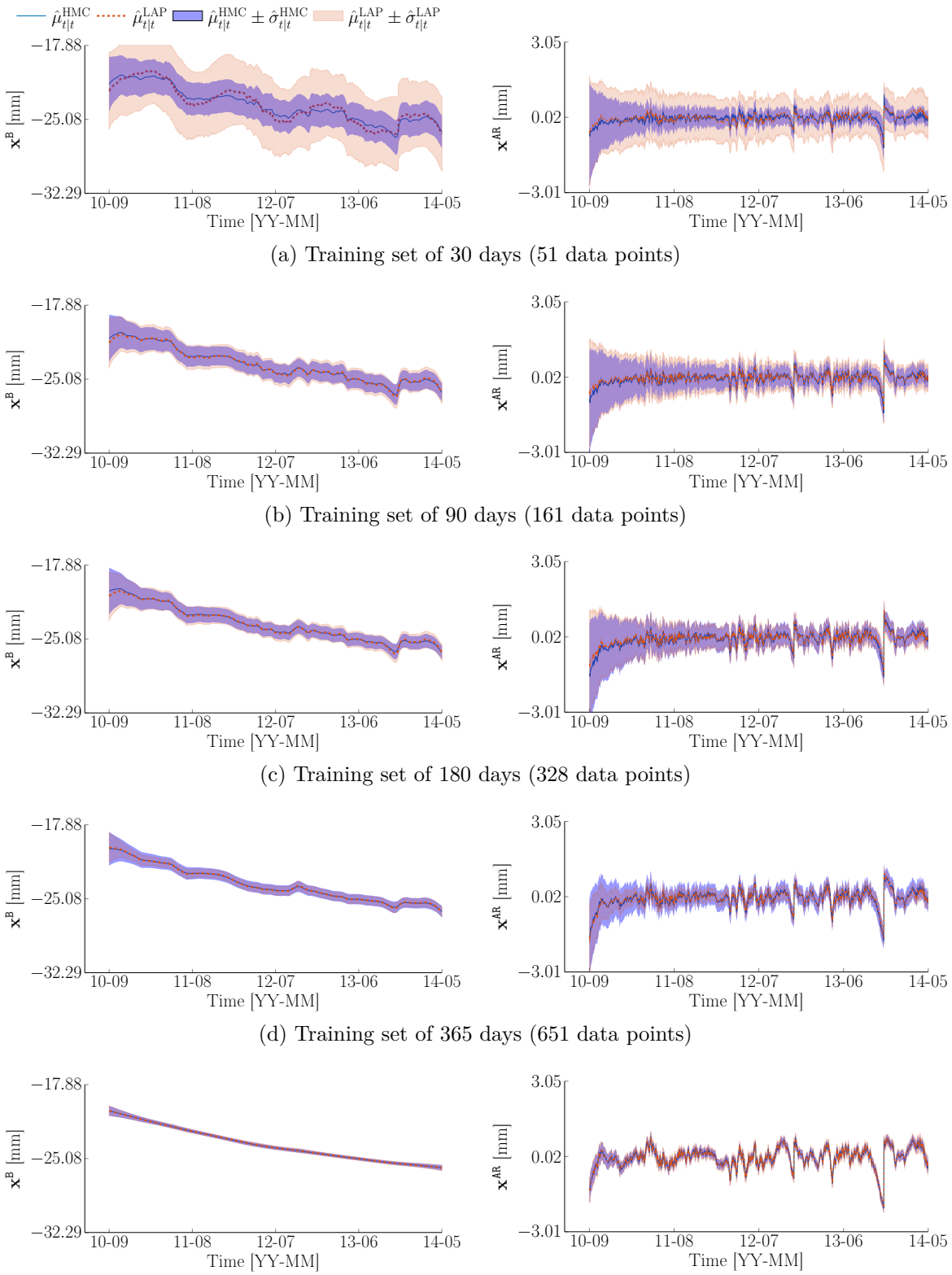
The results show that the posterior PDF for $\sigma_w^{\mathtt{LT}}$ and $\sigma_w^{\mathtt{AR}}$ approximated using both the LAP-P and HMC-P converges to the same PDF when the size of the training data increases. However, it is not the case for the posterior PDFs for $\sigma_w^{\mathtt{AR}}$ and σ_v, for which there is a lack of consistency between the posterior PDFs regarding the TSL. More specifically, Figure 4.23c and d show that the expected values of these posterior PDFs slightly change with respect to the TLS.

The estimation accuracy of hidden state variables obtained using both the LAP-P and HMC-P improves with the dataset size. Their expected values tend to the same values and



(a) $\sigma_w^{\mathtt{LT}}$ (b) $\phi^{\mathtt{AR}}$ (c) $\sigma_w^{\mathtt{AR}}$ (d) σ_v

Figure 4.23 Illustration of the kernel density estimate of the posterior PDFs for each model parameter $p(\theta_i|\mathbf{y}_{1:\mathtt{T}})$ in the original space with respect to the training-set length. The data are collected on a dam in Canada. HMC: Hamiltonian Monte Carlo; LAP: Laplace Approximation.



(a) Training set of 30 days (51 data points)

(b) Training set of 90 days (161 data points)

(c) Training set of 180 days (328 data points)

(d) Training set of 365 days (651 data points)

(e) Training set of 1095 days (2142 data points)

Figure 4.24 Expected value $\hat{\mu}$ and standard deviation $\hat{\sigma}$ for baseline (left) and autoregressive (right) components using Laplace approximation procedure (LAP-P) and Hamiltonian Monte Carlo procedure (HMC-P) with respect to the training-set length. The data are collected on a dam in Canada.

their uncertainty bounds are reduced as the TSL increases. For the TSL of 30 and 90 days, the uncertainties for the hidden state variables obtained using HMC-P are smaller than those obtained using LAP-P. The estimation results for the TSL of 1095 days presented in Figure 4.24e outperform the others in the remaining TSLs. It confirms that the uncertainties of both the model parameters and state estimates can be reduced through an increase in the dataset size. The autoregressive component $\mathbf{x}^{\mathtt{AR}}$ shows a stationary behaviour with a small amplitude even though an abnormal peak with a high amplitude is identified at the end of the year 2013. This jump is likely to have been caused by the presence of a malfunction in the measurement device. Figure 4.25 presents the computational time required for both procedures for approximating the parameter's posterior PDF in this case study.

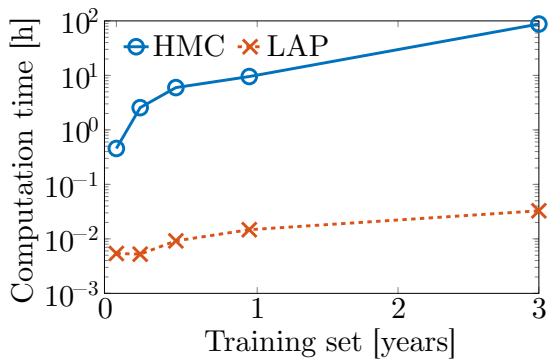


Figure 4.25 Computation time of LAP-P and HMC-P for approximating the parameter's posterior PDF on the real dataset. LAP: Laplace Approximation; HMC: Hamiltonian Monte Carlo.

The results show that HMC-P is again more reliable than LAP-P for small datasets such as 30 and 90 days presented in Figures 4.24a and b. Yet, Figures 4.24c, d and e show that the differences in the estimation between the LAP-P and HMC-P become unnoticeable for the TSLs of $180, 365$, and 1095 days.

4.7 Conclusion

The results show that the MGA algorithms are computationally more efficient than the BGA. Among the MGA algorithms, the NR and AMMT algorithms yield a faster convergence than the remaining algorithms, as they do not require tuning the learning rate. The MGA algorithms are thus an appropriate choice to handle large datasets. Nevertheless, there are not enough evidences to support which MGA algorithm is generally best-suited to optimize the model parameters in BDLMs because the performance may depend on the model architecture

as well as on the datasets.

The comparative studies between the HMC-P and LAP-P allow exposing the advantages as well as the limitations of both LAP-P and HMC-P. More specifically, the LAP-P provides a fast method for the approximation of model parameter posterior PDFs, yet it is prone to be trapped in a local maximum due to its sensitivity towards the selection of initial parameter values when short training sets are employed, for example, less than a year training data. In addition, the accuracy and feasibility of evaluating the model parameter covariance matrix become challenging for either high-dimensional parameter spaces, or for a small dataset size. To ensure a reliable approximation, the model must be (i) trained with a large amount of data and (ii) tested with the different sets of the initial parameter values. On the other hand, the HMC-P is less sensitive towards the initial parameter values and provides more reliable estimation than LAP-P, especially when the amount of data in the training set is limited. However, the computational cost for HMC-P is much higher than for the LAP-P. The results also show that both LAP-P and HMC-P provide a similar estimation accuracy in the case of large training datasets. Therefore, the LAP-P is well suited for the estimation task during model development. The HMC-P should be then compared with the estimation obtained using LAP-P.

CHAPTER 5 ONLINE LEARNING

5.1 Introduction

A key role of SHM is to detect changes in the behaviour of structures, that is, *anomalies*. The purpose of anomaly detection is to allow for preventive infrastructure maintenance in time. Existing sensor technologies allow civil infrastructure to be monitored continuously over time. To leverage this sensing capacity, there is a need for anomaly detection methodologies that are capable of performing real-time analysis, while being robust towards false alarms. *Real time* hereby means performing the data analysis as the new observations become available. The batch learning approach presented in Chapter 4 assumed that the vector of model parameters $\boldsymbol{\theta}_t$ is constant over time and are learned from a fixed training dataset. This assumption is no longer suitable for real-time anomaly detection where the underlying process in stream data can change over time [59, 60]. Furthermore, with batch learning, when a new data point arrives, the entire model needs to be retrained in order to estimate the new model parameters, thus making it computationally inefficient.

This chapter presents a new method combining the anomaly detection approach presented in Section 3.4 with the *Rao-Blackwellized Particle Filter* (RBPF) [121] for performing real-time anomaly detection in BDLMs. RBPF employs the analytical Kalman equations presented in Section 3.2 for estimating the posterior PDF of hidden state variables $\mathbf{x}_t$ and *Sequential Importance Sampling* (SIS) [122, 123] to approximate the posterior PDF of model parameters, $\boldsymbol{\theta}_t$. The main contributions of this chapter are:

- adapting the RBPF method to BDLMs.
- proposing a framework architecture to perform real-time anomaly detection in BDLMs.
- validating the new framework on several real-world datasets.

This chapter is organized as follows. Section 5.2 presents the adaptation of the mathematical formulation of RBPF for BDLMs. Section 5.3 presents the framework architecture designed for BDLMs to perform real-time anomaly detection. Section 5.4 illustrates the potential of new approach on several real datasets collected on different structures.

5.2 Rao-Blackwellized Particle Filter

The posterior PDF for both the hidden state variables and the model parameters, is theoretically defined as

$$p(\mathbf{x}_{1:t}, \boldsymbol{\theta}_{0:t}|\mathbf{y}_{1:t}) \quad = \quad p(\mathbf{x}_{1:t}|\boldsymbol{\theta}_{0:t}, \mathbf{y}_{1:t}) \cdot p(\boldsymbol{\theta}_{0:t}|\mathbf{y}_{1:t}), \tag{5.1}$$

where $p(\mathbf{x}_{1:t}|\boldsymbol{\theta}_{0:t}, \mathbf{y}_{1:t})$ is evaluated using the filtering equations presented in Section 3.2, and $p(\boldsymbol{\theta}_{0:t}|\mathbf{y}_{1:k})$ is obtained using SIS with a set of particles. According to Bayes theorem, the posterior PDF of model parameters can be written as follows

$$\begin{aligned} p(\boldsymbol{\theta}_{0:t}|\mathbf{y}_{1:t}) &\propto p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{0:t}) \cdot p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \mathbf{y}_{1:t-1}) \cdot p(\boldsymbol{\theta}_{0:t-1}|\mathbf{y}_{1:t-1}) \\ &\propto p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_t) \cdot p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) \cdot p(\boldsymbol{\theta}_{0:t-1}|\mathbf{y}_{1:t-1}), \end{aligned} \tag{5.2}$$

where the second formula takes advantage of Markov's assumptions for the transition prior $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$, and where $p(\boldsymbol{\theta}_{0:t-1}|\mathbf{y}_{1:t-1})$ is the posterior from the previous time step. Equation 5.2 is written as being proportional to the posterior because the normalization constant is analytically intractable. The PDF of model parameters can be approximated using the importance sampling method. If the proposal distribution is chosen following

$$q(\boldsymbol{\theta}_{0:t}|\mathbf{y}_{1:t}) = q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \mathbf{y}_{1:t}) \cdot q(\boldsymbol{\theta}_{0:t-1}|\mathbf{y}_{1:t-1}), \tag{5.3}$$

where the previous sets of particles $\boldsymbol{\theta}_{0:t-1}$ do not depend on future observations $\mathbf{y}_t$, that is, $q(\boldsymbol{\theta}_{0:t-1}|\mathbf{y}_{1:t-1}) \equiv q(\boldsymbol{\theta}_{0:t-1}|\mathbf{y}_{1:t})$. The importance weights can then be defined following

$$\begin{aligned} w_t &= \frac{p(\boldsymbol{\theta}_{0:t}|\mathbf{y}_{1:t})}{q(\boldsymbol{\theta}_{0:t}|\mathbf{y}_{1:t})} \\ &\propto \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_t) \cdot p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})}{q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \mathbf{y}_{1:t})} \cdot \frac{p(\boldsymbol{\theta}_{0:t-1}|\mathbf{y}_{1:t-1})}{q(\boldsymbol{\theta}_{0:t-1}|\mathbf{y}_{1:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_t) \cdot p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})}{q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \mathbf{y}_{1:t})} \cdot w_{t-1}. \end{aligned} \tag{5.4}$$

With the additional assumption that the transition PDF for new samples only depend on the most recent parameters and observations, $q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \mathbf{y}_{1:t}) \equiv q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \mathbf{y}_t)$, Equation 5.4 is rewritten as

$$w_t \quad \propto \quad \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_t) \cdot p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})}{q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \mathbf{y}_t)} \cdot w_{t-1}. \tag{5.5}$$

The choice for the proposal distribution $q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \mathbf{y}_t)$ can be, among others, the *prior sampling* or the *optimal sampling* [124]. The prior sampling can lead to inefficient exploration because

it does not take into account the current observation $\mathbf{y}_t$. Contrarily, in the optimal sampling, the current observation is included into the proposal distribution, yet it is commonly difficult to sample from this proposal distribution because of its analytical intractability [125]. The limitation can be addressed using the *auxiliary sampling* method [124] that resamples the particles using their marginal likelihood. The idea behind this sampling technique consists in preselecting the particles $\boldsymbol{\theta}_{t-1}$, that is, *surviving particles* that are likely to evolve into highly plausible particles $\boldsymbol{\theta}_t$ by considering the current observation. The proposal distribution is defined as

$$q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^k, \mathbf{y}_t) \quad = \quad p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^k) \cdot p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^k) \cdot w_{t-1}^k, \tag{5.6}$$

where $k \in \mathcal{K} = \{1, 2, \ldots, \mathtt{K}\}$ is the auxiliary index of the particle at time $t-1$. Therefore, the importance weight w_t^k in Equation 5.5 becomes

$$w_t^k \quad \propto \quad \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_t^k)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^k)}. \tag{5.7}$$

Because the auxiliary sampling method only prioritizes the surviving particles, it is prone to diversity loss in the particles over time. This issue can be tackled by adding artificial noise to the particles in order to increase the exploration capacity [126]. Assuming that there is a set of particles at the time t, $\boldsymbol{\theta}_t^{\mathcal{K}} = \{\boldsymbol{\theta}_t^1, \boldsymbol{\theta}_t^2, \ldots, \boldsymbol{\theta}_t^{\mathtt{K}}\}$, the transition model for the particles is defined as

$$\boldsymbol{\theta}_t^k \quad = \quad \boldsymbol{\theta}_{t-1}^k + \mathbf{u}_t, \tag{5.8}$$

where $\mathbf{u}_t$ is assumed to be a multivariate Gaussian distribution with zero mean and covariance matrix $\mathbf{D}_t$. Assuming that the artificial noise associated with each model parameter in the particle $\boldsymbol{\theta}_t^k$ is independent from each other, $\mathbf{D}_t$ becomes diagonal matrix following

$$\mathbf{D}_t \quad = \quad \mathrm{diag}(\underbrace{[\sigma_{u,1}^2 \ \sigma_{u,2}^2 \ \ldots \ \sigma_{u,\mathtt{p}}^2]^\intercal}_{\boldsymbol{\sigma}_{u,t}^2}), \tag{5.9}$$

where $\mathtt{p}$ is defined as the number of unknown model parameters in the particle vector $\boldsymbol{\theta}_t^k$ and $\sigma_{u,p}$ corresponds to the standard deviation of the artificial noise for the pth model parameter of $\boldsymbol{\theta}_t^k$. $\boldsymbol{\sigma}_{u,t}$, are unknown hyperparameters to be estimated from data. The main steps of the RBPF for BDLMs are summarized in Algorithm 1.

Algorithm 1: Rao-Blackwellized Particle Filter (RBPF) for BDLMs

Given $\boldsymbol{\theta}_0^{\mathcal{K}} \sim p(\boldsymbol{\theta}_0)$, $\mathbf{D}_t$, $\mathcal{K} = \{1, 2, \ldots, \mathtt{K}\}$, $\mathbf{w}_0 = \frac{1}{\mathtt{K}}$, η;
for $t = 1 : \mathtt{T}$ **do**
  for $k = 1 : \mathtt{K}$ **do**
    $(\sim, \sim, \tilde{\mathcal{L}}_t^k, \sim) = \mathrm{SKF}(\boldsymbol{\mu}_{t-1|t-1}^k, \boldsymbol{\Sigma}_{t-1|t-1}^k, \mathbf{y}_t, \mathbf{A}_{t-1}^k, \mathbf{C}_{t-1}^k, \mathbf{Q}_{t-1}^k, \mathbf{R}_{t-1}^k, \mathbf{Z}_{t-1}^k, \boldsymbol{\pi}_{t-1|t-1}^k)$;
  Sample indices $\mathcal{P}$ from $\mathcal{K}$ with probability proportional to $\mathcal{L}_t \odot \mathbf{w}_{t-1}$;
  $\boldsymbol{\theta}_{t-1}^{\mathcal{K}} = \boldsymbol{\theta}_{t-1}^{\mathcal{P}}$, $\boldsymbol{\mu}_{t-1|t-1}^{\mathcal{K}} = \boldsymbol{\mu}_{t-1|t-1}^{\mathcal{P}}$, $\boldsymbol{\Sigma}_{t-1|t-1}^{\mathcal{K}} = \boldsymbol{\Sigma}_{t-1|t-1}^{\mathcal{P}}$, $\boldsymbol{\pi}_{t-1|t-1}^{\mathcal{K}} = \boldsymbol{\pi}_{t-1|t-1}^{\mathcal{P}}$, $\tilde{\mathcal{L}}_t^{\mathcal{K}} = \tilde{\mathcal{L}}_t^{\mathcal{P}}$;
  for $k = 1 : \mathtt{K}$ **do**
    $\boldsymbol{\theta}_t^k = \boldsymbol{\theta}_{t-1}^k + \mathbf{u}_t,\ \mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{D}_t)$;
    $(\boldsymbol{\mu}_{t|t}^k, \boldsymbol{\Sigma}_{t|t}^k, \mathcal{L}_t^k, \boldsymbol{\pi}_{t|t}^k) = \mathrm{SKF}(\boldsymbol{\mu}_{t-1|t-1}^k, \boldsymbol{\Sigma}_{t-1|t-1}^k, \mathbf{y}_t, \mathbf{A}_t^k, \mathbf{C}_t^k, \mathbf{Q}_t^k, \mathbf{R}_t^k, \mathbf{Z}_t^k, \boldsymbol{\pi}_{t-1|t-1}^k)$;
    Compute weight $w_t^k = \frac{\mathcal{L}_t^k}{\tilde{\mathcal{L}}_t^k}$;
  Normalize weights $\mathbf{w}_t = \frac{\mathbf{w}_t}{\sum_{k=1}^{\mathtt{K}} w_t^k}$;

5.3 Framework Architecture

This section presents the framework architecture for BDLM's online learning procedure. Figure 5.1 illustrates the entire workflow where the framework architecture is separated into three main steps: *model construction*, *warm-up*, and *online estimation*. The model construction consists in pre-defining a vector of hidden state variables included in the model for interpreting the time-series data. The warm-up is employed for approximating the initial distribution for each model parameter. For this purpose, it can employ either the HMC or LAP methods presented in §4.4.4. This step ensures that the algorithm does not waste particles at places where the model parameter values are unlikely. Note that the warm-up step is operated in a batch learning procedure (see Chapter 4) with a small training period. The online estimation is performed using the RBPF, as described in Algorithm 1. Each particle, $k \in \mathcal{K} = \{1, 2, 3, \ldots, \mathtt{K}\}$, represents a realization of the posterior PDF of model parameters, thus there are $\mathtt{K}$ Gaussian PDFs describing the hidden state variables. For an intuitive interpretation, we employ the Gaussian mixture approximation presented in §4.4.3 for approximating the posterior predictive PDF of the hidden state variables with a single multivariate Gaussian PDF. The mean and covariance matrix of this distribution are given by Equation 4.30, where w_t^k is the normalized importance weight of the particle $\boldsymbol{\theta}_t^k$. The online-estimation step is recursively repeated as each new data point arrives. Note that

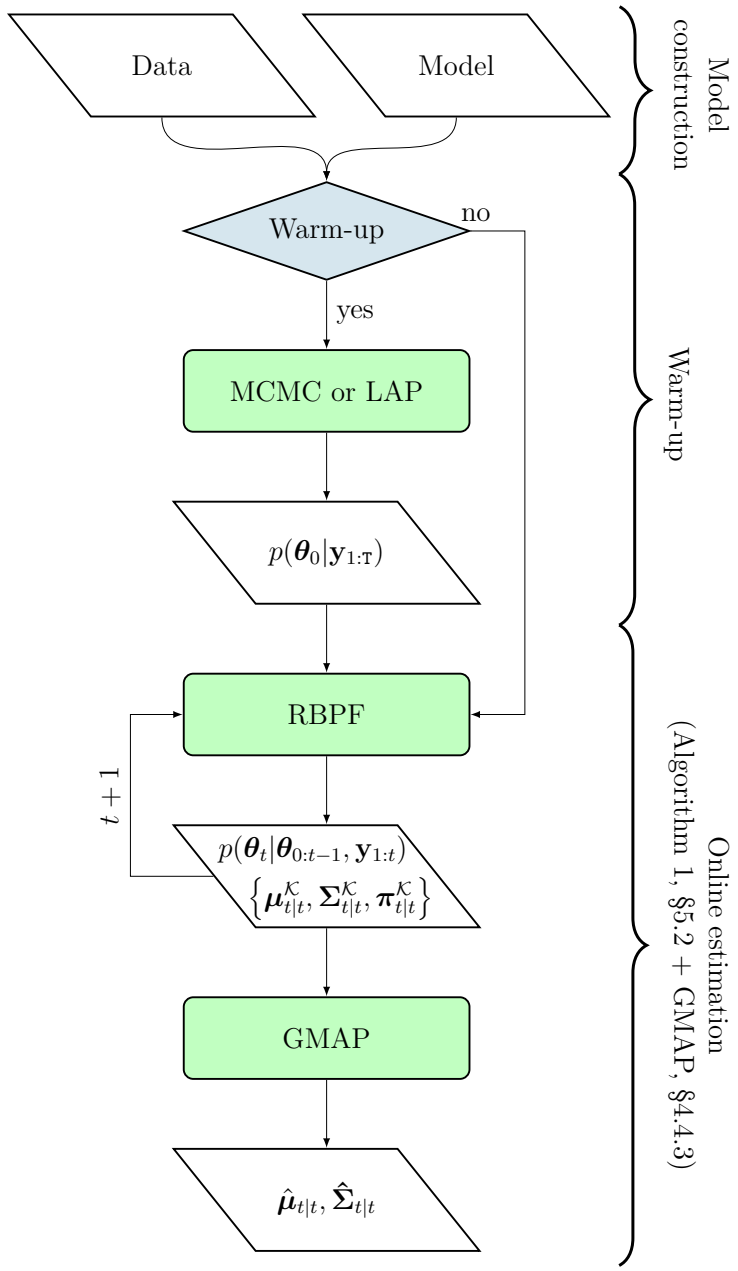


Figure 5.1 Illustration of the general framework of the online learning for the the Bayesian Dynamic Linear Models. RBPF: Rao-Blackwellized Particle Filter; MCMC: Markov Chain Monte Carlo; LAP: Laplace Approximation; GMAP: Gaussian Mixture Approximation.

the warm-up step is optional because in some cases, there are no data available for gaining the prior knowledge about the model parameters. Hence, the online estimation step can be performed directly as new observations become available.

In the context of anomaly detection, the model parameters in the BDLMs are categorized

into *stationary* and *non-stationary* model parameters. The stationary model parameters denoted as $\boldsymbol{\theta}_t^s$, are constant over time. For these model parameters, the introduction of artificial noise, as presented in Section 5.2, can cause a high variability in the hidden state estimation. Therefore, the standard deviations of the artificial noise, $\boldsymbol{\sigma}_{\mathbf{u},t}^s$, need to vanish overtime, so that

$$\boldsymbol{\sigma}_{\mathbf{u},t}^s = \frac{1}{\alpha} \cdot \boldsymbol{\sigma}_{\mathbf{u},t-1}^s, \tag{5.10}$$

where $\alpha > 1$ is a time-scaling factor. The non-stationary model parameters denoted as $\boldsymbol{\theta}_t^d$, are time-varying quantities that allow BDLMs to adapt to the changes of the underlying process in the data such as the occurrence of abnormal events. A key challenge is that the model parameters $\boldsymbol{\theta}_t^d$ may struggle to adapt to the situation where the underlying process in data goes from one regime to another. In such a case, $\boldsymbol{\theta}_t^d$ tends to be stuck to the values of a particular regime. This limitation leads to an increase in the uncertainties during the hidden state estimation. Hence, it can jeopardize the timing as well as the accuracy of anomaly detection. One solution to tackle this issue is to initialize these model parameters when the following conditions are satisfied

$$\begin{cases} \pi_{t-1|t-1}(\text{abnormal}) & \geq & \zeta \\ \pi_{t|t}(\text{abnormal}) & < & \zeta, \end{cases} \tag{5.11}$$

where $\pi_{t|t}(\text{abnormal})$ is the probability of the abnormal state at the time t and $\zeta \in (0, 1)$ is a probability threshold. In addition, the artificial noises of the stationary model parameters $\boldsymbol{\theta}_t^s$ need to be increased in order to provide a better exploitation. These artificial noises are set to its initial values, $\boldsymbol{\sigma}_{\mathbf{u},0}^s$, when the conditions in Equation 5.11 are met.

In the online learning procedure, the standard deviations for the artificial noises, $\boldsymbol{\sigma}_{\mathbf{u},0}$, are initialized based on the variance of the initial distribution for each model parameter. Each model-parameter group is defined following

$$\begin{aligned} \boldsymbol{\sigma}_{\mathbf{u},0}^s &= \gamma^s \sqrt{\text{Var}(\boldsymbol{\theta}_0^s)} \\ \boldsymbol{\sigma}_{\mathbf{u},0}^d &= \gamma^d \sqrt{\text{Var}(\boldsymbol{\theta}_0^d)}, \end{aligned} \tag{5.12}$$

where γ^s, γ^d are the scaling factors for the stationary and non-stationary model parameters and both of them are assumed to be constant over time. In practice, the value of γ^d is usually greater than the one of γ^s. By increasing the exploration capacity of the non-stationary model parameters, it allows the model to rapidly adapt to the changes in the observations. The hyparameters $\alpha, \gamma^s, \gamma^d, \zeta$, and the number of particles `K` need to be tuned before data analysis.

One solution to this issue is to employ an empirical study where different values of these hyperparameters are tested on multiple datasets collected on several structures. Furthermore, the number of particles $\mathtt{K}$ can be identified by testing different values of $\mathtt{K}$ on a small training set during the warm-up step. In common cases, the posterior PDF of model parameters will reach a stationary stage for which the difference in the posterior PDF of model parameters becomes unnoticeable after a specific number of particles. Note that there is a trade-off between the computation time and accuracy for the proposed method. More specifically, a large number of particles reduces the variance of the posterior PDF, yet it requires more computational time as well as more memory storage. A small number of particles requires less computational resources, yet it may yield a high variability in the posterior PDF.

5.4 Applications

This section presents the application of the real-time anomaly detection methodology proposed for BDLMs on displacement datasets that are collected on a dam located in Canada.

5.4.1 Horizontal Displacement of a Dam

This case study examines the capability to detect anomalies in real time through an application on a displacement dataset.

Experiment Setup

This case study uses the same displacement dataset and model architecture as the Model #5 in §3.7.3. The initial values for model parameters are defined using expert judgment and experience as well as prior data analysis so that

$$\boldsymbol{\theta}_0 = \left[\underbrace{0.5}_{\ell^{\mathtt{KR}}} \; \underbrace{0.95}_{\phi^{\mathtt{AR}}} \; \underbrace{0.095}_{\sigma_w^{\mathtt{AR}}} \; \underbrace{0.9999}_{Z^{11}} \; \underbrace{10^{-7}}_{\sigma_w^{\mathtt{LA}}} \; \underbrace{0.95}_{Z^{22}} \right]^{\mathsf{T}} . \tag{5.13}$$

Note that for this case study, $\{\ell^{\mathtt{KR}}, \phi^{\mathtt{AR}}, \sigma_w^{\mathtt{AR}}, Z^{11}\}$ are defined as the stationary model parameters, and $\{\sigma_w^{\mathtt{LA}}, Z^{22}\}$ are defined as the non-stationary model parameters. As presented in Section 5.3, the hyperparameters are $\alpha = 1.01$, $\zeta = 0.5$, $\gamma^s = 0.01$, $\gamma^d = 0.1$, and $\mathtt{K} = 60\,000$. The values of α, ζ, γ^s, and γ^d are obtained using an empirical study where different values of these hyperparameters are tested on multiple datasets. The number of particles $\mathtt{K}$ is obtained by testing different values such as $\mathtt{K} = \{10\,000, 20\,000, 30\,000, 60\,000\, 80\,000, 100\,000\}$ on a small training dataset consisting of 1024 days (1004 data points). The initial distribution for each

model parameter is obtained using the Laplace approximation (LAP) with the same training period of 1024 days.

Results

Figure 5.2 presents the probabilities of the abnormal state for the displacement data over time. The solid line presents the median probability of the abnormal state, $\tilde{p}(s_t = 2)$. The

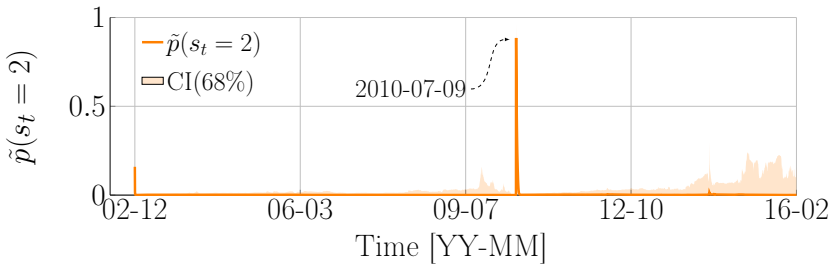
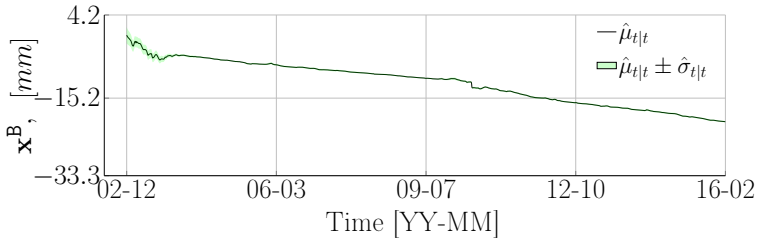


Figure 5.2 Probability of abnormal state for the displacement data.

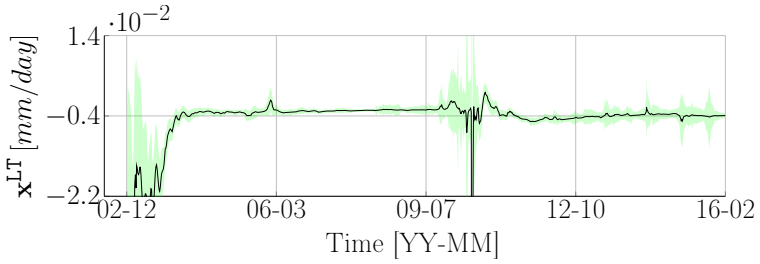
shaded region presents the 68% credible interval. The new approach identifies that there was an anomaly occurring on July 9, 2010. The detection timing for this anomaly matches the one obtained using the batch learning procedure presented in §3.7.3. This anomaly was due to the refection work taking place on the dam in early July. The displacement behaviour returns to the normal state once the work is completed.

Figure 5.3 shows the estimates of the hidden state variables. The mean values $\hat{\mu}_{t|t}$ are presented by the solid line and the uncertainty bounds $\hat{\mu}_{t|t} \pm \hat{\sigma}_{t|t}$ are presented by the shaded region. Figure 5.3a, b, and c show that there is an abrupt change in the baseline component ($\mathbf{x}^{\mathtt{B}}$), local trend ($\mathbf{x}^{\mathtt{LT}}$), and local acceleration ($\mathbf{x}^{\mathtt{LA}}$) when the anomaly occurred. Figure 5.3d shows the periodic pattern caused by the external effect. The autoregressive component presented in Figure 5.3e shows a stationary process. It is noted that a large variability is shown across all hidden state variables at early stage because of the lack of information, yet this variability vanishes over time as more data are collected. This case study shows how the baseline behaviour can be separated in real time from the reversible external effect.

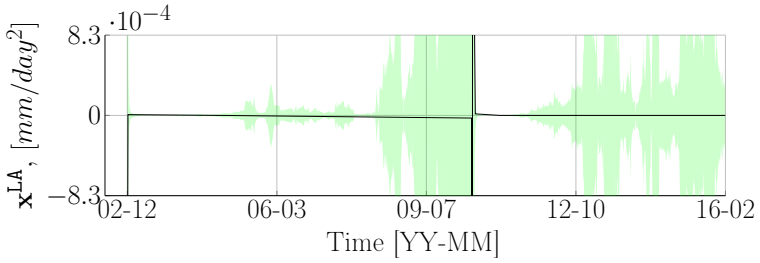
Figure 5.4 presents the evolution of the model parameters over time. The median values $\tilde{\boldsymbol{\theta}}_t$ are presented by the solid line and the 68% credible interval are presented shaded region. Figure 5.5 illustrates the kernel density estimate of the PDF for each model parameter on June 30, 2014. Figures 5.4a, b, c, and e show that the large uncertainty in the stationary model parameters during the initial period is due to the imperfect initial conditions. The



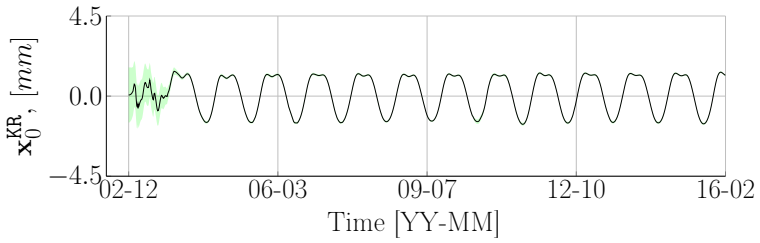
(a) baseline component



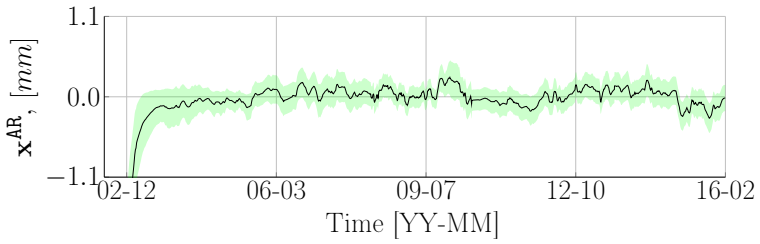
(b) local trend



(c) local acceleration

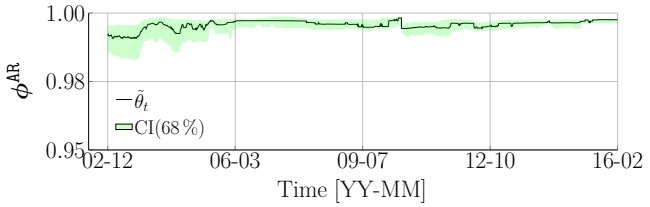


(d) kernel pattern

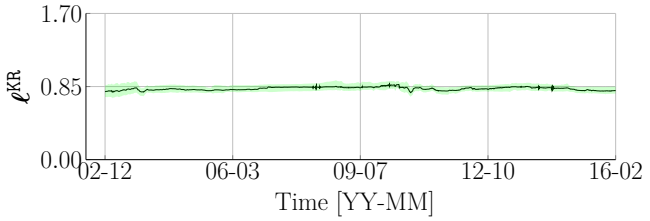


(e) autoregressive component

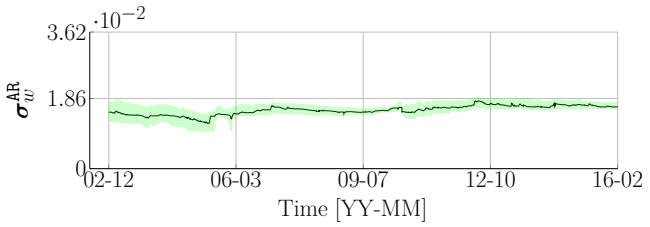
Figure 5.3 Hidden state estimates.



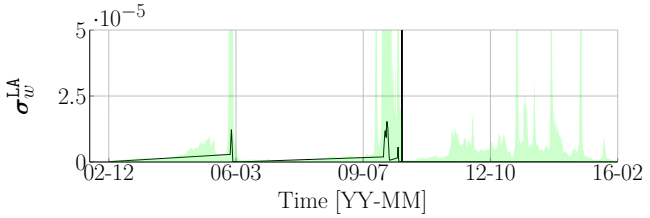
(a) autoregression coefficient



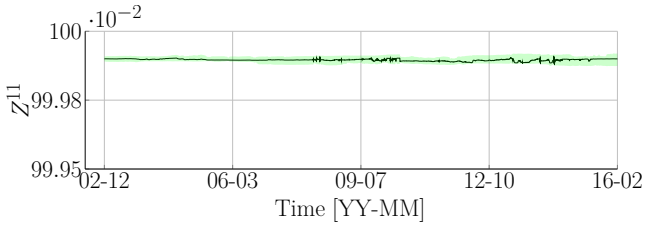
(b) kernel lengthscale



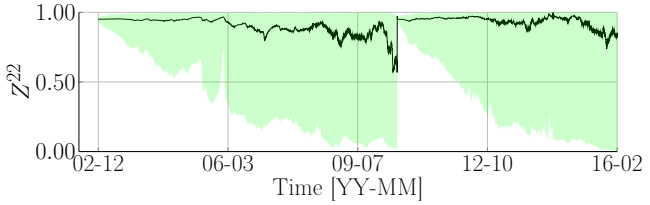
(c) autoregression standard deviation



(d) local acceleration standard deviation



(e) transition probability of the normal state



(f) transition probability of the abnormal state

Figure 5.4 Illustration of the estimation of model parameters using Rao-Blackwellized particle filter.

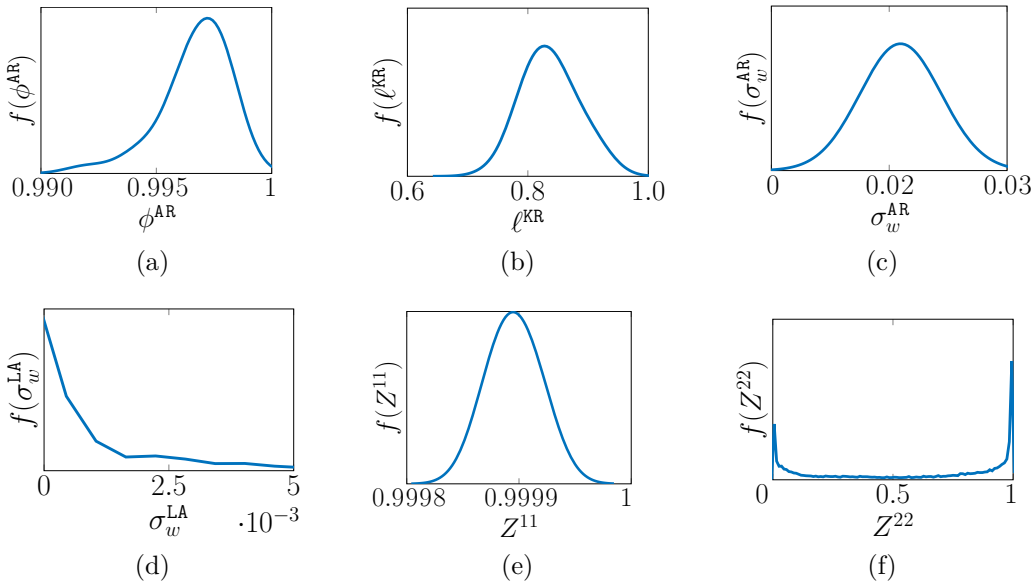


Figure 5.5 Kernel density estimate of the posterior PDFs for each model parameter taken at 3 am on June 30, 2014. (a) Autoregression coefficient; (b) Kernel lengthscale; (c) Autoregression standard deviation; (d) Local acceleration standard deviation; (e) Transition probability of the normal state; (f) Transition probability of the abnormal state.

uncertainty then reduces over time as the number of data points increases. When the anomaly took place, there is an abrupt change across all model parameters. However, the most significant change is found in the local acceleration standard deviation, $\sigma^{\mathtt{LA}}$ associated with the abnormal state. When the abnormal events are absent, a large variability can be observed in the non-stationary model parameters such as Z^{22} and $\sigma_w^{\mathtt{LA}}$ (see Figure 5.4d and f). This behaviour can be explained by the heavy tailed distribution of $\sigma_w^{\mathtt{LA}}$ and the presence of a bimodal distribution in Z^{11}, as illustrated in Figure 5.5d and f.

5.4.2 Horizontal Displacement of a Dam with Artificial Anomalies

This case study illustrates the potential of the real-time anomaly detection methodology on a dataset including multiple anomalies.

Experiment Setup

This case study employs the same dataset as in §4.6.1 except that two artificial anomalies are introduced to this dataset. The first anomaly is added from January 10 to February 6, 2009 with a slope of 15% of the displacement baseline. The second one is introduced from

July 12 to August 1, 2011 with the same slope as the previous one. Each anomaly period (≈ 1 month) consists of 60 data points. Figure 5.6 shows the superposition of the original and artificial-anomaly dataset. The other settings such as the initial values for model parameters,

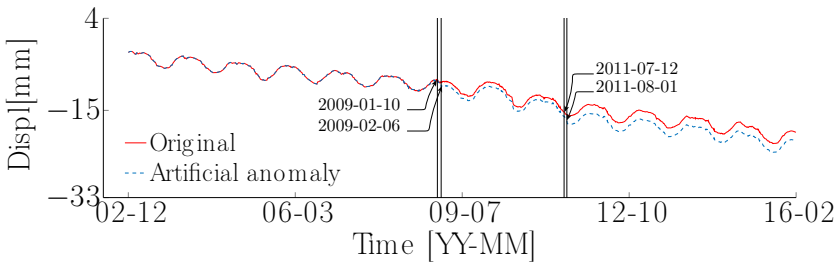


Figure 5.6 The superposition of the original and artificial-anomaly dataset.

hyperparameters, training period for LAP, and number of particles remain identical as the previous case study.

Results

In addition to the anomaly associated with the intervention occurring on July 9, 2010, two other anomalies are identified on February 6, 2009 and August 6, 2011. These two anomalies correspond to the times where the artificial anomalies were introduced in the original dataset. Figure 5.7 presents the probability of abnormal state for the displacement dataset over time. The median values of the abnormal state probability $\tilde{p}(s_t = 2)$ are presented by the solid line.

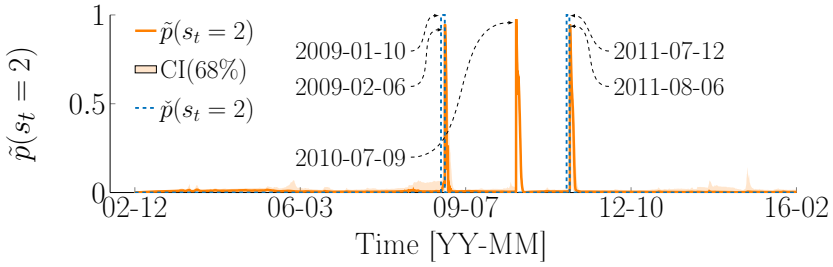


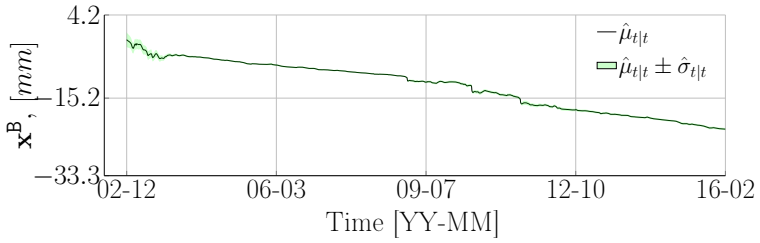
Figure 5.7 Probability of abnormal state for the displacement data.

The 68% credible intervals are presented by the shaded region. The probabilities of abnormal state for the artificial anomalies $\check{p}(s_t = 2)$ are presented by the dashed line. Because the anomaly is introduced by gradually adding a small slope to the original data, it takes time to create a significant change in their underlying process. This explains why the timing of

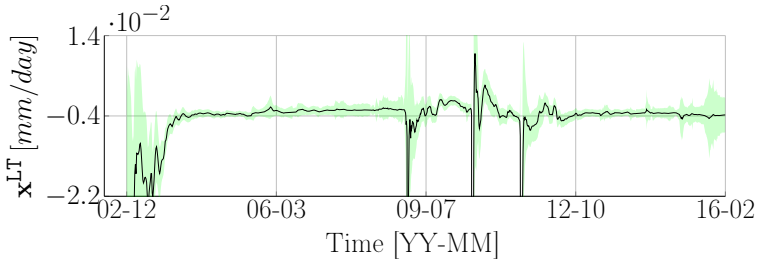
anomaly detection provided by the model, does not match exactly with the starting dates of the artificial anomaly. Figure 5.8 presents the estimates of hidden state variables for the entire dataset. The solid line represents the expected values, $\hat{\mu}_{t|t}$ and the shaded region represents the uncertainty bounds, $\hat{\mu}_{t|t} \pm \hat{\sigma}_{t|t}$. As for the previous case study, the sudden changes are found in the baseline, local trend, and local acceleration when the anomalies occurred. The large variability across all hidden state variables during the initial period decreases over time as more data become available. As illustrated in Figure 5.8d, despite the presence of multiple anomalies, the model is capable of separating the structural behaviour from the external effect. In addition, Figure 5.8e shows that the autoregressive component follows a stationary process.

The estimation of model parameters for this case study is presented in Figure 5.9. The median values of the model parameters, $\tilde{\boldsymbol{\theta}}_t$, and its 68% credible interval, are presented by the solid line and shaded region. This case study shows the same behaviour as the previous case study. More specifically, a large variability in the stationary model parameters is observed at an early stage, and it then vanishes as more data are collected. A sudden change is identified across the model parameters in the presence of the abnormal events. Figure 5.9 shows that the major changes corresponding to the abnormal events are found in the local acceleration standard deviation $\sigma_w^{\mathtt{LA}}$. Figure 5.9f shows that when the anomalies occurred, the changes in Z^{22} were not as clear as the one observed in the previous case study (see Figure 5.4f). This can be explained by the two following reasons:

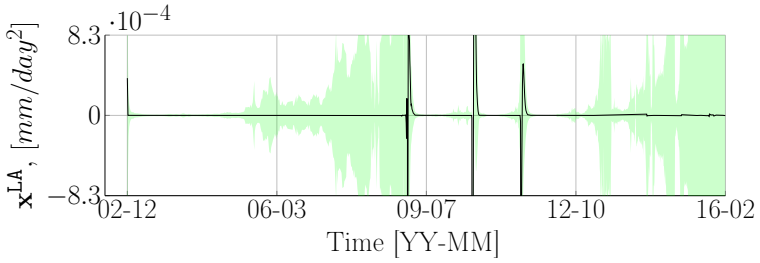
1. When the anomalies are absent, the non-stationary model parameters such as the transition probability Z^{22} does not have an impact on the importance weight w_t in Equation 5.7. Therefore, the values of Z^{22} can be within a range between 0 to 1. This results in a bimodal distribution in Z^{22} for the previous case study (see Figure 5.5f) where the majority of particles are found close to either 0 or 1. In the presence of an anomaly, a sudden change has been observed in the median of Z^{22} because the particle values close to 0 move toward 1.
2. In this case study, most particle values are found in a range from 0.5 to 1 and the median of Z^{22} is close to 1. Hence, there are no significant changes in Z^{22} when the anomalies occurred.



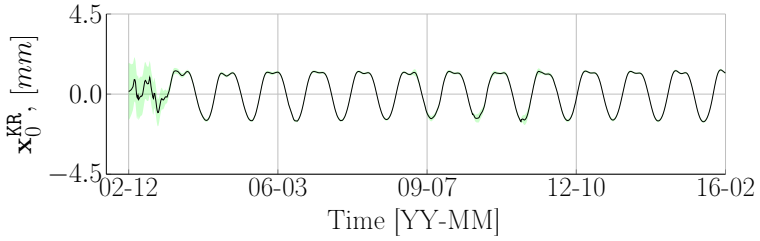
(a) baseline component



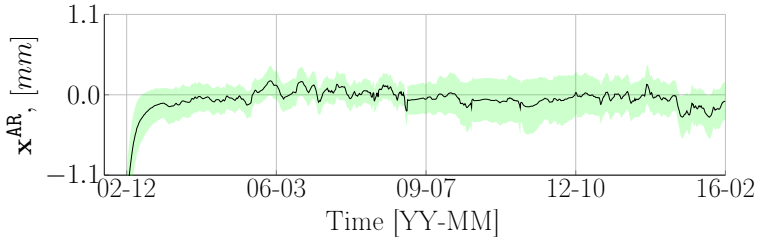
(b) local trend



(c) local acceleration

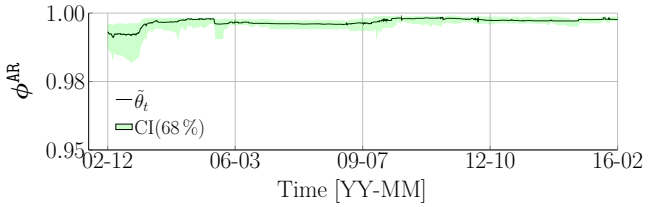


(d) kernel pattern

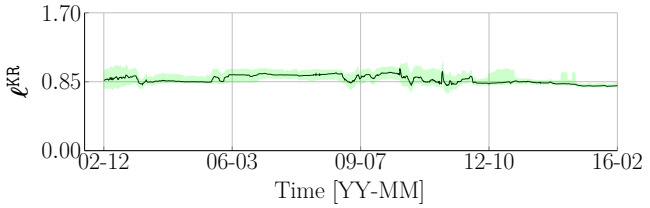


(e) autoregressive component

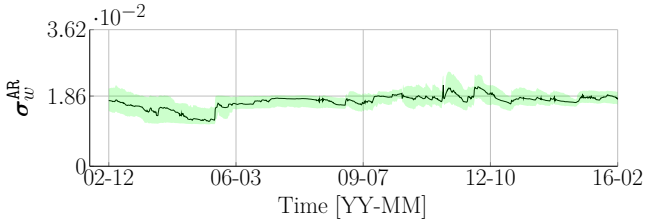
Figure 5.8 Illustration of the estimation of the hidden state variables.



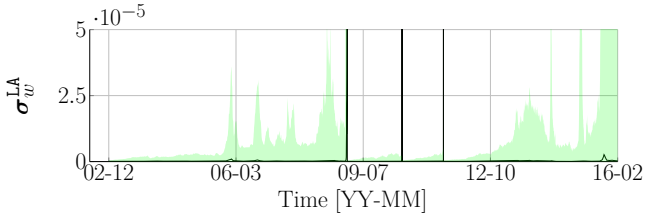
(a) autoregression coefficient



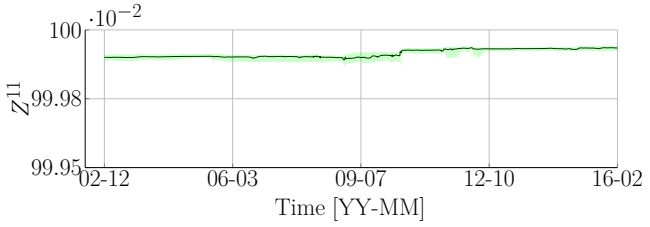
(b) kenel lengthscale



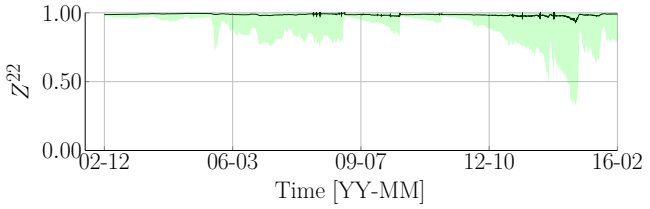
(c) local acceleration standard deviation



(d) autoregression standard deviation



(e) transition probability of the normal state



(f) transition probability of the abnormal state

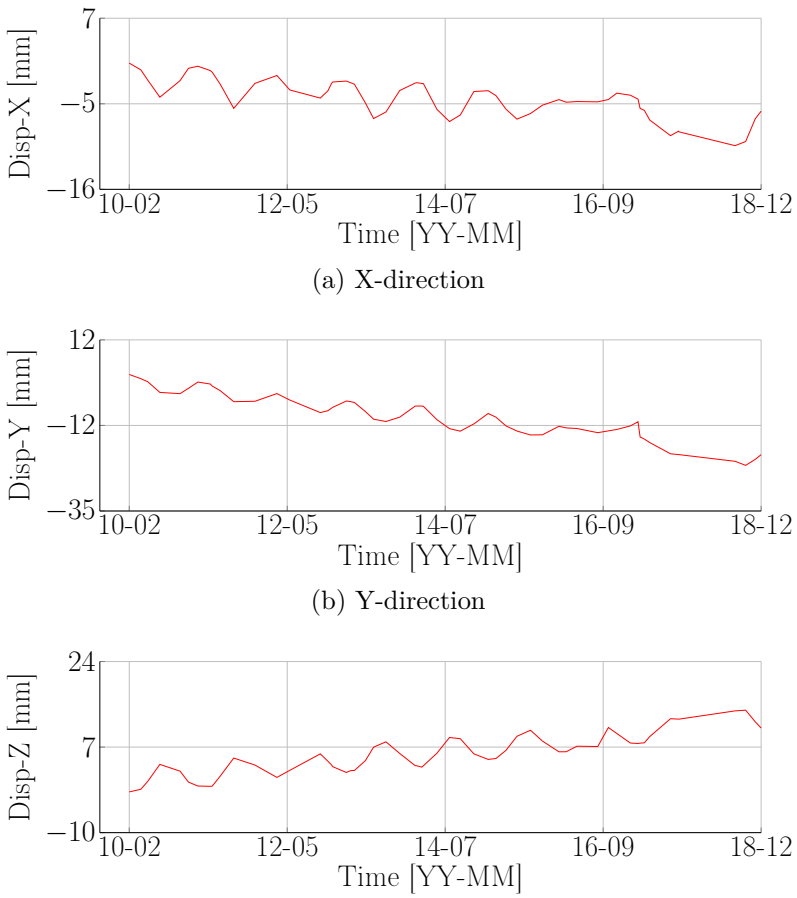
Figure 5.9 Estimation of model parameters using Rao-Blackwellized particle filter.

5.4.3 Three-Dimensional Displacement of a Dam

This case study illustrates the potential of the real-time anomaly detection methodology in the context of an application on a small dataset. For this purpose, the three displacement datasets corresponding to the X, Y, and Z-directions (see Figure 3.10) are employed in this case study.

Data Description

Figure 5.10 presents the displacement data in three directions X, Y, and Z. The displacement



(a) X-direction

(b) Y-direction

(c) Z-direction

Figure 5.10 Illustration of the raw displacement data in three directions.

data are collected from February, 2010 to December, 2018. The number of data points for the X, Y, and Z directions are $57, 57$, and 56. The observation-error standard deviation $\sigma_v = 0.3\,\text{mm}$ was provided by the instrumentation engineers. Figure 5.11 presents a superposition of the time-step length for all three directions. Note that the data collection frequency is extremely

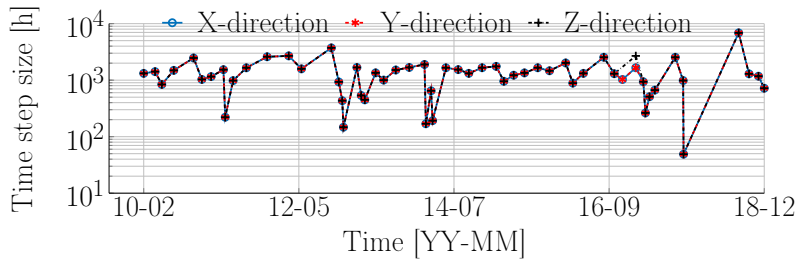


Figure 5.11 Superposition of three time-step lengths.

low during the entire recording period of 8 years. It is observed that there is a periodic pattern and a trend in all three directions for the displacement.

Architecture for Model #9

The vector of hidden state variables for Model #9 includes a baseline (B) component, a local trend (LT), a local acceleration (LA), and a kernel-regression component with a period of 365.24 days. The baseline component is used to model the displacement behaviour over time. The local trend is employed to model the rate of changes in the baseline component. The local acceleration is used to model the rate of changes in the local trend. The kernel-regression component including 11 hidden state variables describes the periodic pattern. The vector of hidden state variables for Model #9 defined as

$$\mathbf{x}_t = \left[x^{\mathtt{B}}\ x^{\mathtt{LT}}\ x^{\mathtt{LA}}\ x_0^{\mathtt{KR}}\ x_1^{\mathtt{KR}}\ \ldots\ x_{10}^{\mathtt{KR}}\right]_t^{\mathsf{T}}. \tag{5.14}$$

Note that like for Model #5 (see §3.7.3), two model classes, normal and abnormal, are built for Model #9 in which the local acceleration of the abnormal model class is forced to be equal to zero. The vector of unknown model parameters for Model #9 is

$$\boldsymbol{\theta}_t = \left[\ell^{\mathtt{KR}}\ \sigma_{w,0}^{\mathtt{KR}}\ \sigma_{w,1}^{\mathtt{KR}}\ Z^{11}\ \sigma_w^{\mathtt{LA}}\ Z^{22}\right]^{\mathsf{T}}, \tag{5.15}$$

where $\ell^{\mathtt{KR}} \in (0, +\infty)$ is the kernel lengthscale, $\sigma_{w,0}^{\mathtt{KR}} \in (0, +\infty)$ is the standard deviation of the kernel pattern, $\sigma_{w,1}^{\mathtt{KR}} \in (0, +\infty)$ is the standard deviation of the hidden state variables associated with the control point's values, $Z^{11} \in [0, 1]$ is the transition probability of the normal state, $\sigma^{\mathtt{LA}} \in (0, +\infty)$ is the local acceleration standard deviation, and $Z^{22} \in [0, 1]$ is the transition probability of the abnormal state. The natural logarithm transformation function is used for the kernel lengthscale and the standard deviations; The logistic sigmoid function is applied to the transition probabilities. The full model matrices are presented in

Appendix A.9. The initial values of model parameters for all three directions are defined in the original space using expert judgment and experience as well as prior data analysis such that

$$\begin{aligned} \text{X-direction} \quad \boldsymbol{\theta}_0 &= [0.5\ 0.27\ 0.027\ 0.9999\ 10^{-7}\ 0.95]^{\intercal} \\ \text{Y-direction} \quad \boldsymbol{\theta}_0 &= [0.5\ 0.62\ 0.062\ 0.9999\ 10^{-7}\ 0.95]^{\intercal} \\ \text{Z-direction} \quad \boldsymbol{\theta}_0 &= [0.5\ 0.41\ 0.041\ 0.9999\ 10^{-7}\ 0.95]^{\intercal}, \end{aligned} \tag{5.16}$$

where the ordering of model parameters remains identical as in Equation 5.15. Because the number of data points in each direction is limited, the warm-up step is not employed for approximating the initial distribution of each model parameter in the transformed space. For this case, the initial distribution for each model parameter is assumed to be a Gaussian

$$f\left(\theta_{0,p}^{\mathtt{TR}}\right) = \mathcal{N}\left(\theta_{0,p}^{\mathtt{TR}}, \left(0.1 \cdot \theta_{0,p}^{\mathtt{TR}}\right)^2\right), \forall p = 1, 2, \ldots, 6, \tag{5.17}$$

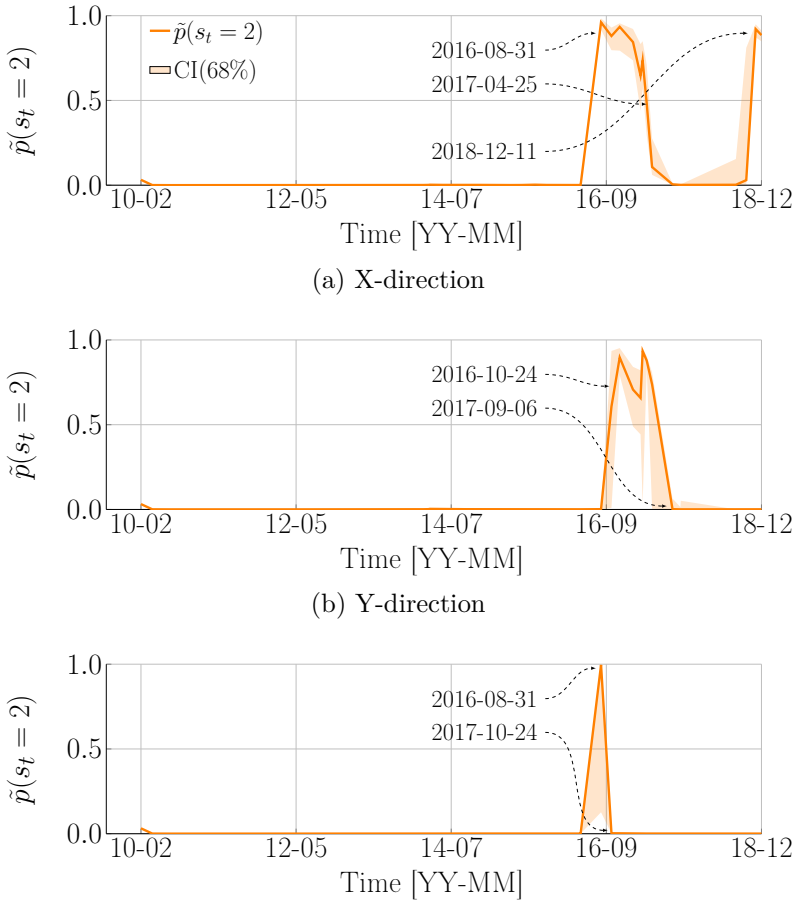
where $\theta_{0,p}^{\mathtt{TR}}$ correspond to p^{th} model parameter value of $\boldsymbol{\theta}_0$ in the transformed space. The hyperpameters remain identical as the previous case studies: $\alpha = 1.01$, $\zeta = 0.5$, $\gamma^s = 0.01$, $\gamma^d = 0.1$, and $\mathtt{K} = 60\,000$. Note that these initial setups are selected using the empirical study as shown in Section 5.3.

Results

Figure 5.12 presents the probabilities of the abnormal state for the displacement data in three directions over time. The solid line presents the median probability of the abnormal state, $\tilde{p}(s_t = 2)$. The shaded area presents its 68% credible interval. An anomaly has been identified in all three directions in 2016. In addition, another anomaly is found in the X-direction displacement in 2018. Here, the causes of these possible anomalies remain unknown.

Figures 5.13-5.15 show the estimates of hidden state variables for the displacement data in three directions. The mean values, $\hat{\mu}_{t|t}$ and its uncertainty bounds, $\hat{\mu}_{t|t} \pm \hat{\sigma}_{t|t}$, are presented by the solid line and shaded region. Like the previous case studies presented in §5.4.1 and §5.4.2, when the anomalies occurred, sudden changes in the baseline component ($\mathbf{x}^{\mathtt{B}}$), local trend ($\mathbf{x}^{\mathtt{LT}}$), local acceleration ($\mathbf{x}^{\mathtt{LA}}$) have been observed for all three directions. Figures 5.13d, 5.14d, and 5.15d show that Model #9 is able to separate the structural behaviour from the periodic external effect. A high variability is found at early stage across all hidden state variables for three directions because of the lack of information. This variability then reduces over time as more data are collected.

Figures 5.16-5.18 present the estimation of model parameters over time for three directions. The median values, $\tilde{\boldsymbol{\theta}}_t$, and its 68% credible interval, are presented by the solid line and



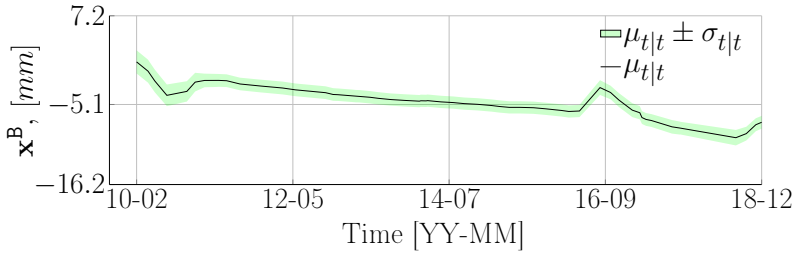
(a) X-direction

(b) Y-direction

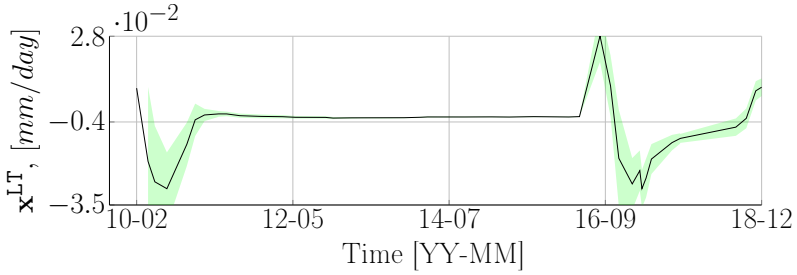
(c) Z-direction

Figure 5.12 Probability of the abnormal state for the displacement data in three directions.

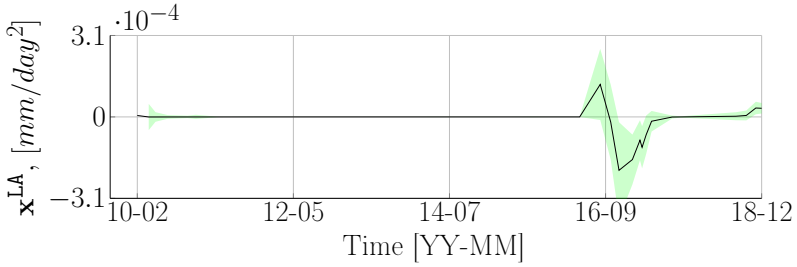
shaded area. As for the previous case studies, a major change is found in the local acceleration standard deviation $\sigma^{\mathtt{LA}}$ in the presence of the anomaly for all three directions, as presented in Figures 5.16d, 5.17d, and 5.18d.



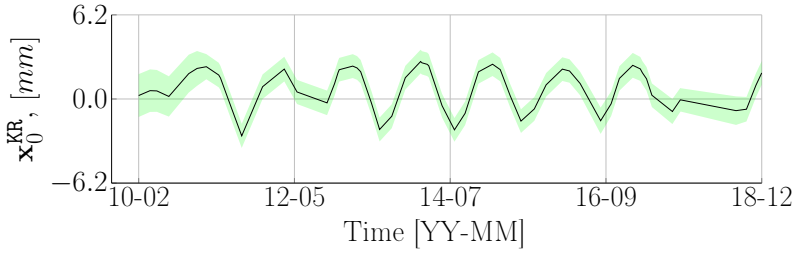
(a) baseline component



(b) local trend

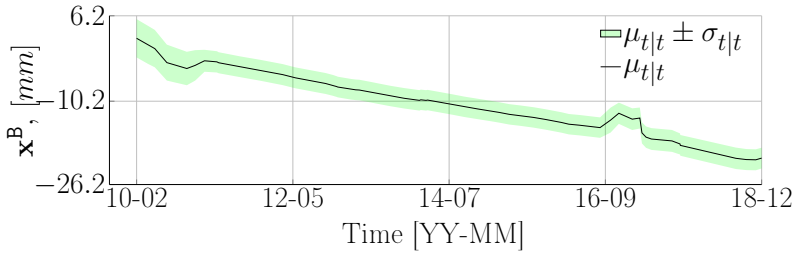


(c) local acceleration

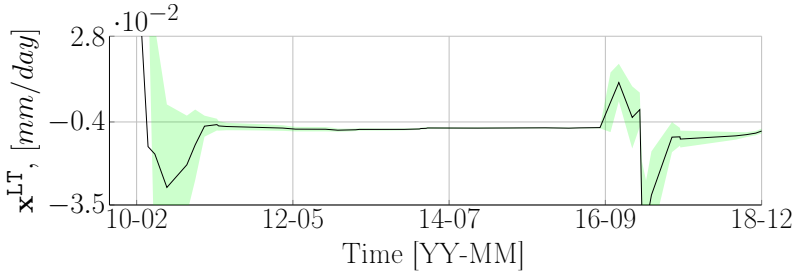


(d) kernel pattern

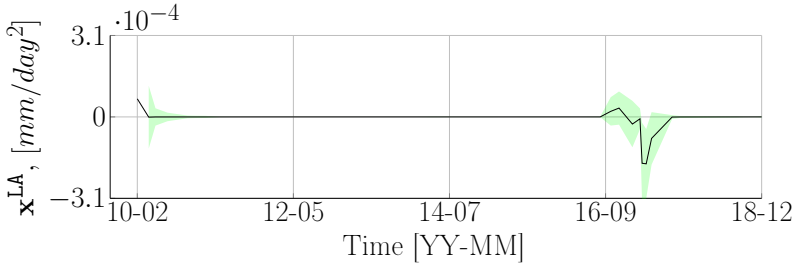
Figure 5.13 Estimation of the hidden state variables for X-direction displacement.



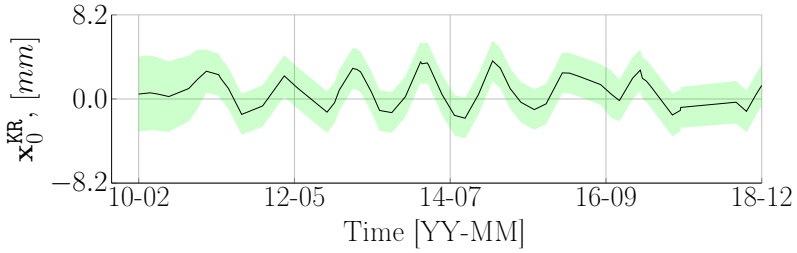
(a) baseline component



(b) local trend

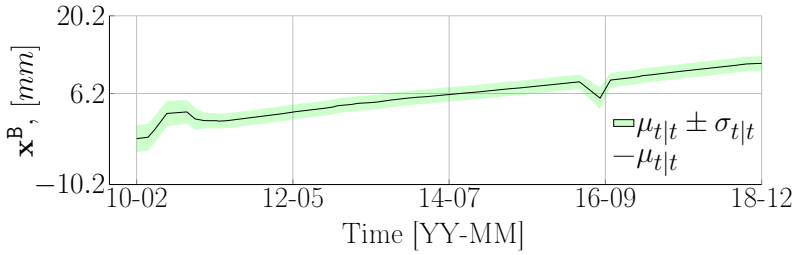


(c) local acceleration

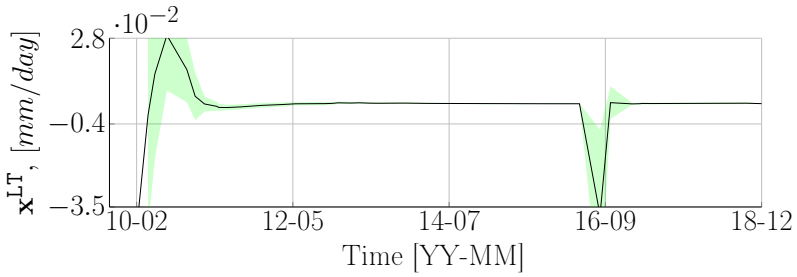


(d) kernel pattern

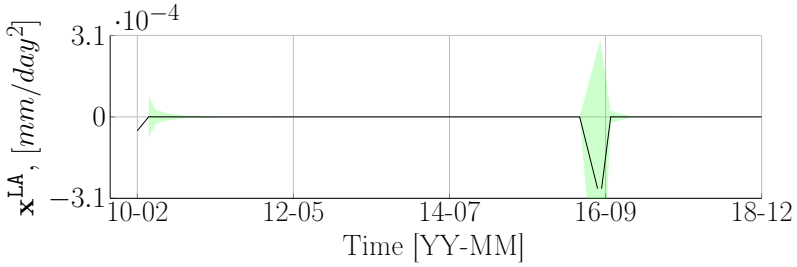
Figure 5.14 Estimation of the hidden state variables for Y-direction displacement.



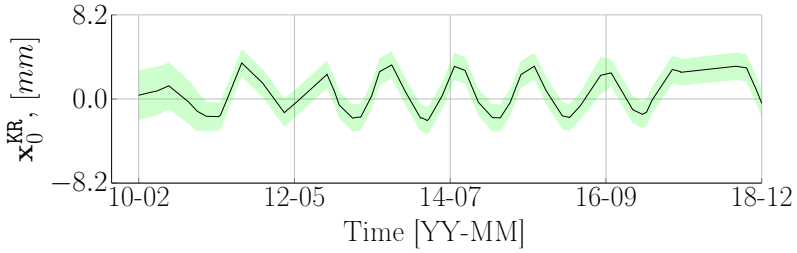
(a) baseline component



(b) local trend

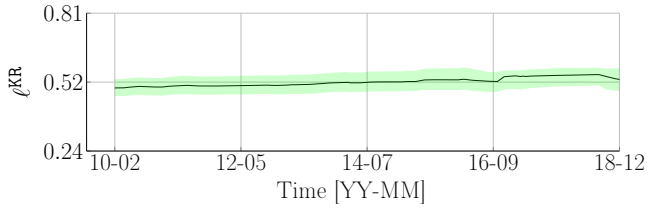


(c) local acceleration

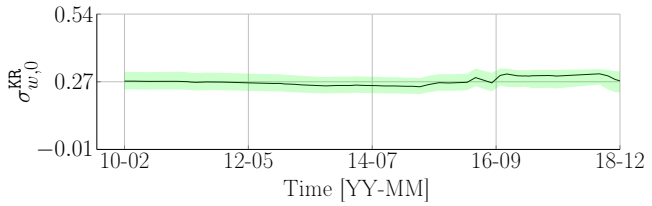


(d) kernel pattern

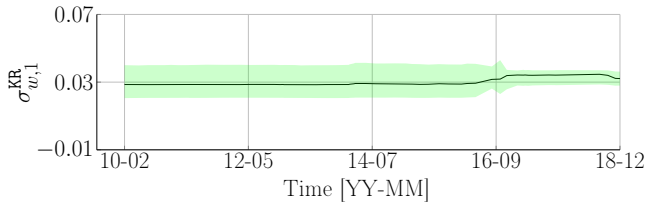
Figure 5.15 Estimation of the hidden state variables for Z-direction displacement.



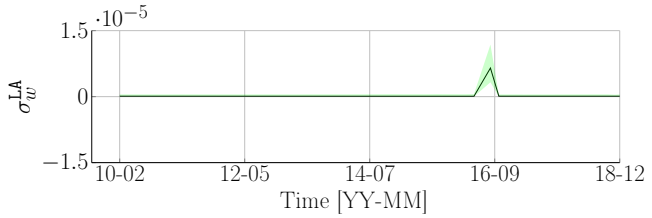
(a) kernel lengthscale



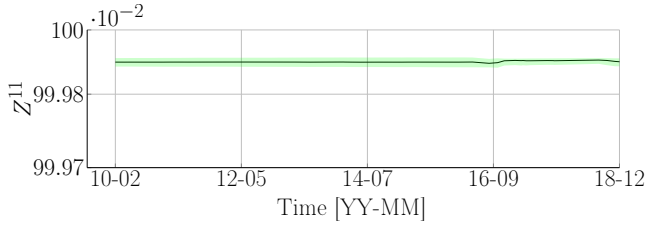
(b) standard deviation of the kernel pattern



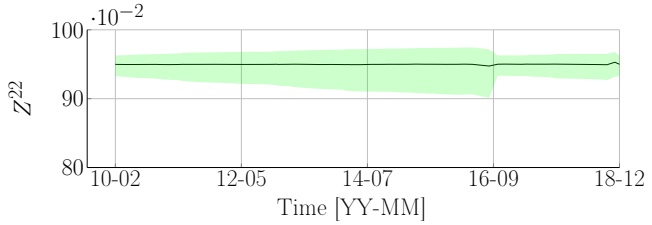
(c) standard deviation of the control point's values



(d) local acceleration standard deviation

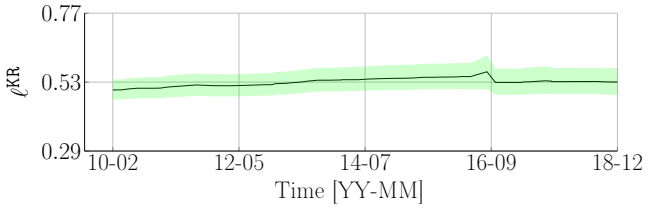


(e) transition probability of the normal state

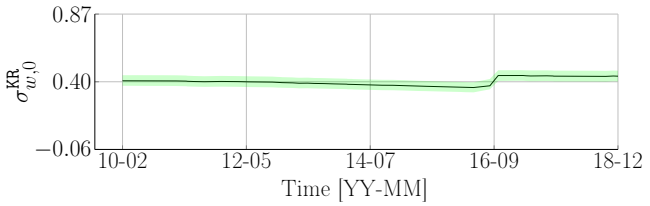


(f) transition probability of the abnormal state

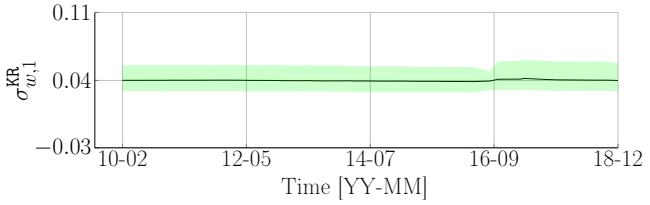
Figure 5.16 Estimation of model parameters for the X-direction using Rao-Blackwellized particle filter.



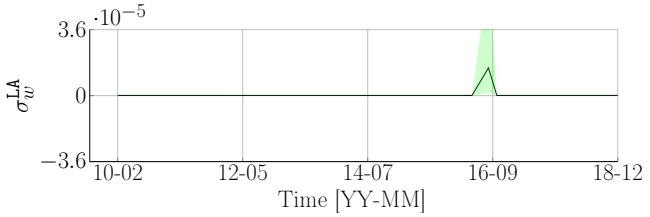
(a) kernel lengthscale



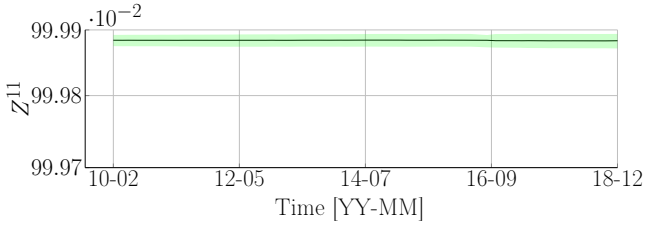
(b) standard deviation of the kernel pattern



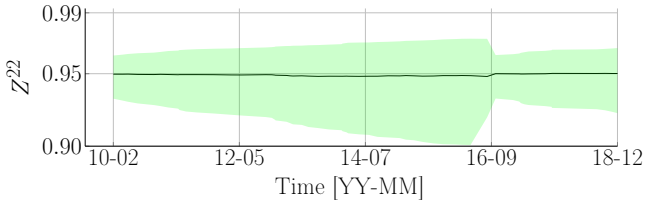
(c) standard deviation of the control point's values



(d) local acceleration standard deviation

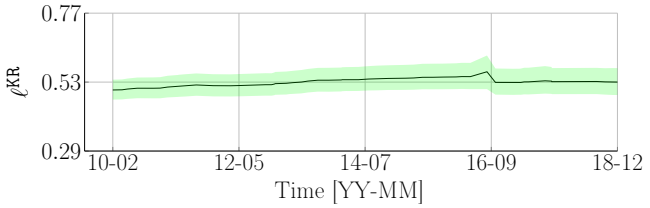


(e) transition probability of the normal state

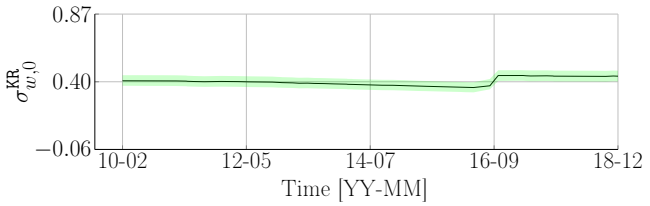


(f) transition probability of the abnormal state

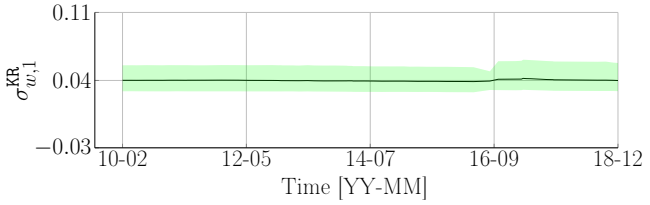
Figure 5.17 Estimation of model parameters for the Y-direction using Rao-Blackwellized particle filter.



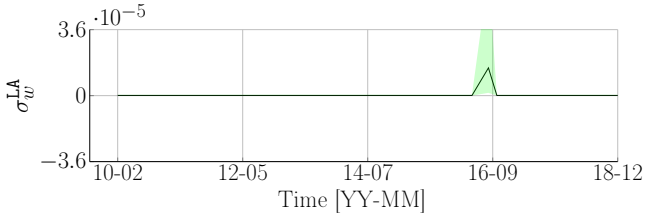
(a) kernel lengthscale



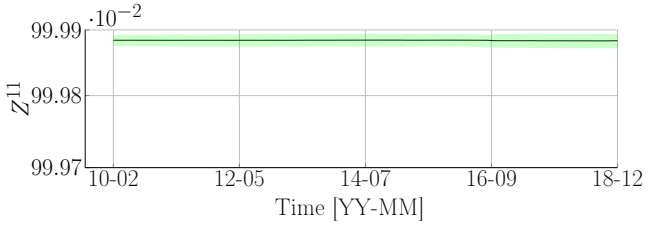
(b) standard deviation of the kernel pattern



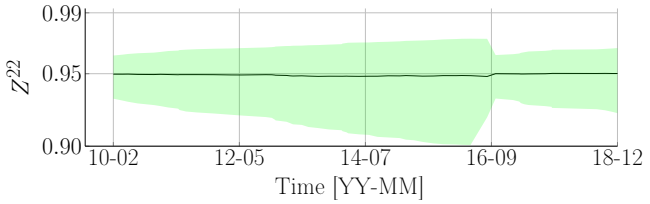
(c) standard deviation of the control point's values



(d) local acceleration standard deviation



(e) transition probability of the normal state



(f) transition probability of the abnormal state

Figure 5.18 Estimation of model parameters for the Z-direction using Rao-Blackwellized particle filter.

5.5 Conclusion

The new approach is capable of detecting the anomaly caused by the refection work and also the anomalies being artificially introduced to the original dataset. The approach also shows its ability to detect anomalies on small datasets for which data collection frequency is low and non-uniform. The changes in the structural behaviour can influence multiple sensors as shown in the third case study. In addition to the real-time anomaly detection, the proposed method allows estimating the hidden state variables as well as model parameters as the data is collected. Because the new approach leverages the parallelism of the GPU computation, the estimation task of each time step was completed within the second range. This computational time is negligible in comparison with the data collection frequency in which the most frequent time-step is in the hour range.

The initial distribution of model parameters can be obtained using either the LAP approach as shown in the first and second case studies or an empirical study like for the third case study. For all three case studies, the period of the external effect is a known quantity. Therefore, a small dataset should be available for carrying on the prior data analysis. Furthermore, the hyperparameters need to be tuned before performing the anomaly-detection task. The generic hyperparameters can be obtained using the empirical study. The observation-error standard deviation, σ_v, in all three case studies is assumed to be constant over time. In practice, this hypothesis might no longer be valid because of the presence of either the sensor drift or the imperfect installation conditions.

Future work should investigate the possibility (i) of processing several time series simultaneously and (ii) of taking into account the phenomena such as the sensor drift and the imperfect installation conditions in the current framework in order to increase the timing accuracy of anomaly detection as well as reduce the number of false alarms.

CHAPTER 6 CONCLUSION

6.1 Summary of Research Findings

This thesis proposes several new machine learning methods for interpreting time-series data in the context of structural health monitoring. A brief summary of the findings for each chapter are presented below.

Chapter 3 presented a kernel regression approach for modeling periodic external effects and a switching-Kalman-filter-based method for detecting anomalies in the observed structural responses. The case study has shown that the kernel regression approach was able to handle complex non-harmonic periodic patterns such as the traffic load data collected on a bridge. Also, it requires less computational time than the existing methods in BDLMs. For the anomaly detection methodology, its mechanism considers the prior probability of anomalies, the anomaly kinematic model, and the probability to transition from a normal to an abnormal state. The advantages of this mechanism are that it increases robustness towards false alarms and it does not require training data labeled with normal and abnormal conditions. The application on the displacement data collected on a dam has shown that the proposed methodology succeeded in detecting the changes in the behaviour of the displacement data recorded on a dam.

Chapter 4 introduced Maximum Likelihood Estimate (MLE) and Bayesian approaches for learning unknown model parameters in BDLMs. More specifically, the MLE approach employs either batch or mini-batch gradient ascent algorithms for finding an optimal vector of model parameters. The comparative case study has demonstrated that the mini-batch gradient ascent algorithm outperformed the batch gradient ascent when it comes to large datasets. On the other hand, the Bayesian approach enables the BDLMs to take into account the parameter estimate uncertainties. For this purpose, two procedures based on Hamiltonian Monte Carlo (HMC) and Laplace approximation (LAP) methods are proposed for approximating the posterior PDF of model parameters. The case study carrying on a dam's displacement data has shown that both procedures provided a similar estimation accuracy in the case of large training datasets. When dealing with small training datasets, the HMC-based procedure yielded a more reliable estimation than LAP-based procedure, yet it was more computationally demanding. Therefore, the LAP-based procedure is well suited for the estimation task during model development. The HMC-based procedure should be then compared with the estimation obtained using the LAP-based procedure.

Chapter 5 proposed a general framework for real-time anomaly detection. This framework employs the Rao-Blackwellized Particle Filter (RBPF) for estimating the hidden state variables as well as the posterior PDF of model parameters for each time step. Also, it involves the methodologies being proposed in the previous chapters. The results obtained from case studies have shown that the proposed framework was capable of detecting the anomalies caused by the refection work and also the anomalies being artificially introduced on the original displacement dataset. Furthermore, this framework has illustrated its capability to perform the real-time anomaly detection on small datasets for which data collection frequency is extremely low and non-uniform. Because the new framework leverages the parallelism of the GPU computation, the estimation task of each time step was completed within the second range. This computational time is negligible in comparison with the data collection frequency, and thus allows for the deployment on a large number of structures for real-time data analysis.

The conclusion we can draw from the findings summarized above is that the proposed methodologies are capable of (i) isolating the structural behaviour from raw data including the external effects caused by environmental conditions as well as observation errors, (ii) automatically finding optimal model parameters as well as approximating, online and offline, the posterior probability density function of model parameters, and (iii) detecting anomalies in real time without human supervision and without requiring labeled training data. The applications of these methodologies on different structural responses from a dam and a bridge show their ability to be easily transferable from one structure to another and from one measurement type to another. Putting this all together, the proposed methodologies offer a promising path toward the large-scale deployment of SHM systems for monitoring health and conditions of a population of structures in real time.

6.2 Limitations

This section discusses the limitation of the methodologies proposed in this dissertation. Although the potential of the proposed methodologies have been illustrated on several applications, some aspects need to be addressed in order to further increase their robustness and applicability.

6.2.1 Criterion for Detecting Anomalies

For the anomaly detection methodology proposed in this dissertation, the occurrence of an anomaly currently relies on the probability of abnormal state. If the probability of abnormal state is above a certain threshold, an anomaly is detected. With a criterion, a high threshold

value may miss anomalies, while a low threshold value may increase false alarms. Therefore, the abnormal probability alone is not a sufficient criterion for defining anomalies. One possible solution could be to rely on the probability of abnormal state as well as the structural behaviour including the baseline component, local trend, and local acceleration in identifying anomalies. This could help separate legitimate and illegitimate alarms.

6.2.2 Non-Structural Anomaly

In the context of SHM, non-structural anomalies can be sensor drifts and sensor faults caused by the environmental conditions, installation conditions, and physical changes in the sensors. As a result, non-structural anomalies may lead to inaccurate measurement recordings of the structural responses. The framework proposed for the anomaly detection does not take into account these phenomena. Therefore, it makes the anomaly detection task more challenging when interpreting time series from a structure that contains non-structural anomalies. The reason for this challenge is that the proposed framework cannot distinguish the structural anomalies from the non-structural ones, and as a consequence, it typically increases the rate of false alarm.

6.2.3 Initializing Model Parameters and Hyperparameter Tuning

The performance of the proposed methodologies depends on the initialization of model parameters and the hyperparameter tuning. For instance, good guesses for the value of initial parameter such as the pattern period (see §3.6.2) might be needed to ensure an efficient optimization. The initialization task is currently done using the prior data analysis such as data visualization and statistical data analysis. Therefore, a small dataset is required for carrying out this analysis. On the other hand, the hyperparameters are selected using an empirical study where the different values of these hyperparameter are tested on multiple datasets. However, these values may no longer be valid for several datasets. For addressing this challenge, it is required to develop a generic procedure for tuning hyperparameters that can adapt to each individual dataset.

6.3 Future Research

In addition to limitations mentioned in §6.2 to be addressed, this section lays out some of directions for future research.

6.3.1 Topology Learning

Topology learning is a technique that identifies relationships between different sensors [127]. In the context of SHM, a monitoring system for a structure commonly comes with multiple sensors measuring different structural responses, where some structural responses may be spatially related to each other. These dependencies induce spatial patterns into observed structural responses. For instance, multiple sensors measuring displacements are deployed across a dam in order to monitor its deformation. Therefore, these displacement sensors are likely to be correlated [128]. An approach capable of finding spatial patterns can potentially improve the performance for the anomaly detection and forecasting. To extract spatial patterns, a possible framework architecture we could consider is to couple the BDLMs with the convolutional LSTM (ConvLSTM) [129]. In the field of transportation, several applications [130, 131] using the ConvLSTM for modeling spatial patterns in traffic networks have shown promising results.

6.3.2 Anomaly Detection and Information Redundancy

False alarms are the main factor limiting the economic viability of SHM applications. For instance, the methodology proposed in Chapter 5 can only be used to detect anomalies for an individual dataset. As mentioned in §6.3.1, the structures being monitored are commonly equipped with multiple sensors, and some of them may be correlated. As a result, there is a high likelihood that anomalies may occur on several correlated datasets at the same time. For example, the case study presented in §5.4.3 has shown that the anomaly affected all three displacement datasets. Hence, a methodology that allows using the information redundancy contained in multiple datasets has the potential to reduce false alarms.

Figure 6.1 presents a framework describing the possible steps for Multiple-Datasets Analysis (MDA). In this example, we assume that the states are defined for two levels; a dataset-level $s_i = \{normal, abnormal\}, \forall i = 1 : \mathtt{D}$, and a system-level $s_{\text{system}} = \{normal, abnormal\}$. The abnormal state is presented by a warning sign. For each dataset $\mathcal{D}_i$, $p(s_i|\mathcal{D}_i)$ is estimated independently using the online learning framework (OLF) presented in Section 5.3. These probabilities will then be combined in a second conditional probability model that will estimate $p(s_{\text{system}}|\mathcal{D}_i)$. The main challenge associated with this framework will be to develop this conditional probability model. The specificities of this model remain to be further investigated.

6.3.3 Non-Periodic Pattern Modeling

In practice, time-series data can exhibit both periodic and non-periodic patterns. For instance, the structural responses such as the flow rate and pressure data recorded on a dam mostly

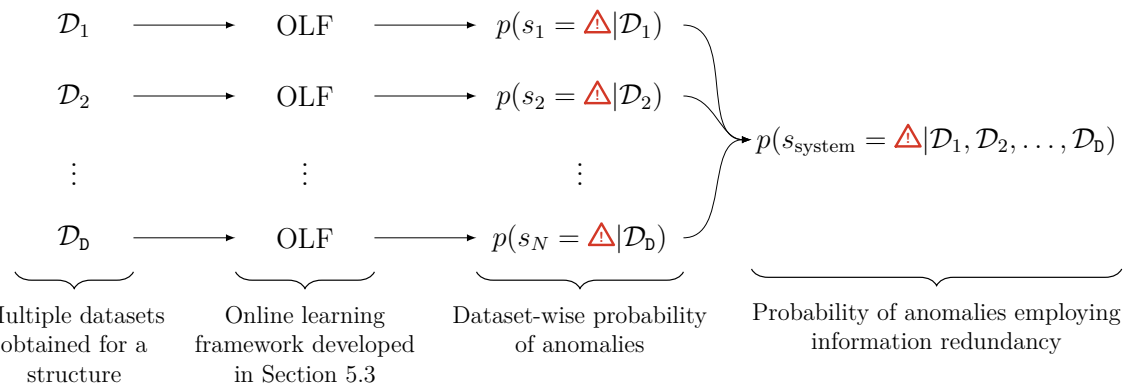


Figure 6.1 Illustration of multiple-datasets analysis (MDA) framework in the context of Structural Health Monitoring.

display non-periodic patterns. Figure 6.2 illustrates the raw data of both the flow rate and pressure that exhibit non-periodic patterns. The current form of BDLMs is unable

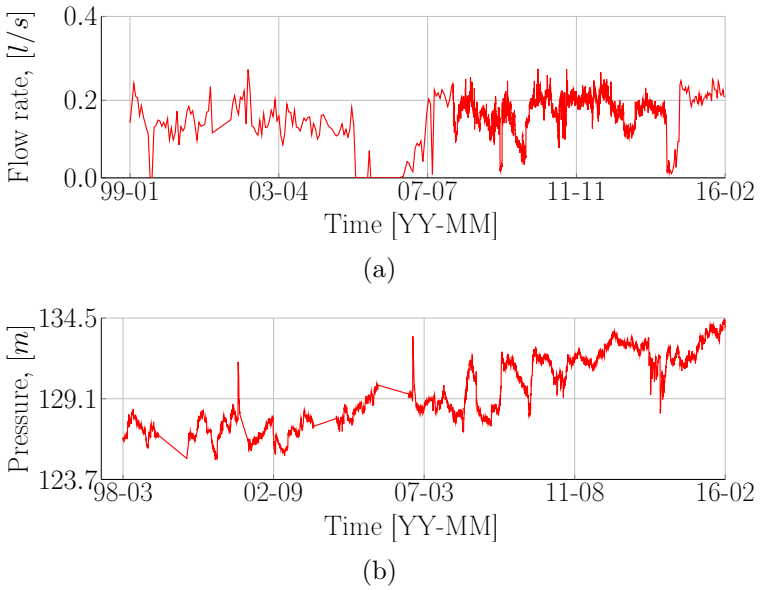


Figure 6.2 Illustration of non-periodic patterns. (a) Flow-rate data; (b) Pressure data.

to effectively model non-periodic patterns. More speciffically, non-periodic patterns are commonly considered as the model errors captured by the autoregressive component, leading to a poor predictive capacity. For this reason, the current BDLMs need to be extended to deal with non-periodic patterns. One potential solution to this issue could be to combine BDLMs with *Long Short Term Memory* (LSTM) [132] which specializes in modeling non-periodic pattern in time series.

REFERENCES

[1] Cebr, *Engineering and economic growth: a global view.* London, UK: Royal Academy of Engineering, 2016. [Online]. Available: https://www.raeng.org.uk/publications/reports/engineering-and-economic-growth-a-global-view

[2] A. Ansar, B. Flyvbjerg, A. Budzier, and D. Lunn, "Does infrastructure investment lead to economic growth or economic fragility? evidence from China," *Oxford Review of Economic Policy*, vol. 32, no. 3, pp. 360–390, 2016.

[3] B. Égert, T. J. Kozluk, and D. Sutherland, "Infrastructure and growth: empirical evidence," *OECD*, 2009.

[4] CCA, CPWA, CSCE, and FCM, *Canadian Infrastructure Report Card.* Ottawa, Canada: The Canadian Infrastructure Report Card, 2016. [Online]. Available: http://canadianinfrastructure.ca/en/index.html

[5] ASCE, *2017 Report card for America's infrastructure.* Reston, VA: American Society of Civil Engineers, 2017. [Online]. Available: https://www.infrastructurereportcard.org

[6] C. Macilwain, "Out of service," *Nature*, vol. 462, p. 846, December 2009.

[7] D. Stiff and P. Smetanin, "Public infrastructure underinvestment: The risk to canada's economic growth," Risk Analytica, Tech. Rep., 2010.

[8] CANCEA, *Infrastructure Update 2018: Ontario Infrastructure Investment - Federal and Provincial Risks and Rewards.* Ontario, Canada: Residential and Civil Construction Alliance of Ontario, 2018. [Online]. Available: http://rccao.com/news/files/RCCAO_Infrastructure-Update-2018.pdf

[9] P. Cawley and R. Adams, "The location of defects in structures from measurements of natural frequencies," *The Journal of Strain Analysis for Engineering Design*, vol. 14, no. 2, pp. 49–57, 1979.

[10] R. Adams, P. Cawley, C. Pye, and B. Stone, "A vibration technique for non-destructively assessing the integrity of structures," *Journal of Mechanical Engineering Science*, vol. 20, no. 2, pp. 93–100, 1978.

[11] M. Sun, W. Staszewski, and R. Swamy, "Smart sensing technologies for structural health monitoring of civil engineering structures," *Advances in Civil Engineering*, vol. 2010, 2010.

[12] F. Catbas, T. Kijewski-Correa, and A. Aktan, *Structural Identification of Constructed Facilities. Approaches, Methods and Technologies for Effective Practice of St-Id.* Reston, VA: American Society of Civil Engineers (ASCE), 2013.

[13] J. Lynch and K. Loh, "A summary review of wireless sensors and sensor networks for structural health monitoring," *Shock and Vibration Digest*, vol. 38, no. 2, pp. 91–130, 2006.

[14] Z. Chen, X. Zhou, X. Wang, L. Dong, and Y. Qian, "Deployment of a smart structural health monitoring system for long-span arch bridges: A review and a case study," *Sensors*, vol. 17, no. 9, p. 2151, 2017.

[15] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[16] L. Bull, K. Worden, G. Manson, and N. Dervilis, "Active learning for semi-supervised structural health monitoring," *Journal of Sound and Vibration*, vol. 437, pp. 373–388, 2018.

[17] R. Kromanis and P. Kripakaran, "Support vector regression for anomaly detection from measurement histories," *Advanced Engineering Informatics*, vol. 27, no. 4, pp. 486–495, 2013.

[18] P. C. Chang, A. Flatau, and S. Liu, "Health monitoring of civil infrastructure," *Structural Health Monitoring*, vol. 2, no. 3, pp. 257–267, 2003.

[19] C. Farrar and K. Worden, "An introduction to structural health monitoring," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 303–315, 2007.

[20] F. N. Çatbaş, T. Kijewski-Correa, and A. E. Aktan, "Structural identification of constructed systems," *ASCE, Reston, VA*, 2013.

[21] J.-A. Goulet, P. Kripakaran, and I. F. C. Smith, "Multimodel structural performance monitoring," *Journal of Structural Engineering*, vol. 136, no. 10, pp. 1309–1318, 2010.

[22] J. M. W. Brownjohn, A. De Stefano, Y.-L. Xu, H. Wenzel, and A. E. Aktan, "Vibration-based monitoring of civil infrastructure: challenges and successes," *Journal of Civil Structural Health Monitoring*, vol. 1, no. 3-4, pp. 79–95, 2011.

[23] R. Kromanis and P. Kripakaran, "Shm of bridges: characterising thermal response and detecting anomaly events using a temperature-based measurement interpretation approach," *Journal of Civil Structural Health Monitoring*, vol. 6, no. 2, pp. 237–254, 2016.

[24] H. Sohn, K. Worden, and C. R. Farrar, "Statistical damage classification under changing environmental and operational conditions," *Journal of Intelligent Material Systems and Structures*, vol. 13, no. 9, pp. 561–574, 2002.

[25] J. M. W. Brownjohn, "Structural health monitoring of civil infrastructure," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 589–622, 2006.

[26] R. Kromanis and P. Kripakaran, "Data-driven approaches for measurement interpretation: analysing integrated thermal and vehicular response in bridge structural health monitoring," *Advanced Engineering Informatics*, vol. 34, pp. 46–59, 2017.

[27] F. Salazar, R. Morán, M. Á. Toledo, and E. Oñate, "Data-based models for the prediction of dam behaviour: a review and some methodological considerations," *Archives of Computational Methods in Engineering*, vol. 24, no. 1, pp. 1–21, 2017.

[28] K. P. Murphy, *Machine learning: a probabilistic perspective.* Cambridge, Massachusetts: The MIT Press, 2012.

[29] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach.* Essex, England: Pearson Education Limited, 2016.

[30] Y. Engel, *Algorithms and representations for reinforcement learning.* Jerusalem, Israel: Hebrew University of Jerusalem, 2005.

[31] S. Ferry and G. Willm, "Méthodes d'analyse et de surveillance des déplacements observés par le moyen de pendules dans les barrages," in *VIth International Congress on Large Dams, ICOLD*, vol. 2, New York, 1958, pp. 1179–1201.

[32] G. Willm and N. Beaujoint, "Les méthodes de surveillance des barrages au service de la production hydraulique d'électricité de france, problèmes anciens et solutions nouvelles,"

in *IXth International Congress on Large Dams, ICOLD*, Istanbul, Turkey, 1967, pp. 529–550.

[33] F. Lugiez, N. Beaujoint, and X. Hardy, "L'auscultation des barrages en exploitation au service de la production hydraulique d'électricité de france, des principes aux résultats," in *Xth International Congress on Large Dams, ICOLD*, Montreal, Canada, 1970, pp. 577–600.

[34] P. Léger and S. Seydou, "Seasonal thermal displacements of gravity dams located in northern regions," *Journal of Performance of Constructed Facilities*, vol. 23, no. 3, pp. 166–174, 2009.

[35] M. Tatin, M. Briffaut, F. Dufour, A. Simon, and J.-P. Fabre, "Thermal displacements of concrete dams: Accounting for water temperature in statistical models," *Engineering Structures*, vol. 91, pp. 26 – 39, 2015.

[36] P. Léger and M. Leclerc, "Hydrostatic, temperature, time-displacement model for concrete dams," *Journal of Engineering Mechanics*, vol. 133, no. 3, pp. 267–277, 2007.

[37] J. Mata, A. Tavares de Castro, and J. Sá da Costa, "Constructing statistical models for arch dam deformation," *Structural Control and Health Monitoring*, vol. 21, no. 3, pp. 423–437, 2014.

[38] S. Gamse and M. Oberguggenberger, "Assessment of long-term coordinate time series using hydrostatic-season-time model for rock-fill embankment dam," *Structural Control and Health Monitoring*, vol. 24, no. 1, pp. e1859–n/a, 2017.

[39] R. Kromanis and P. Kripakaran, "Predicting thermal response of bridges using regression models derived from measurement histories," *Computers & Structures*, vol. 136, pp. 64–77, 2014.

[40] W.-H. Hu, Á. Cunha, E. Caetano, R. G. Rohrmann, S. Said, and J. Teng, "Comparison of different statistical approaches for removing environmental/operational effects for massive data continuously collected from footbridges," *Structural Control and Health Monitoring*, vol. 24, no. 8, p. e1955, 2017.

[41] I. Laory, T. Trinh, D. Posenato, and I. Smith, "Combined model-free data-interpretation methodologies for damage detection during continuous monitoring of structures," *Journal of Computing in Civil Engineering*, vol. 27, no. 6, pp. 657–666, 2013.

[42] N. Dervilis, K. Worden, and E. Cross, "On robust regression analysis as a means of exploring environmental and operational conditions for shm data," *Journal of Sound and Vibration*, vol. 347, pp. 279–296, 2015.

[43] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts: MIT Press, 2016.

[45] J. Mata, "Interpretation of concrete dam behaviour with artificial neural network and multiple linear regression models," *Engineering Structures*, vol. 33, no. 3, pp. 903 – 910, 2011.

[46] R. Fedele, G. Maier, and B. Miller, "Health assessment of concrete dams by overall inverse analyses and neural networks," *International Journal of Fracture*, vol. 137, no. 1-4, pp. 151–172, 2006.

[47] F. Kang, J. Liu, J. Li, and S. Li, "Concrete dam deformation prediction model for health monitoring based on extreme learning machine," *Structural Control and Health Monitoring*, pp. e1997–n/a, 2017.

[48] J. Zhang, X. Cao, J. Xie, and P. Kou, "An improved long short-term memory model for dam displacement prediction," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[49] F. Salazar González, "A machine learning based methodology for anomaly detection in dam behaviour," Ph.D. dissertation, Polytechnic University of Catalonia, Barcelona, Spain, 2017.

[50] V. Ranković, A. Novaković, N. Grujović, D. Divac, and N. Milivojević, "Predicting piezometric water level in dams via artificial neural networks," *Neural Computing and Applications*, vol. 24, no. 5, pp. 1115–1121, 2014.

[51] D. Santillán, J. Fraile-Ardanuy, and M. Á. Toledo, "Dam seepage analysis based on artificial neural networks: The hysteresis phenomenon," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. Dallas, TX, USA: IEEE, 2013, pp. 1–8.

[52] P. Pandey and S. Barai, "Multilayer perceptron in damage detection of bridge structures," *Computers and Structures*, vol. 54, no. 4, pp. 597–608, 1995.

[53] J. Gu, M. Gul, and X. Wu, "Damage detection under varying temperature using artificial neural networks," *Structural Control and Health Monitoring*, vol. 24, no. 11, p. e1998, 2017.

[54] Z. Tang, Z. Chen, Y. Bao, and H. Li, "Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring," *Structural Control and Health Monitoring*, vol. 26, no. 1, p. e2296, 2019.

[55] N. Dervilis, M. Choi, I. Antoniadou, K. Farinholt, S. Taylor, R. Barthorpe, G. Park, K. Worden, and C. Farrar, "Novelty detection applied to vibration data from a cx-100 wind turbine blade under fatigue loading," in *Journal of Physics: Conference Series*, vol. 382, no. 1. IOP Publishing, 2012, p. 012047.

[56] N. Bakhary, H. Hao, and A. J. Deeks, "Damage detection using artificial neural network with consideration of uncertainties," *Engineering Structures*, vol. 29, no. 11, pp. 2806–2815, 2007.

[57] A. R. P. Bandara, "Damage identification and condition assessment of building structures using frequency response functions and neural networks," Ph.D. dissertation, Queensland University of Technology, Brisbane, Australia, 2013.

[58] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[59] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proceedings of the 29th International Conference on Very Large Data Bases-Volume 29*. Berlin, Germany: VLDB Endowment, 2003, pp. 81–92.

[60] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, ser. VLDB '04. Toronto, Canada: VLDB Endowment, 2004, pp. 180–191.

[61] T. J. Sejnowski and C. R. Rosenberg, "Parallel networks that learn to pronounce english text," *Complex systems*, vol. 1, no. 1, pp. 145–168, 1987.

[62] E. Carden and J. Brownjohn, "Arma modelled time-series classification for structural health monitoring of civil infrastructure," *Mechanical systems and signal processing*, vol. 22, no. 2, pp. 295–314, 2008.

[63] P. Omenzetter and J. M. W. Brownjohn, "Application of time series analysis for bridge monitoring," *Smart Materials and Structures*, vol. 15, no. 1, pp. 129–138, jan 2006.

[64] J.-B. Bodeux and J.-C. Golinval, "Application of armav models to the identification and damage detection of mechanical and civil engineering structures," *Smart materials and structures*, vol. 10, no. 3, p. 479, 2001.

[65] P. Omenzetter and J. M. W. Brownjohn, "Application of time series analysis for bridge monitoring," *Smart Materials and Structures*, vol. 15, no. 1, p. 129, 2006.

[66] B. Peeters and G. De Roeck, "One-year monitoring of the z 24-bridge: environmental effects versus damage events," *Earthquake Engineering and Structural Dynamics*, vol. 30, no. 2, pp. 149–171, 2001.

[67] B. Peeters, J. Maeck, and G. De Roeck, "Dynamic monitoring of the z24-bridge: separating temperature effects from damage," in *Proceedings of the European COST F3 Conference on System Identification and Structural Health Monitoring*, Madrid, Spain, 2000, pp. 377–386.

[68] I. Tien, M. Pozzi, and A. Der Kiureghian, "Probabilistic framework for assessing maximum structural response based on sensor measurements," *Structural Safety*, vol. 61, pp. 43–56, 2016.

[69] M. Wu and A. W. Smyth, "Application of the unscented kalman filter for real-time nonlinear structural system identification," *Structural Control and Health Monitoring*, vol. 14, no. 7, pp. 971–990, 2007.

[70] E. N. Chatzi and A. W. Smyth, "The unscented kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing," *Structural Control and Health Monitoring*, vol. 16, no. 1, pp. 99–123, 2009.

[71] E. N. Chatzi and A. W. Smyth, "Particle filter scheme with mutation for the estimation of time-invariant parameters in structural health monitoring applications," *Structural Control and Health Monitoring*, vol. 20, no. 7, pp. 1081–1095, 2013.

[72] A. Olivier and A. W. Smyth, "On the performance of online parameter estimation algorithms in systems with various identifiability properties," *Frontiers in Built Environment*, vol. 3, p. 14, 2017.

[73] A. Olivier and A. W. Smyth, "Particle filtering and marginalization for parameter identification in structural systems," *Structural Control and Health Monitoring*, vol. 24, no. 3, p. e1874, 2017.

[74] F. Cadini, C. Sbarufatti, M. Corbetta, and M. Giglio, "A particle filter-based model selection algorithm for fatigue damage identification on aeronautical structures," *Structural Control and Health Monitoring*, vol. 24, no. 11, p. e2002, 2017.

[75] K.-V. Yuen and K. Huang, "Real-time substructural identification by boundary force modeling," *Structural Control and Health Monitoring*, vol. 25, no. 5, p. e2151, 2018.

[76] H.-Q. Mu, S.-C. Kuok, and K.-V. Yuen, "Stable robust extended kalman filter," *Journal of Aerospace Engineering*, vol. 30, no. 2, p. B4016010, 2016.

[77] J.-A. Goulet, "Bayesian dynamic linear models for structural health monitoring," *Structural Control and Health Monitoring*, vol. 24, pp. e2035–n/a, 2017.

[78] I. Solhjell, "Bayesian forecasting and dynamic models applied to strain data from the göta river bridge," Master's thesis, University of Oslo, Oslo, Norway, 2009.

[79] M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models*, ser. Springer Series in Statistics. New York, USA: Springer, 1999.

[80] M. West, "Bayesian dynamic modelling," *Bayesian Inference and Markov Chain Monte Carlo: In Honour of Adrian FM Smith*, pp. 145–166, 2013.

[81] C.-J. Kim, C. R. Nelson *et al.*, "State-space models with regime switching: classical and gibbs-sampling approaches with applications," *MIT Press Books*, vol. 1, 1999.

[82] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. New Jersey, USA: Wiley, 2006.

[83] Y. Ni, X. Hua, K. Fan, and J. Ko, "Correlating modal properties with temperature using long-term monitoring data and support vector machine technique," *Engineering Structures*, vol. 27, no. 12, pp. 1762 – 1773, 2005.

[84] K.-V. Yuen and S.-C. Kuok, "Ambient interference in long-term monitoring of buildings," *Engineering Structures*, vol. 32, no. 8, pp. 2379 – 2386, 2010.

[85] K.-V. Yuen and S.-C. Kuok, "Modeling of environmental influence in structural health assessment for reinforced concrete buildings," *Earthquake Engineering and Engineering Vibration*, vol. 9, no. 2, pp. 295–306, Jun 2010.

[86] K. P. Murphy, "Switching kalman filters," Citeseer, Tech. Rep., 1998.

[87] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software.* New York, USA: John Wiley & Sons, 2004.

[88] H. A. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[89] A. R. Runnalls, "Kullback-leibler approach to gaussian mixture reduction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 989–999, 2007.

[90] S. L. Lauritzen, *Graphical models.* New York, USA: Oxford University Press, 1996.

[91] R. R. Labbe, *Kalman and Bayesian Filters in Python*, 2018. [Online]. Available: https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python

[92] S. McKinley and M. Levine, "Cubic spline interpolation," *College of the Redwoods*, vol. 45, no. 1, pp. 1049–1060, 1998.

[93] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Cambridge, MA, USA: MIT Press, 2001.

[94] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Found. Trends Mach. Learn.*, vol. 4, no. 3, pp. 195–266, Mar. 2012.

[95] D. J. Henderson and C. F. Parmeter, *Applied Nonparametric Econometrics.* New York, USA: Cambridge University Press, 2015.

[96] D. Duvenaud, "Automatic model construction with gaussian processes," Ph.D. dissertation, University of Cambridge, Cambridge, UK, 2014.

[97] S. S. Rao, "A course in time series analysis," Technical report, Texas A&M University, Tech. Rep., 2008.

[98] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*, 3rd ed. Boca Raton, USA: CRC Press, 2014.

[99] M. V. Shcherbakov, A. Brebels, N. L. Shcherbakova, A. P. Tyukov, T. A. Janovsky, and V. A. Kamaev, "A survey of forecast error measures," *World Applied Sciences Journal*, vol. 24, no. 2013, pp. 171–176, 2013.

[100] A. Gelman, J. Hwang, and A. Vehtari, "Understanding predictive information criteria for bayesian models," *Statistics and computing*, vol. 24, no. 6, pp. 997–1016, 2014.

[101] J.-A. Goulet and K. Koo, "Empirical validation of bayesian dynamic linear models in the context of structural health monitoring," *Journal of Bridge Engineering*, vol. 23, no. 2, p. 05017017, 2018.

[102] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in Neural Information Processing Systems*, 2014, pp. 2933–2941.

[103] J. Martens, "Deep learning via Hessian-free optimization." in *Proceedings of the 27th international conference on machine learning*, Haifa, Israel, 2010.

[104] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[105] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *lecture 6a from University of Toronto*, 2012.

[106] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[107] T. Schaul, S. Zhang, and Y. LeCun, "No more pesky learning rates," in *International Conference on Machine Learning*, Atlanta, GA, USA, 2013, pp. 343–351.

[108] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. Cambridge, MA, USA: MIT Press, 2009.

[109] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[110] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *The annals of applied statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[111] R. M. Neal, "Mcmc using hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 2, no. 11, 2011.

[112] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid monte carlo," *Physics letters B*, vol. 195, no. 2, pp. 216–222, 1987.

[113] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*. Boca Raton, USA: CRC press, 2011.

[114] M. Betancourt, "A conceptual introduction to hamiltonian monte carlo," *arXiv preprint arXiv:1701.02434*, 2017.

[115] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1593–1623, 2014.

[116] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical programming*, vol. 120, no. 1, pp. 221–259, 2009.

[117] A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Statistical science*, pp. 457–472, 1992.

[118] A. Gelman, D. Lee, and J. Guo, "Stan: A probabilistic programming language for bayesian inference and optimization," *Journal of Educational and Behavioral Statistics*, vol. 40, no. 5, pp. 530–543, 2015.

[119] A. Gelman, D. Simpson, and M. Betancourt, "The prior can often only be understood in the context of the likelihood," *Entropy*, vol. 19, no. 10, p. 555, 2017.

[120] A. W. Bowman and A. Azzalini, *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*. New York: OUP Oxford, 1997, vol. 18.

[121] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. California, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 176–183.

[122] S. Särkkä, *Bayesian Filtering and Smoothing*. New York, USA: Cambridge University Press, 2013.

[123] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[124] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[125] C. Nemeth, P. Fearnhead, and L. Mihaylova, "Sequential monte carlo methods for state and parameter estimation in abruptly changing environments," *arXiv preprint arXiv:1510.02604*, 2015.

[126] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo methods in practice*. New York, USA: Springer, 2001, pp. 197–223.

[127] M. Toledano, I. Cohen, Y. Ben-Simhon, and I. Tadeski, "Real-time anomaly detection system for time series at scale," in *KDD 2017 Workshop on Anomaly Detection in Finance*, Halifax, Nova Scotia, 2018, pp. 56–65.

[128] W. Dai, B. Liu, X. Meng, and D. Huang, "Spatio-temporal modelling of dam deformation using independent component analysis," *Survey review*, vol. 46, no. 339, pp. 437–443, 2014.

[129] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[130] X. Cheng, R. Zhang, J. Zhou, and W. Xu, "Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting," in *2018 International Joint Conference on Neural Networks (IJCNN)*. Rio, Brazil: IEEE, 2018, pp. 1–8.

[131] R. Asadi and A. Regan, "A spatial-temporal decomposition based deep neural network for time series forecasting," *arXiv preprint arXiv:1902.00636*, 2019.

[132] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

APPENDIX A MODEL MATRICES

This appendix presents the mathematical formulations of the transition matrix ($\mathbf{A}_t$), the observation matrix ($\mathbf{C}_t$), the observation error covariance matrix ($\mathbf{R}_t$), and the model error covariance matrix ($\mathbf{Q}_t$) for the models being used in this dissertation. Let us introduce the notation

$$
\begin{aligned}
\left[\tilde{\boldsymbol{k}}_t^{\mathtt{KR}}\right]_{1\times n} &= [\tilde{k}_1^{\mathtt{KR}}\ \tilde{k}_2^{\mathtt{KR}} \ldots \tilde{k}_n^{\mathtt{KR}}]_t && \text{normalized kernel value vector} \\
\mathbf{I}_n &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} && \text{n-by-n identity matrix} \\
\mathbf{0}_{n\times m} &= \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} && \text{n-by-m zero matrix} \\
\Delta t &= \text{timestamp}_t - \text{timestamp}_{t-1} && \text{time-step length.}
\end{aligned}
\tag{A.1}
$$

A.1 Model #1

$$
\begin{aligned}
\mathbf{A}_t &= \text{block diag}\left(\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & \left[\tilde{\boldsymbol{k}}_t^{\mathtt{KR}}\right]_{1\times 6} \\ \mathbf{0}_{1\times 6} & \mathbf{I}_6 \end{bmatrix}, \phi^{\mathtt{AR}}\right) \\
\mathbf{C}_t &= [1\ 0\ 1\ \mathbf{0}_{1\times 6}\ 1] \\
\mathbf{R}_t &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t &= \text{block diag}\left(\mathbf{0}_{2\times 2}, \mathbf{0}_{6\times 6}, (\sigma_w^{\mathtt{AR}})^2\right).
\end{aligned}
\tag{A.2}
$$

A.2 Model #2

$$
\begin{aligned}
\mathbf{A}_t &= \text{block diag}\left(\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, 1, \phi^{\mathtt{AR}}\right) \\
\mathbf{C}_t &= [1\ 0\ h(t,\mathcal{D})\ 1] \\
\mathbf{R}_t &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t &= \text{block diag}\left(\mathbf{0}_{2\times 2}, 0, (\sigma^{\mathtt{AR}})^2\right)
\end{aligned}
\tag{A.3}
$$

A.3 Model #3

$$
\begin{aligned}
\mathbf{A}_t &= \text{block diag}\left(\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos\omega^{\mathtt{P}_1} & \sin\omega^{\mathtt{P}_1} \\ -\sin\omega^{\mathtt{P}_1} & \cos\omega^{\mathtt{P}_1} \end{bmatrix}, \begin{bmatrix} \cos\omega^{\mathtt{P}_2} & \sin\omega^{\mathtt{P}_2} \\ -\sin\omega^{\mathtt{P}_2} & \cos\omega^{\mathtt{P}_2} \end{bmatrix}, \phi^{\mathtt{AR}}\right) \\
\mathbf{C}_t &= [1\ 0\ 1\ 0\ 1\ 0\ 1] \\
\mathbf{R}_t &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t &= \text{block diag}\left(\mathbf{0}_{2\times 2}, \mathbf{0}_{2\times 2}, \mathbf{0}_{2\times 2}, (\sigma^{\mathtt{AR}})^2\right),
\end{aligned}
\tag{A.4}
$$

where ω is the angular frequency with the periods of $\mathtt{P}_1 = 365.24$ days and $\mathtt{P}_2 = 182.62$ days.

A.4 Model #4

$$
\begin{aligned}
\mathbf{A}_t &= \text{block diag}\left(1, \begin{bmatrix} 0 & \left[\tilde{\boldsymbol{k}}_t^{\mathtt{KR}}\right]_{1\times 100} \\ \mathbf{0}_{100\times 1} & \mathbf{I}_{100} \end{bmatrix}, \phi^{\mathtt{AR}}\right) \\
\mathbf{C}_t &= [1\ 1\ \mathbf{0}_{1\times 101}\ 1] \\
\mathbf{R}_t &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t &= \text{block diag}\left((\sigma_w^{\mathtt{B}})^2, \begin{bmatrix} \left(\sigma_{w,0}^{\mathtt{KR}}\right)^2 & \mathbf{0}_{1\times 100} \\ \mathbf{0}_{100\times 1} & \left(\sigma_{w,1}^{\mathtt{KR}}\right)^2 \cdot \mathbf{I}_{100} \end{bmatrix}, (\sigma_w^{\mathtt{AR}})^2\right).
\end{aligned}
\tag{A.5}
$$

A.5 Model #5

A.5.1 Normal model class

$$
\begin{aligned}
\mathbf{A}_t^1 &= \text{block diag}\left(\begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & \left[\tilde{\boldsymbol{k}}_t^{\mathtt{KR}}\right]_{1\times 10} \\ \mathbf{0}_{10\times 1} & \mathbf{I}_{10} \end{bmatrix}, \phi^{\mathtt{AR}}\right) \\
\mathbf{C}_t^1 &= [1\ 0\ 0\ \mathbf{0}_{1\times 11}\ 1] \\
\mathbf{R}_t^1 &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t^{1(1)} &= \text{block diag}\left(\mathbf{0}_{3\times 3}, \mathbf{0}_{11\times 11}, (\sigma^{\mathtt{AR}})^2\right) \\
\mathbf{Q}_t^{2(1)} &= \text{block diag}\left(\mathbf{0}_{3\times 3}, \mathbf{0}_{11\times 11}, (\sigma^{\mathtt{AR}})^2\right).
\end{aligned}
\tag{A.6}
$$

A.5.2 Abnormal model class

$$
\begin{aligned}
\mathbf{A}_t^2 &= \text{block diag}\left(\begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & \left[\tilde{\boldsymbol{k}}_t^{\mathtt{KR}}\right]_{1\times 10} \\ \mathbf{0}_{10\times 1} & \mathbf{I}_{10} \end{bmatrix}, \phi^{\mathtt{AR}}\right) \\
\mathbf{C}_t^1 &= [1\ 0\ 0\ \mathbf{0}_{1\times 11}\ 1] \\
\mathbf{R}_t^2 &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t^{1(2)} &= \text{block diag}\left(\mathbf{0}_{3\times 3}, \mathbf{0}_{11\times 11}, (\sigma^{\mathtt{AR}})^2\right) \\
\mathbf{Q}_t^{2(2)} &= \text{block diag}\left((\sigma^{\mathtt{LA}})^2 \cdot \begin{bmatrix} \frac{\Delta t^2}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}, \mathbf{0}_{11\times 11}, (\sigma^{\mathtt{AR}})^2\right).
\end{aligned}
\tag{A.7}
$$

A.6 Model #6

$$
\begin{aligned}
\mathbf{A}_t &= \text{block diag}\left(1, \phi^{\mathtt{AR,D}}, 1, \begin{bmatrix} \cos\omega^{\mathtt{P}_1} & \sin\omega^{\mathtt{P}_1} \\ -\sin\omega^{\mathtt{P}_1} & \cos\omega^{\mathtt{P}_1} \end{bmatrix}, \begin{bmatrix} \cos\omega^{\mathtt{P}_2} & \sin\omega^{\mathtt{P}_2} \\ -\sin\omega^{\mathtt{P}_2} & \cos\omega^{\mathtt{P}_2} \end{bmatrix}, \phi^{\mathtt{AR,T}}\right) \\
\mathbf{C}_t &= \begin{bmatrix} 1 & 1 & 0 & \beta_1^{\mathtt{R}} & 0 & \beta_2^{\mathtt{R}} & 0 & \beta_3^{\mathtt{R}} \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \\
\mathbf{R}_t &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t &= \text{block diag}\left(0, (\sigma^{\mathtt{AR,D}})^2, 0, \mathbf{0}_{2\times 2}, \mathbf{0}_{2\times 2}, (\sigma^{\mathtt{AR,T}})^2\right),
\end{aligned}
\tag{A.8}
$$

where ω is the angular frequency with the periods of $\mathtt{P}_1 = 1$ day and $\mathtt{P}_2 = 365.24$ days.

A.7 Model #7

$$
\begin{aligned}
\mathbf{A}_t &= \text{block diag}\left(1, \begin{bmatrix} \cos\omega^{\mathtt{P}_1} & \sin\omega^{\mathtt{P}_1} \\ -\sin\omega^{\mathtt{P}_1} & \cos\omega^{\mathtt{P}_1} \end{bmatrix}, \phi^{\mathtt{AR}}\right) \\
\mathbf{C}_t &= [1\ 1\ 0\ 1] \\
\mathbf{R}_t &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t &= \text{block diag}\left(1, \mathbf{0}_{2\times 2}, (\sigma^{\mathtt{AR}})^2\right),
\end{aligned}
\tag{A.9}
$$

where ω is the angular frequency with the periods of $\mathtt{P}_1 = 365.24$ days.

A.8 Model #8

$$
\begin{aligned}
\mathbf{A}_t &= \text{block diag}\left(\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos\omega^{\mathtt{P}_1} & \sin\omega^{\mathtt{P}_1} \\ -\sin\omega^{\mathtt{P}_1} & \cos\omega^{\mathtt{P}_1} \end{bmatrix}, \phi^{\mathtt{AR}}\right) \\
\mathbf{C}_t &= [1\ 0\ 1\ 0\ 1] \\
\mathbf{R}_t &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t &= \text{block diag}\left(\mathbf{0}_{2\times 2}, \mathbf{0}_{2\times 2}, (\sigma^{\mathtt{AR}})^2\right),
\end{aligned}
\tag{A.10}
$$

where ω is the angular frequency with the periods of $\mathtt{P}_1 = 365.24$ days.

A.9 Model #9

A.9.1 Normal model class

$$
\begin{aligned}
\mathbf{A}_t^1 &= \text{block diag}\left(\begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & \left[\tilde{\boldsymbol{k}}_t^{\mathtt{KR}}\right]_{1\times 10} \\ \mathbf{0}_{10\times 1} & \mathbf{I}_{10} \end{bmatrix}\right) \\
\mathbf{C}_t^1 &= [1\ 0\ 0\ \mathbf{0}_{1\times 11}] \\
\mathbf{R}_t^1 &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t^{1(1)} &= \text{block diag}\left(\mathbf{0}_{3\times 3}, \mathbf{0}_{11\times 11}\right) \\
\mathbf{Q}_t^{2(1)} &= \text{block diag}\left(\mathbf{0}_{3\times 3}, \mathbf{0}_{11\times 11}\right).
\end{aligned}
\tag{A.11}
$$

A.9.2 Abnormal model class

$$
\begin{aligned}
\mathbf{A}_t^2 &= \text{block diag}\left(\begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & \left[\tilde{\boldsymbol{k}}_t^{\mathtt{KR}}\right]_{1\times 10} \\ \mathbf{0}_{10\times 1} & \mathbf{I}_{10} \end{bmatrix}\right) \\
\mathbf{C}_t^1 &= [1\ 0\ 0\ \mathbf{0}_{1\times 11}] \\
\mathbf{R}_t^2 &= \left[(\sigma_v)^2\right] \\
\mathbf{Q}_t^{1(2)} &= \text{block diag}\left(\mathbf{0}_{3\times 3}, \mathbf{0}_{11\times 11}\right) \\
\mathbf{Q}_t^{2(2)} &= \text{block diag}\left(\left(\sigma^{\mathtt{LA}}\right)^2 \cdot \begin{bmatrix} \frac{\Delta t^2}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}, \mathbf{0}_{11\times 11}\right).
\end{aligned}
\tag{A.12}
$$

APPENDIX B OPTIMIZATION ALGORITHMS FOR BDLMS

This appendix presents the implementation of the algorithms for optimizing the model parameters in BDLMs. Let us introduction the notation

`sum`	sum of array elements
`cumsum`	cumulative sum operation of a vector
`rand`	uniformly distributed random numbers
`find`	find indices and values of nonzero elements
`min`	minimum elements of an array
`all`	determine if all array elements are nonzero
`round`	round to nearest integer
`repmat`	repeat copies of array
`diag`	get diagonal elements of matrix

(B.1)

B.1 Evaluation of First and Second Derivatives

This section presents the algorithm used for evaluating the first and second derivatives of the function $\mathcal{F}$ with respect to the pth element of a vector of model parameters $\boldsymbol{\theta}$. These derivatives are evaluated numerically using the central differentiation method.

Algorithm 2: Evaluation of gradient and Hessian

function `evalGradientHessian`$(\mathcal{F}, \boldsymbol{\theta}, \mathbf{y}_{1:t}, \delta_{\boldsymbol{\theta}}, p)$

$\boldsymbol{\theta}^{+}(p) = \boldsymbol{\theta}(p) + \delta_{\boldsymbol{\theta}}$; $\boldsymbol{\theta}^{-}(p) = \boldsymbol{\theta}(p) - \delta_{\boldsymbol{\theta}}$;

$g_p = \frac{\mathcal{F}(\boldsymbol{\theta}^{+}, \mathbf{y}_{1:t}) - \mathcal{F}(\boldsymbol{\theta}^{-}, \mathbf{y}_{1:t})}{2\delta_{\boldsymbol{\theta}}}$; $h_p = \frac{\mathcal{F}(\boldsymbol{\theta}^{+}, \mathbf{y}_{1:t}) - 2\mathcal{F}(\boldsymbol{\theta}, \mathbf{y}_{1:t}) + \mathcal{F}(\boldsymbol{\theta}^{-}, \mathbf{y}_{1:t})}{(\delta_{\boldsymbol{\theta}})^2}$

B.2 Model parameter update

This section presents the algorithm including five optimizers for updating the model parameters.

Algorithm 3: Model parameter update

function `parameterUpdate`$(g_p, h_p, \boldsymbol{\theta}^{n-1}, \mathbf{r}^{n-1}, \boldsymbol{\nu}^{n-1}, \mathbf{s}^{n-1}, \bar{\mathbf{h}}^{n-1}, \epsilon, \beta, \beta_1, \beta_2, \eta, p, optimizer)$

  if $optimizer =$ *NR* **then**

    $\boldsymbol{\theta}^n(p) = \boldsymbol{\theta}^{n-1}(p) + (h_p)^{-1} g_p;$

  else if $optimizer =$ *MMT* **then**

    $\boldsymbol{\theta}^n(p) = \boldsymbol{\theta}^{n-1}(p) + \beta \mathbf{r}^{n-1} + \eta g_p;$

  else if $optimizer =$ *RMSprop* **then**

    $\boldsymbol{\nu}^n(p) = \beta_1 \boldsymbol{\nu}^{n-1}(p) + (1-\beta_1)(g_p)^2;$

    $\boldsymbol{\theta}^n(p) = \boldsymbol{\theta}^{n-1}(p) + \eta \left(\sqrt{\boldsymbol{\nu}^n(p)} + \epsilon\right)^{-1} g_p;$

  else if $optimizer =$ *ADAM* **then**

    $\boldsymbol{\nu}^n(p) = \beta_1 \boldsymbol{\nu}^{n-1}(p) + (1-\beta_1)(g_p)^2;$

    $\mathbf{s}^n(p) = \beta_2 \mathbf{s}^{n-1}(p) + (1-\beta_2) g_p;$

    $\hat{\boldsymbol{\nu}}^n(p) = \frac{\boldsymbol{\nu}^n}{1-(\beta_1)^n};$

    $\hat{\mathbf{s}}^n(p) = \frac{\mathbf{s}^n}{1-(\beta_2)^n};$

    $\boldsymbol{\theta}^n(p) = \boldsymbol{\theta}^{n-1}(p) + \eta \hat{\mathbf{s}}^n(p) \left(\sqrt{\hat{\boldsymbol{\nu}}^n(p)} + \epsilon\right)^{-1};$

  else if $optimizer =$ *AMMT* **then**

    $\mathbf{s}^n(p) = \beta \mathbf{s}^{n-1}(p) + (1-\beta) g_p;$

    $\bar{\mathbf{h}}^n(p) = \beta \bar{\mathbf{h}}^{n-1}(p) + (1-\beta) h_p;$

    $\boldsymbol{\theta}^n(p) = \boldsymbol{\theta}^{n-1}(p) + (\bar{\mathbf{h}}^n(p))^{-1} \mathbf{s}^n(p);$

B.3 Objective Function

This section presents the algorithm used for evaluating the log-likelihood function.

Algorithm 4: Objective function

function `objectiveFun`$(\mathcal{F}, \boldsymbol{\theta}, \mathbf{y}, \mathtt{T}_{tr}, \mathtt{T})$

  $L_{tr} = \mathcal{F}(\boldsymbol{\theta}, \mathbf{y}_{1:\mathtt{T}_{tr}});$

  if $\mathtt{T}_{tr} < \mathtt{T}$ **then**

    $L_v = \mathcal{F}(\boldsymbol{\theta}, \mathbf{y}_{\mathtt{T}_{tr}+1:\mathtt{T}});$

  else

    $L_v = NaN;$

B.4 Index Selection

The section presents the algorithm for selecting the model parameter to be updated for the next epoch.

Algorithm 5: Selection of the index for the next epoch

```
function getIndex(d)
    rs = rand;  p_d = cumsum(d)/sum(d);  p_r = p_d − rs;  p_r(p_r < 0) = +∞;
    p = find(p_r == min(p_r), 1, 'first')
```

B.5 Convergence Check

This section presents the stopping criteria for the batch gradient ascent (BGA) and mini-batch gradient ascent (MGA) algorithms. For this purpose, Convergence check 1 is used for the BGA algorithm and Convergence check 2 is employed for the MGA algorithm.

Algorithm 6: Convergence check 1

```
function convergenceCheck_1(L^n, L^{n−1}, c, n, N_maxEpoch, tol, p)
    cond_1 = L^n > L^{n−1} & |(L^n − L^{n−1}) / L^{n−1}| < tol;
    if cond_1 or n = N_maxEpoch then
        c(p) = 1;
        converged = all(c);
    else
        c(p) = 0;
```

Algorithm 7: Convergence check 2

```
function convergenceCheck_2(L_v^n, L_v^{n−1}, n, N_maxEpoch, tol)
    if L_v^n < tol · L_v^{n−1} or n = N_maxEpoch then
        converged = 1;
    else
        converged = 0;
```

B.6 Batch Gradient Ascent

This section presents the details of the batch gradient ascent algorithm implemented for BDLMs.

Algorithm 8: Batch Gradient Ascent (BGA)

```
function BGA(F, θ^0, y_{1:T}, δ_θ, N_maxEpoch, ϵ, β, β_1, β_2, tol, optimizer)
    n = 0; L^0 = F(θ^0); T_tr = T;
    while converged do
        n = n + 1;
        Find p = getIndex(d);
        Compute g_p, h_p = evalGradientHessian(F, θ^{n-1}, y_{1:T_t}, δ_θ, p);
        θ^n = parameterUpdate(g_p, h_p, θ^{n-1}, r^{n-1}, ν^{n-1}, s^{n-1}, h̄^{n-1}, ϵ, β, β_1, β_2, η, p, optimizer);
        L^n_tr = objectiveFun(F, θ^n, y_{1:T}, T_tr, T);
        if L^n_tr > L^{n-1}_tr then
            d(i) = L^n_tr − L^{n-1}_tr;
        else
            θ^n = θ^{n-1}; L^n_tr = L^{n-1}_tr;
        converged, c = convergenceCheck_1(L^n_tr, L^{n-1}_tr, c, n, N_maxEpoch, tol, i);
```

B.7 Mini-Batch Gradient Ascent

This section presents the details of the mini-batch gradient ascent algorithm implemented for BDLMs.

Algorithm 9: Mini-Batch Gradient Ascent (MGA)

```
function MGA(θ_0, y_{1:T}, F, δ_θ, N_maxEpoch, ϵ, β, β_1, β_2, τ, tol, λ, optimizer)
    n = 0; L^0 = F(θ_0, y_{1:T}); N_maxM = round(T/l_MB); p = length(θ_0);
    M_θ = repmat(θ_0, [1, p]); T_tr = round(τT);
    while converged do
        n = n + 1;  m = 0; θ^{m-1} = θ^{n-1};
        while m < N_maxM do
            m = m + 1;
            Shuffle t_s ∈ [1, T − l_MB];
            for p = 1 : p do
                Compute  g_p, h_p = evalGradientHessian (F, θ^{m-1}, y_{t_s:t_s+l_MB}, δ_θ, p);
                M_θ(p, p) = parameterUpdate(g_p, h_p, θ^{n-1}, r^{n-1}, ν^{n-1}, s^{n-1}, h̄^{n-1}, ϵ, β, β_1, β_2, η, p, optimizer);
            M_θ = [M_θ, diag(M_θ)];
            for p = 1 : p + 1 do
                t_tr = round(τ l_MB);
                L_v^m(p) = objectiveFun(F, M_θ(:, p), y_{t_s:t_s+l_MB}, t_tr, t_s + l_MB);
            Find index p_max corresponding to the maximal value of L_tr^m;
            θ^m = M_θ(:, p_max);
        L_tr^n, L_v^n = objectiveFun(F, θ^m, y_{1:T}, T_tr, T);
        if L_tr^n > λ L_tr^{n-1} then
            θ^n = θ^m;
        else
            θ^n = θ^{n-1}; L_v^n = L_v^{n-1};
        converged = convergenceCheck_2 (L_v^n, L_v^{n-1}, n, N_maxEpoch, tol);
```

APPENDIX C MEASURE OF FORECAST ACCURACY

The mathematical formulations for the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Log Predictive Density (LPD) are written as following

$$\begin{aligned} \text{MAE} &= \frac{1}{\mathtt{T}} \sum_{t=1}^{\mathtt{T}} |\mu_{y_t} - y_t| \\ \text{RMSE} &= \sqrt{\frac{\sum_{t=1}^{\mathtt{T}} (\mu_{y_t} - y_t)^2}{\mathtt{T}}}, \\ \text{LPD} &= \sum_{t=1}^{\mathtt{T}} \ln \mathcal{N}(y_t; \mu_{y_t}, \sigma_{y_t}^2 + \sigma_v^2), \end{aligned} \tag{C.1}$$

where μ_{y_t} is the forecast value, y_t is the observation at time t, $\sigma_{\hat{y}_t}$ is the model-prediction-error standard deviation, and σ_v is the observation error standard deviation.