POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Analytical Bayesian Parameter Inference for Probabilistic Models with Engineering Applications

BHARGOB DEKA
Département de génie civil, géologique et des mines

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
génie civil

September 2022

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Analytical Bayesian Parameter Inference for Probabilistic Models with Engineering Applications

présentée par **Bhargob DEKA**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Richard GOURDEAU, président
James-A. GOULET, membre et directeur de recherche
Andrew W. SMYTH, membre externe
François LEDUC-PRIMEAU, membre

DEDICATION

To my parents, Rina Deka, and Tankeswar Deka
To my support, Sophie Lampron-de Souza
To my friends, family, and colleagues...

I send your way love & admiration.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my academic advisor Prof. James-A. Goulet, for his continuous support and guidance throughout this thesis.

I would also like to thank and acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC), Hydro-Québec (HQ), Hydro-Québec's Research Institute (IREQ), and Institute For Data Valorization (IVADO) for the financial support of this research. I would like to further thank Hydro-Québec for providing the datasets used in this thesis. Many thanks to Benjamin Miquel, Vincent Roy, and Patrice Côté from Hydro-Québec for their suggestions and support 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. A special thanks to Luong Ha Nguyen, Dai Vuong Van and Saeid Amiri who have personally helped me in this thesis. I would also like to acknowledge Zachary Hamida and Shervin Khazaeli, with whom I had endless discussions about various research and life topics, which kept me motivated throughout my work. Last but not the least, I would like to thank Ianis Gaudot who helped me get familiar with the environment at Polytechnique when I first joined the research group.

Last but not the least, I would like to thank my family for their unconditional support throughout my life. To my parents who encouraged me to pursue higher education. To my girlfriend Sophie, who has always been by my side throughout these years. To my friends here and back home who have always shown me love and support, either in person or online.

RÉSUMÉ

Les problèmes d'ingénierie reposent sur des modèles pour prédire les phénomènes physiques et il est essentiel, à des fins de prise de décision, que ces modèles soient probabilistes, afin que nous soyons conscients de ce que nous ne savons pas à leur sujet. Les approches probabilistes courantes incluent les modèles d'espace d'états utilisés pour prévoir les séries chronologiques, et les réseaux de neurones Bayésiens utilisés pour effectuer des tâches de régression. De tels modèles impliquent des paramètres inconnus non seulement pour modéliser des phénomènes physiques mais aussi pour quantifier les incertitudes épistémiques et aléatoires du modèle. En pratique, l'estimation de ces paramètres peut être exigeante en termes de calcul, ce qui empêche les modèles existants d'être mis à l'échelle pour être utilisés dans des applications d'ingénierie pratiques à grande échelle.

Par exemple, dans les modèles d'espace d'états, l'estimation des variables d'état cachées est peu coûteuse en termes de calcul car nous pouvons nous appuyer sur une formulation analytique pour effectuer l'inférence Bayésienne. D'autre part, l'incertitude aléatoire est quantifiée par les paramètres de variance dans les matrices de covariance des erreurs de processus ($\mathbf{Q}$) et d'observation ($\mathbf{R}$), qui doivent être connues avec précision pour une estimation exacte des variables cachés. L'obtention d'estimations optimales pour ces paramètres inconnus est généralement la tâche la plus exigeante en termes de calcul dans la procédure d'estimation d'états cachés. Même si la matrice $\mathbf{R}$ peut être, dans de nombreuses situations, considérée comme connue à partir des spécifications de l'instrument de mesure, il reste toujours un défi de développer une méthode de calcul efficace capable d'effectuer une estimation en ligne de forme fermée de la matrice $\mathbf{Q}$ pour plusieurs séries chronologiques. De plus, l'inférence en ligne traitable analytiquement ne peut pas être effectuée pour les modèles d'espace d'états multiplicatifs qui permettraient de déduire les paramètres du modèle en tant qu'états cachés à l'aide d'expressions algébriques de forme fermée. D'autre part, l'inférence de paramètres analytiques peut être effectuée dans des réseaux de neurones Bayésiens en utilisant la method tractable approximate Gaussienne Inference (TAGI), mais est uniquement limitée à la modélisation de l'incertitude aléatoire homoscédastique.

Cette thèse contribue à développer des méthodes Bayésiennes analytiques traitables pour l'inférence de paramètres afin d'améliorer les performances et l'évolutivité des modèles probabilistes dans le contexte des applications d'ingénierie. Les principales contributions sont : a) une méthode analytique applicable aux modèles d'espace d'états multiplicatifs qui tire parti de l'approximation multiplicative Gaussienne (GMA), b) une méthode de régression

qui s'appuie sur GMA pour modéliser la dépendance non linéaire entre deux séries temporelles et permettant l'estimation en ligne du coefficient de régression en fonction des valeurs des variables d'états ainsi que des séries chronologiques interdépendantes, c) une méthode d'inférence Bayésienne analytique appelée l'inférence approximative de la variance Gaussienne (AGVI) qui permet d'effectuer une estimation en ligne sous forme fermée du terme de variance de l'erreur de processus univariée, d) l'extension de la méthode AGVI pour déduire les paramètres de variance de l'erreur de processus multivariée, et e) l'application de l'AGVI pour modéliser analytiquement l'incertitude aléatoire hétéroscédastique pour les tâches de régression utilisant TAGI pour les réseau neuronal Bayésien. Les méthodes proposées sont vérifiées avec des données synthétiques et validées avec des ensembles de données réels pour les applications de surveillance de la santé structurelle ainsi que des ensembles de données de régression de référence. Les méthodes proposées dans cette thèse dépassent les performances des approches existantes en termes de capacité prédictive tout en étant jusqu'à des ordres de grandeur plus rapides.

ABSTRACT

Engineering problems rely on models to predict physical phenomena and it is critical for decision-making purposes that these models be probabilistic, so that we are aware about what we do not know about them. Common probabilistic approaches include state-space models that are used for forecasting time series and Bayesian neural networks that are used for performing regression tasks. Such models involve unknown parameters for not only modeling physical phenomena but also for quantifying the model's epistemic and aleatory uncertainties. In practice, estimating these parameters can be computationally demanding, so that it prevents existing models from being scaled up to be used in large-scale practical engineering applications.

For instance, in state-space models, estimating hidden state variables is computationally cheap because we can rely on an analytical formulation for performing Bayesian inference. On the other hand, the aleatory uncertainty is quantified by the variance parameters in the process ($\mathbf{Q}$) and observation ($\mathbf{R}$) error covariance matrices, which need to be known accurately for an exact state estimation. Obtaining optimal estimates for these unknown parameters is typically the most computationally demanding task in the state estimation procedure. Even though in many situations the matrix $\mathbf{R}$ can be considered to be known from the measuring instrument specifications, it still remains a challenge to develop a computationally efficient online method which is able to perform the closed-form Bayesian estimation of the matrix $\mathbf{Q}$ for multiple time series. Moreover, analytically tractable online inference cannot be carried out for multiplicative state-space models which would allow model parameters to be inferred as hidden states using closed-form algebraic expressions. On the other hand, analytical parameter inference can be carried out in Bayesian neural networks using the tractable approximate Gaussian inference (TAGI) but is restricted to modeling only homoscedastic aleatory uncertainty.

The contribution of this thesis is to develop analytically tractable Bayesian methods for parameter inference in order to improve the performance and scalability of probabilistic models in the context of engineering applications. The main contributions are: a) an analytical method applicable to multiplicative state-space models that takes advantage of the Gaussian multiplicative approximation (GMA), b) a state-based regression method that builds upon GMA to model the nonlinear dependency between two time series enabling online estimation of the state-dependent regression coefficient as well as the interdependent time series, c) an analytical Bayesian inference method called the approximate Gaussian variance inference

(AGVI) which enables performing closed-form online estimation of the univariate process error variance term, d) the extension of the AGVI method to infer multivariate process error variance parameters, and e) applying AGVI to analytically model the heteroscedastic aleatory uncertainty for benchmark regression tasks within the TAGI framework for Bayesian neural networks. The proposed methods are verified with synthetic data and validated with real datasets for structural health monitoring applications as well as benchmark regression datasets. The methods proposed in this thesis are shown to exceed the performance of existing approaches in terms of predictive capacity while being up to orders of magnitude faster.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

In the thesis, I use lower case slanted letters for deterministic variables, upper case slanted letters for random variables, slanted lower case with bold font to denote vectors, and upright upper case with bold font for matrices. The typewriter style is used either for specific names or to represent the number of variables in a set, vector, or matrix.

Symbols

$\mathbf{A}$	Transition matrix
$\mathtt{AR}$	Autoregressive component
$a_i^{(j)}$	i^{th} activation unit in j^{th} hidden layer
$\mathtt{A}$	Number of hidden nodes
$\mathtt{B}$	Batch size
$\mathbf{C}$	Observation matrix
cov	Covariance operator
$\mathtt{D}$	Total number of observations
diag	Create diagonal matrix or get diagonal elements of matrix
$\mathcal{D}$	Set of observations
$\mathcal{D}_{\mathtt{T}}$	Training set
$\mathcal{D}_{\mathtt{V}}$	Validation set
$\mathtt{DKR}$	Double kernel regression
$\mathbb{E}$	Expectation operator
$\mathtt{E}$	Epoch
$\exp(\cdot)$	Exponential operator
$f(\cdot)$	Probability density function
$f(\boldsymbol{x}_t\|\boldsymbol{x}_{t-1})$	Transition model
$f(\boldsymbol{y}_t\|\boldsymbol{x}_t)$	Observation model
$f(\boldsymbol{x}_{t-1}\|\boldsymbol{y}_{1:t-1})$	Prior PDF of hidden states at time t
$f(\boldsymbol{x}_t\|\boldsymbol{y}_{1:t-1})$	Prior predictive PDF of hidden states t
$f(\boldsymbol{y}_t\|\boldsymbol{y}_{1:t-1},\boldsymbol{\theta})$	Marginal prior predictive PDF of the observations $\boldsymbol{y}_t$ given $\boldsymbol{y}_{1:t-1}$
$f(\boldsymbol{y}_t\|\boldsymbol{y}_{1:t-1})$	Evidence
$\hat{f}(\boldsymbol{\theta})$	Target distribution
$\tilde{f}(\boldsymbol{\theta})$	Objective function
$g(\cdot)$	Transition equation

$\mathbf{G}$	Innovation covariance matrix
$h(\cdot)$	Observation equation
$H(\boldsymbol{\theta}, \boldsymbol{r})$	Hamiltonian function
$\mathbf{H}$	Hessian matrix
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{J}$	Kalman smoother gain matrix
$\mathbf{K}$	Kalman gain matrix
κ	Scaling factor for sigma points
$k(\cdot)$	Univariate kernel function
$\boldsymbol{\mathcal{L}}$	Log-likelihood
`LL`	Local level
`LT`	Local trend
`LA`	Local acceleration
ℓ	Kernel length
`L`	Number of hidden layers
$\boldsymbol{L^W}$	Upper triangular random matrix
$\overrightarrow{\boldsymbol{L^W}}$	Random vector of all elements in $\mathbf{L^W}$
$\boldsymbol{M}$	Random mean of a Gaussian PDF
$\mathcal{N}(\cdot)$	Gaussian distribution
$\mathcal{O}$	Computational complexity
`S`	Periodic
$\mathbf{Q}$	Process error covariance matrix
$q(\boldsymbol{\theta'}\|\boldsymbol{\theta})$	Proposal distribution
$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$	Expected complete data log-likelihood at the current parameters $\boldsymbol{\theta}^i$
$\mathbf{R}$	Observation error covariance matrix
$\boldsymbol{r}$	Innovation vector
`T`	Total number of timestamps
t	Timestamp
`TM`	Trend multiplicative
$\boldsymbol{v}$	Vector of observation errors
$\boldsymbol{V}$	Random variable representing observation errors
var	Variance operator
$\widetilde{\overline{V^2}}$	Transformed random variable $\overline{V^2}$ using the exponential function
$\boldsymbol{w}$	Vector of process errors
$\boldsymbol{W}$	Random variable representing process errors
W^2	Gaussian random variable for the square of the process error W

$\overline{W^2}$	Gaussian random variable for the process error's variance term σ_W^2
$W^i W^j$	Product of any i^{th} and j^{th} process errors
$\overline{W^i W^j}$	Random variable for the mean of $W^i W^j$
$\boldsymbol{W^p}$	Random vector of all the product terms $W^i W^j$
$\overline{\boldsymbol{W^p}}$	Random vector for all random variables in $\boldsymbol{\mu^{W^p}}$
x	Covariate or hidden state variable
$\boldsymbol{x}$	Vector of covariates or hidden state variables
$\mathtt{X}$	Size of the hidden state vector
$\boldsymbol{\mathcal{X}}$	Sigma-point set
$X^{\mathtt{p}}$	Product of two Gaussian random variables
$\tilde{\boldsymbol{x}}$	Augmented hidden state vector
$x^{\mathtt{cp}}$	Reference variable for control-points
$x^{\phi^{\mathtt{R}}}$	Hidden state variable representing the regression coefficient for a control-point
$x_0^{\phi^{\mathtt{R}}}$	Hidden state for the state-dependent regression coefficient
$x^{\mathtt{D}}$	Interdependent hidden state
$x^{\mathtt{ref}}$	Independent or reference hidden state
$\boldsymbol{x}^{\mathtt{SK}}$	Hidden state vector associated with the raw kernel outputs
$\tilde{\boldsymbol{x}}^{\mathtt{SK}}$	Hidden state associated with the normalized kernel outputs
y	Observation
$\boldsymbol{y}$	Vector of observations
$\boldsymbol{\mathcal{Y}}$	Transformed observations
$\hat{\boldsymbol{y}}$	Predicted observations
$z_i^{(j)}$	i^{th} hidden unit in j^{th} layer
$\boldsymbol{z}^{(\mathtt{O})}$	Vector of hidden units in the output layer
α	Gain parameter associated with the mean
β	Gain parameter associated with the error's variance
β^H	Acceptance ratio
Δt	Time-step duration
ϵ	Standard Gaussian variable
μ	Expected value
$\boldsymbol{\mu}$	Vector of expected values
$\boldsymbol{\mu}_{t\|t}$	Posterior mean vector for the hidden states $\boldsymbol{x}$ at time t
$\tilde{\boldsymbol{\mu}}$	Mean vector of the augmented state vector $\tilde{\boldsymbol{x}}$
μ^{W^2}	Mean parameter for the random variable W^2
$\mu^{\overline{W^2}}$	Mean parameter for the random variable $\overline{W^2}$
$\boldsymbol{\mu}^{\boldsymbol{W^p}}$	Mean vector of $\boldsymbol{W^p}$

ρ	User-defined forgetting factor
σ	Standard deviation
$\boldsymbol{\Sigma}$	Covariance matrix
$\boldsymbol{\Sigma}_{t\|t}$	Posterior covariance matrix for the hidden states $\boldsymbol{x}$ at time t
$\boldsymbol{\Sigma}_{\boldsymbol{XY},t\|t-1}$	Cross-covariance matrix between the hidden states $\boldsymbol{X}$ and the observations $\boldsymbol{Y}$
$\tilde{\boldsymbol{\Sigma}}$	Covariance matrix of the augmented state vector $\tilde{\boldsymbol{x}}$
σ_W^2	Variance associated with the process error term W
σ_V^2	Variance associated with the observation error term V
$(\sigma^{W^2})^2$	Variance parameter for the random variable W^2
$(\sigma^{\overline{W^2}})^2$	Variance parameter for the random variable $\overline{W^2}$
$\boldsymbol{\Sigma}^{\boldsymbol{XW}}$	Covariance matrix between $\boldsymbol{X}$ and $\boldsymbol{W}$
$\boldsymbol{\theta}$	Model parameters
$\boldsymbol{\theta_0}$	Initial parameters
$\boldsymbol{\theta}^*$	MAP or MLE estimate for parameters
$\check{\boldsymbol{\theta}}$	True parameter values
$\boldsymbol{\theta}^{(\mathtt{L})}_{\overline{V^2}}$	Parameter set connected to the output unit $\overline{V^2}$
$\odot$	Element-wise multiplication
$\prod$	Product operator
∇	Gradient operator
$\frac{\partial f}{\partial x}$	Partial derivative operator
∞	Infinity
$\phi(\cdot)$	Activation function
$\phi^{\mathtt{AR}}$	Autoregressive coefficient
$\boldsymbol{\eta}^{(\mathtt{0})}$	Hyperparameters

Acronyms

AGVI	Approximate Gaussian variance inference
AKF	Adaptive Kalman filter
ALMF	Adaptive limited memory filter
BDLM	Bayesian dynamic linear model
BNN	Bayesian Neural Network
CKF	Cubature Kalman filter
CMM	Covariance-matching method
EKF	Extended Kalman filter
EM	Expectation-maximization

GMA	Gaussian multiplicative approximation
ICM	Innovation correlation method
KF	Kalman filter
KR	Kernel regression
KS	Kalman smoother
LAP	Laplace approximation
MCMC	Markov chain Monte Carlo
MLE	Maximum likelihood estimation
MAP	Maximum a posteriori estimation
NR	Newton-Raphson
NEES	Normalised estimation error squared
NIS	Normalised innovation error squared
PDF	Probability density function
RTS	Rauch-Tung-Striebel
RMSE	Root mean square error
SHM	Structural health monitoring
SSM	State-space model
SWVAKF	Sliding window variational adaptive Kalman filter
TAGI	Tractable approximate Gaussian inference
UKF	Unscented Kalman filter

CHAPTER 1 Introduction

1.1 Motivation

Engineering applications often rely on models to represent physical phenomena. In civil engineering and more specifically in the context of structural health monitoring, data-driven models are developed to assess the deterioration of infrastructures over time such that abnormal structural behavior can be identified [17]. This is carried out by modeling the different structural responses such as displacements, modal frequencies, stresses, and strains, etc. as a proxy for the structure's health [18]. These models involve *parameters* that define the model structure used to represent the phenomenon. In general, the basic setup to learn these model parameters relies on optimization methods that only provide a single optimal value, known as *point estimates*. As a result, we also obtain point estimates for the model predictions. Making informed decisions requires models that are probabilistic in nature, which provide not only point estimates but also uncertainties associated with the model parameters.

Bayesian inference provides the mathematical framework for obtaining the conditional probability of a random variable given observation [19]. For continuous model parameters, such a framework is used for evaluating their posterior probability density function (PDF) which allows us to quantify two forms of uncertainty, i.e., *epistemic* and *aleatory* [19, 20]. The uncertainty associated with the model parameters that is reducible as more information is collected is referred to as epistemic uncertainty. For an *identifiable problem* [21], where an infinite amount of independent training data is available, the parameters' posterior PDF approaches a Dirac delta distribution such that the function is zero everywhere except at the true value. In practical cases where only a limited amount of data is available, epistemic uncertainty remain over the parameters.

Moreover, in complex practical applications, a discrepancy always remains between the model predictions and the reality. This may arise from misrepresentation of the physical phenomenon as a result of a lack of understanding or availability of explanatory variables that can fully describe a phenomenon. This form of uncertainty is referred to as aleatory; in the context of a given model structure, it cannot be reduced regardless of the amount of training data available as the model itself is inadequate [22]. Furthermore, the aleatory uncertainty may vary as a function of the input values for which missing explanatory variables is a primary cause. Hence, it is important to quantify both epistemic and aleatory uncertainties in order to understand whether the model's uncertainty is due to a lack of data or model inadequacy.

State-space models and *Bayesian neural networks* are key probabilistic models used for performing time series forecasting and regression tasks, respectively. State-space models describe dynamic systems through unobserved state variables referred to as *hidden states* that are inferred from imperfect observations [20]. Bayesian neural networks provide a probabilistic approach to the traditional neural networks with regard to quantifying uncertainty over its parameters, i.e., the weights and biases using the Bayesian inference framework. Both these approaches involve unknown parameters for not only modeling the phenomenon but also for quantifying the model's epistemic and aleatory uncertainties.

For instance, in state-space models, it is computationally cheap to estimate the expected values and the covariance matrix that are quantifying the mean and the epistemic uncertainties for the hidden state variables because we can rely on an analytical formulation for performing Bayesian inference. In contrast, obtaining optimal estimates for the variance parameters in the process ($\mathbf{Q}$) and observation ($\mathbf{R}$) error covariance matrices that quantifies the model's aleatory uncertainties is typically the most computationally demanding task in the state estimation procedure [20]. Even though in many situations the matrix $\mathbf{R}$ can be considered to be known from the measuring instrument specifications, it remains a challenge to develop a computationally efficient method which is able to perform closed-form online estimation of the matrix $\mathbf{Q}$ for multiple time series. In Bayesian neural networks, analytical parameter inference can be carried out using the *tractable approximate Gaussian inference* (TAGI), but it is restricted to modeling *homoscedastic* aleatory uncertainty quantified by a constant observation error variance across the input covariate-domain. In practical regression tasks, it is necessary to quantify the *heteroscedastic* aleatory uncertainty as a function of the input covariates.

Furthermore, additional parameters that exist in the transition and observation equations of state-space models can be inferred as hidden states using a multiplicative structure. However, such an analytical formulation does not exist for multiplicative state-space models that would allow closed-form inference of the model parameters as hidden states. A closed-form inference is only possible with Bayesian dynamic linear models (BDLM) which are a type of state-space models having linear transition and observation equations, and a Gaussian assumption for the hidden states' PDF. The *Kalman filter* is employed to obtain the closed-form posterior moments for the hidden states. For multiplicative state-space models, we have to rely on nonlinear filtering methods such as the cubature Kalman filter and particle filters [20, 23] that use sampling-based approaches which are computationally expensive compared to the Kalman filter. Such a closed-form analytically tractable formulation for the multiplicative state-space model will not only allow model parameters to be treated as hidden states but will also present the possibility of creating new generic components involving the product of

any two hidden states representing specific patterns in the data.

Additionally, the existing BDLM framework only allows for modeling linear relationships between the independent and the dependent time series described by a constant regression coefficient [21]. However, it is not possible to model a nonlinear relationship between two time series in such a way that the regression coefficient is not a constant parameter, but depends on the current value of the independent time series. To achieve this, we need a framework that allow performing closed-form inference for the product of a regression coefficient with the independent hidden state variable to obtain the interdependent state variable. Hence, a key limitation to be addressed is to develop the mathematical formulation for obtaining closed-form moments for the product of any two hidden states.

In practice, estimating the parameters defining the dynamic system in state-space models and the variance parameters quantifying the aleatory uncertainties in both state-space models and Bayesian neural networks can be order of magnitude more computationally demanding than estimating the expected values and the parameters quantifying epistemic uncertainties. The challenge associated with estimating these parameters prevents existing models from being scaled up to be used in large-scale practical engineering applications.

1.2 Research Objectives

This thesis aims at developing analytically tractable Bayesian methods for parameter inference in order to improve the performance and scalability of probabilistic models in the context of engineering applications. The core objectives of this thesis are:

- Develop an analytical method to handle multiplicative state-space models.
- Formulate an analytical Bayesian inference method that will enable performing closed-form online estimation of the process error covariance matrix $\mathbf{Q}$ for multiple time series.
- Derive a framework to analytically model the heteroscedastic aleatory uncertainty in Bayesian neural networks.

1.3 Thesis Outline

The content of this thesis is organized as follows: Chapter 2 presents a literature review on parameter inference methods for two key probabilistic models: the state-space models and Bayesian neural networks. Chapter 3 introduces an analytical framework for handling

multiplicative state-space models by taking advantage of the Gaussian multiplicative approximation (GMA) that explicitly provides closed-form equations for moment computation for the product of two hidden states. Chapter 4 describes a state-based regression method that builds upon GMA to model the nonlinear dependency between two time series enabling online estimation of the state-dependent regression coefficient as well as the interdependent time series. Chapter 5 presents an analytical Bayesian inference method called the approximate Gaussian variance inference (AGVI) which enables performing closed-form online estimation of the univariate process error variance term in the context of state-space models applied to time series. Chapter 6 presents the extension of the AGVI method to infer multivariate process error variance parameters in the full $\mathbf{Q}$ matrix for multiple time series and ensures that the $\mathbf{Q}$ matrix is positive semi-definite at any instant of time. Chapter 7 provides a framework to analytically model the heteroscedastic aleatory uncertainty in Bayesian neural networks for regression tasks. Finally, Chapter 8 provides the thesis conclusions, its limitations, and future research directions.

1.4 Co-Authored Papers

The list of co-authored papers that are part of this thesis is:

- Deka, B., Ha Nguyen, L., Amiri, S., & Goulet, J. A. (2022). The Gaussian multiplicative approximation for state-space models. Structural Control and Health Monitoring, 29(3), e2904.
- Deka, B., Vuong, V. D., Goulet, J. A., Côté, P. & Miquel, B. (Submitted, 2022). Dam Behavior Prediction Using an Ensemble of Bayesian Dynamic Linear Model and Bayesian LSTM Networks. 16^{th} *International Benchmark Workshop on Numerical Analysis of Dams*, International Commission on Large Dams, Ljubljana, Slovenia.

CHAPTER 2 Literature Review

2.1 Introduction

In engineering, we use parametric models to represent physical phenomena. In the most basic setup, these model parameters are estimated using optimization methods that provide a single optimal value i.e., a *point estimate* for each parameter. However, in order to express the uncertainty associated with our predictions, we need methods that can quantify the uncertainties associated with these model parameters.

Bayesian inference provides the mathematical framework for quantifying the posterior distribution of the parameters given the data $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{\mathtt{D}}$, where $\mathtt{D}$ refers to the total number of data points, and $\boldsymbol{x}_i \in \mathbb{R}^{\mathtt{X}}$ are the input features, and $\boldsymbol{y}_i \in \mathbb{R}^{\mathtt{Y}}$ are the observations corresponding to the i^{th} data point. With *Bayes rule*, we can obtain the conditional probability density $f(\cdot|\cdot)$ for any two continuous random variables x and y such that

$$f(x|y) = \frac{f(y|x) \cdot f(x)}{f(y)}.$$

For continuous model parameters ($\boldsymbol{\theta}$), we can use the Bayes rule to infer their posterior probability density function (PDF) such that

$$f(\boldsymbol{\theta}|\mathcal{D}) = \frac{f(\mathcal{D}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})}{f(\mathcal{D})},$$

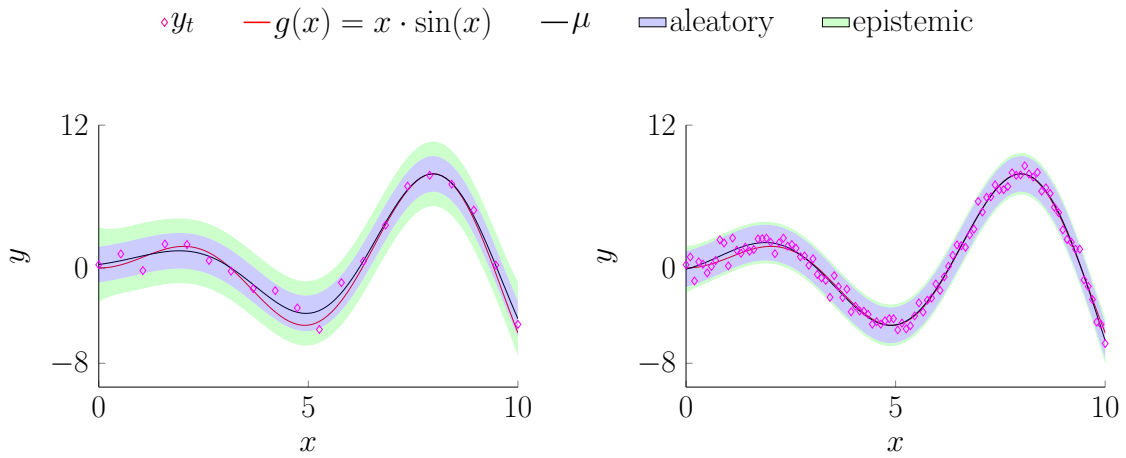
where the numerator $f(\mathcal{D}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})$ denotes the joint PDF obtained from the product of the likelihood of data given the parameters $f(\mathcal{D}|\boldsymbol{\theta})$, and the prior PDF of the model parameters $f(\boldsymbol{\theta})$ and the denominator $f(\mathcal{D}) = \int f(\mathcal{D}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta}) d\boldsymbol{\theta}$ denotes the marginal likelihood, which is also referred to as the *evidence*. Using the posterior PDF $f(\boldsymbol{\theta}|\mathcal{D})$, we can compute the posterior predictive PDF $f(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ for the output random variables $\boldsymbol{Y}$ given the input values $\boldsymbol{x}$ by marginalizing the joint PDF $f(\boldsymbol{y}, \boldsymbol{\theta}|\boldsymbol{x}, \mathcal{D})$ over all possible parameter values so that

$$f(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}) = \int f(\boldsymbol{y}, \boldsymbol{\theta}|\boldsymbol{x}, \mathcal{D}) d\boldsymbol{\theta} \ = \int f(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) \cdot f(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta},$$

where $f(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$ is the model that generates the outputs $\boldsymbol{y}$ given the inputs $\boldsymbol{x}$ and the parameters $\boldsymbol{\theta}$.

For an *identifiable problem* [21], where we have access to an infinite amount, i.e., $\mathtt{D} \to \infty$, of independent data, the parameters' posterior PDF approaches a Dirac delta function shown

by $f(\boldsymbol{\theta}|\mathcal{D}) \approx \delta(\boldsymbol{\theta} - \check{\boldsymbol{\theta}})$ such that the function is zero everywhere except at the true value $\check{\boldsymbol{\theta}}$. This is because we have reached a point where we have an exact knowledge of our model parameters and as a result, we would also obtain point estimates for the model outputs. This type of uncertainty that is reducible by collecting more data is known as *epistemic*. In complex practical applications, regardless of how much data we collect, a discrepancy typically remains between the predictions for our model and the real phenomenon that we are trying to model. This often happens because we do not have a complete understanding of all the possible explanatory variables that describe a physical phenomenon. The type of uncertainty that arises because of model inadequacy is typically referred to as *aleatory*. Figure 2.1 shows the distinction between the epistemic and the aleatory uncertainty associated with the model predictions represented by the green and the blue regions. As the number of observation increases from $\mathtt{D} = 20$ to 100, we observe that the epistemic uncertainty in green shrinks to a negligible value while the aleatory one remains constant. An aleatory uncertainty that is assumed to be constant over the domain of the explanatory variables is known as *homoscedastic* [24].

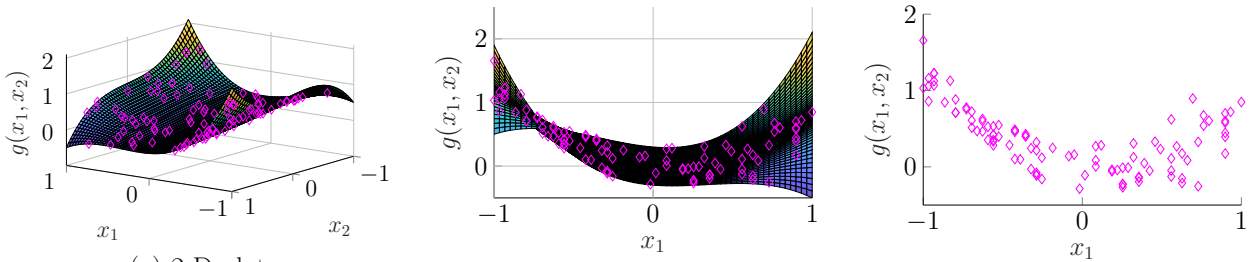


(a) $\mathtt{D} = 20$ (b) $\mathtt{D} = 100$

Figure 2.1 Visual interpretation showing the epistemic and the aleatory uncertainty associated with the model predictions represented by the green and blue regions respectively. The model predictions are shown in black where the true function $g(x) = x \cdot \sin(x)$ is shown in red. As the number of observation increases from a) $\mathtt{D} = 20$ to b) $\mathtt{D} = 100$, we observe that the epistemic uncertainty shrinks to a negligible value while the aleatory remains constant.

In most methods, the aleatory uncertainty is modeled as being homoscedastic. However, in real-world problems this assumption does not hold and we often observe that the aleatory uncertainty varies as a function of the input values. For example, Figure 2.2a shows a two-dimensional plot where we consider two input variables x_1 and x_2 to generate data using a

two-dimensional deterministic function given by $g(x_1, x_2)$. If we consider that one of the input variable is unavailable, let's say x_2, and rotate the plot to show the data only with respect to x_1 as shown by Figure 2.2b, we observe that the uncertainty associated with the data varies as we move from -1 to 1 within the domain of x_1 as shown by Figure 2.2c. This type of aleatory uncertainty that varies with the input values is known as *heteroscedastic*, for which a primary cause in practical applications is the lack of access to some of the explanatory variables.



(a) 2-D plot (b) Rotated 2-D plot (c) 1-D representation

Figure 2.2 Heteroscedastic aleatory uncertainty as a result of missing explanatory variables. Figure a) shows the 2-D plot generated using the function $g(x_1, x_2)$, b) shows the rotated view of the 2-D plot in (a) to show the surface plot with respect to the x_1 axis, and c) shows the varying uncertainty of the data as we move from -1 to 1 within the domain of x_1.

For decision making, it is important to know whether the uncertainty is due to a limited amount of data or because of model inadequacy. Therefore, we need probabilistic models capable of producing reliable predictions and robust estimates for both the epistemic and aleatory uncertainties. The subsequent sections will outline two key probabilistic models, namely the *state-space models* and the *Bayesian neural networks* used for forecasting time series and regression tasks in engineering problems.

2.2 State-Space Models

State-space models (SSM) [20, 21, 25] are probabilistic models used for modeling dynamic systems that are indirectly observed through imperfect data [20]. Applications of such models can be found in fields such as navigation, aerospace, telecommunication and control-engineering. In civil engineering, SSM find their applications in the monitoring of structural degradation over time [21, 26, 27]. For instance, by observing a structure's condition x_t, we can obtain two new states: the degradation speed $\dot{x}_t$ and the acceleration $\ddot{x}_t$, that are indirectly observed [27].

To formulate a SSM, the first step is to define the dynamic model's structure through unobserved *hidden states* $\boldsymbol{x}_t = [x_1, x_2, \cdots, x_{\mathtt{X}}]_t^\intercal, \in \mathbb{R}^{\mathtt{X}}$, and the second step is to define the relationship between these hidden states $\boldsymbol{x}_t$ and the observations $\boldsymbol{y}_t = [y_1, y_2, \cdots, y_{\mathtt{Y}}]_t^\intercal, \in \mathbb{R}^{\mathtt{Y}}$ at a time step t. The dynamic model is defined using the *Markov hypothesis* according to which the hidden states $\boldsymbol{x}_t$ are only dependent on $\boldsymbol{x}_{t-1}$ considering that all the information from the past time steps $\{0,\, 1,\, \ldots t-1\}$ are contained in $\boldsymbol{x}_{t-1}$. Using the Markov hypothesis, we can define the *transition model* $f(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ that relates the hidden states $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ such that $f(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}) = f(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, where we use a short-hand notation $1:t-1$ for $\{1, 2, \cdots, t-1\}$. The relationship between the hidden states $\boldsymbol{x}_t$ and the observations $\boldsymbol{y}_t$ are modeled using an *observation model* given by $f(\boldsymbol{y}_t|\boldsymbol{x}_t)$. The generic form of a SSM can be summarized as follows

$$\begin{aligned} \boldsymbol{x}_t &= g(\boldsymbol{x}_{t-1}, \boldsymbol{w}_t), \\ \boldsymbol{y}_t &= h(\boldsymbol{x}_t, \boldsymbol{v}_t), \end{aligned} \tag{2.1}$$

where $\boldsymbol{x}_t$ is the hidden state vector, $\boldsymbol{y}_t$ is the observation vector, $g(\cdot)$ and $h(\cdot)$ are the linear/nonlinear functions for the transition and the observation equation, $\boldsymbol{w}_t$ is the process error, and $\boldsymbol{v}_t$ is the observation error. The main purpose of using SSMs is to perform hidden state estimation where using the prior knowledge of the hidden states $f(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$, the transition model $f(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, and the observation model $f(\boldsymbol{y}_t|\boldsymbol{x}_t)$, and the observation y_t, we can employ Bayesian inference to obtain the posterior PDF $f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ such that

$$f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{f(\boldsymbol{y}_t|\boldsymbol{x}_t) \cdot f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{f(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})}, \tag{2.2}$$

where the prior predictive PDF $f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int f(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \cdot f(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}) d\boldsymbol{x}_{t-1}$ and the evidence $f(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}) = \int f(\boldsymbol{y}_t|\boldsymbol{x}_t) \cdot f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) d\boldsymbol{x}_t$. SSMs can be linear or nonlinear based on the functions g and h describing the transition and observation equations. For nonlinear models, the inference step described by the Equation 2.2 might be either intractable or computationally expensive [19,20]. The following section will describe the *Bayesian dynamic linear models* (BDLM) which are a special type of SSM employing linear transition and observation equations.

2.2.1 Bayesian Dynamic Linear Models

BDLM employs linear dynamic models together with additive errors such that the transition and the observation equations are

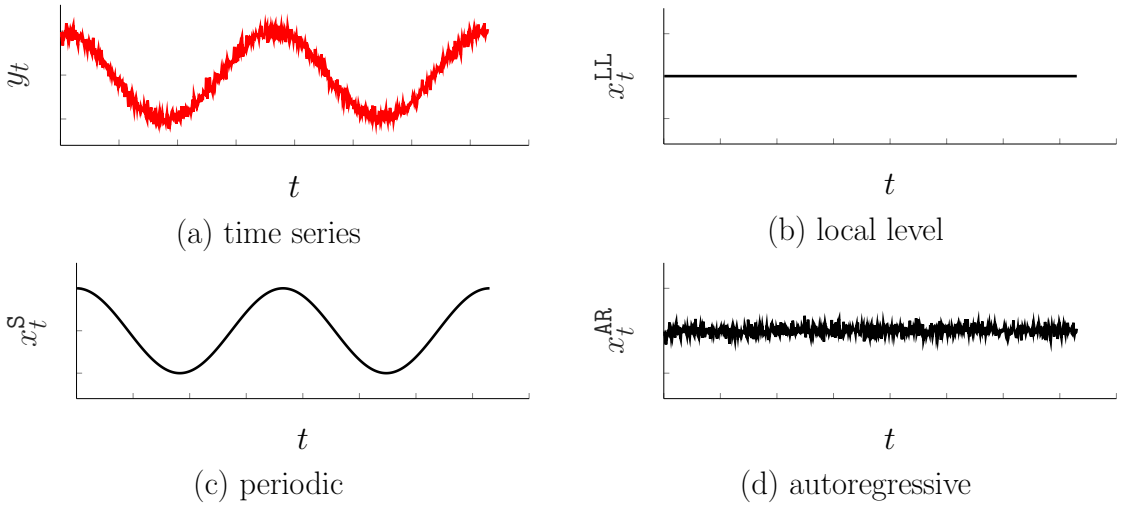
$$\begin{aligned} \boldsymbol{x}_t &= \mathbf{A}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t, \quad \boldsymbol{w} : \boldsymbol{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \\ \boldsymbol{y}_t &= \mathbf{C}\boldsymbol{x}_t + \boldsymbol{v}_t, \quad \boldsymbol{v} : \boldsymbol{V} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \end{aligned}$$

where $\mathbf{A}$ is the transition matrix, $\mathbf{C}$ is the observation matrix, $\mathbf{Q}$ is the process error covariance matrix, and $\mathbf{R}$ is the observation error covariance matrix. BDLM may comprise one or more *generic components* where each models a specific pattern having its own linear dynamic structure. There are five main components namely the *local level* (`LL`), the *local trend* (`LT`), the *local acceleration* (`LA`), the *periodic* (`S`), and the *autoregressive* (`AR`). In the context of structural health monitoring, the local level, the local trend, and the local acceleration components are used to model the baseline of any time series without the external effects such as temperature or water level. These components capture the irreversible pattern which can be used for anomaly detection [28]. The periodic or the kernel regression (`KR`) components are used to identify any external effects having a periodic pattern that can be harmonic or non-harmonic in nature [29]. The autoregressive component is used to capture the residuals term that is not captured by the other components. Several components can be assembled to model a wide range of patterns. Figure 2.3 shows such an example where the time series y_t is decomposed into three specific patterns using a local level, a periodic, and an autoregressive component. For a full description of the mathematical formulation and application of the generic components, the reader can consult the work of Goulet [18] and West [30].

For linear-Gaussian systems, exact inference can be carried out for the posterior PDF of the hidden states using a recursive process called *filtering*. The Bayesian filtering techniques with Gaussian assumption for the hidden states' PDF comes under a special class known as the *Gaussian filters* [20, 31, 32].

2.2.2 Gaussian Filters

In the Gaussian filters, the prior PDF $f(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$ of the hidden states and the likelihood $f(\boldsymbol{y}_t|\boldsymbol{x}_t)$ are assumed to be Gaussian which makes the posterior $f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ of the hidden states also Gaussian [32]. When the dynamic model is linear, the process as well as the observation errors are *white*, i.e., uncorrelated random variables having a zero mean and a finite variance, and the errors are *additive*, i.e., they are added to the state or the observation variable, then the Gaussian filter is equivalent to the filtering technique known as the *Kalman filter* [33].



(a) time series (b) local level

(c) periodic (d) autoregressive

Figure 2.3 An example showing the generic components in BDLM for time series modeling. The Figure a) shows the time series y_t in red, (b) the local level, (c) the periodic, and (d) the autoregressive component in black.

Kalman Filter

The Kalman filter is a recursive filtering method that provides closed-form solutions for the first two moments of the hidden states in a linear-Gaussian dynamic system. Hence, the Kalman Filter reduces to calculating the mean vector and the covariance matrices recursively to get the exact Gaussian posterior PDF for the hidden states. The posterior mean vector and the covariance matrix at time t are obtained using a two-step procedure; the *prediction step* and the *update step*. In the prediction step, we compute the moments for the prior predictive PDF of $\boldsymbol{X}_t|\boldsymbol{y}_{1:t-1} \equiv \boldsymbol{X}_{t|t-1} \sim \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$ given by

$$\begin{aligned} \mathbb{E}[\boldsymbol{X}_{t|t-1}] &\equiv \boldsymbol{\mu}_{t|t-1} &&= \mathbf{A}\boldsymbol{\mu}_{t-1|t-1}, \\ \mathrm{cov}(\boldsymbol{X}_{t|t-1}) &\equiv \boldsymbol{\Sigma}_{t|t-1} &&= \mathbf{A}\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{A}^\intercal + \mathbf{Q}, \end{aligned}$$

where $\boldsymbol{\mu}_{t-1|t-1}$ and $\boldsymbol{\Sigma}_{t-1|t-1}$ are the prior moments of $\boldsymbol{X}_{t-1}|\boldsymbol{y}_{1:t-1} \equiv \boldsymbol{X}_{t-1|t-1}$ obtained using all the observations $\boldsymbol{y}_{1:t-1}$ up to the time step $t-1$. Given that the observations at time t

are available, we can obtain the posterior moments of $\boldsymbol{X}_{t|t}$ using the update step shown by

$$\begin{aligned} f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) &= \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}), \\ \boldsymbol{\mu}_{t|t} &= \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t \boldsymbol{r}_t, \\ \boldsymbol{\Sigma}_{t|t} &= (\mathbf{I} - \mathbf{K}_t \mathbf{C}) \boldsymbol{\Sigma}_{t|t-1}, \\ \boldsymbol{r}_t &= \boldsymbol{y}_t - \hat{\boldsymbol{y}}_t, \\ \hat{\boldsymbol{y}}_t &= \mathbf{C} \boldsymbol{\mu}_{t|t-1}, \\ \mathbf{K}_t &= \boldsymbol{\Sigma}_{t|t-1} \mathbf{C}^\intercal \mathbf{G}_t^{-1}, \\ \mathbf{G}_t &= \mathbf{C} \boldsymbol{\Sigma}_{t|t-1} \mathbf{C}^\intercal + \mathbf{R}, \end{aligned}$$

where $\boldsymbol{\mu}_{t|t}$ and $\boldsymbol{\Sigma}_{t|t}$ are the posterior mean vector and covariance matrix for $\boldsymbol{X}_{t|t}$, $\boldsymbol{r}_t$ is the innovation vector, $\mathbf{I}$ is the identity matrix, $\mathbf{K}_t$ is the Kalman gain matrix, and $\mathbf{G}_t$ is the innovation covariance matrix. The Kalman filter method is carried out recursively from time step $t-1$ to t as new observations are collected and can be summarized as follows

$$(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}, \boldsymbol{\mathcal{L}}_t) = \text{Kalman filter}(\boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1}, \boldsymbol{y}_t, \mathbf{A}, \mathbf{Q}, \mathbf{C}, \mathbf{R}), \tag{2.3}$$

where $\boldsymbol{\mathcal{L}}_t$ is the log-likelihood obtained for the observations $\boldsymbol{y}_t$ to be used for the parameter estimation that will be discussed in Section 2.3. Hence, using the prior knowledge of the hidden states at $t-1$, the $\mathbf{A}, \mathbf{Q}, \mathbf{C}$, and $\mathbf{R}$ matrices that define the linear dynamic system, and the observations $\boldsymbol{y}_{1:t}$, we obtain the posterior knowledge for the hidden states at time t.

The Kalman filter is an exact state estimator for linear dynamic systems. However, in many real applications, the linear assumption may not hold. In the case of nonlinear dynamic systems, closed-form solutions are not available and require approximations for the posterior PDF leading to sub-optimal solutions [31]. There are two main approaches in the literature for performing this approximation:

a. Local approach: In this approach, the posterior PDF is assumed to have a known type and is computed using numerical approximations [32]. This approximation can be performed either through linearisation of the nonlinear function, e.g., the *extended Kalman filter* (EKF) [34] or by approximating the mean vector and the covariance matrix of the posterior PDF directly using weighted samples, e.g., the *unscented Kalman filter* (UKF) [35] and the *cubature Kalman filter* (CKF) [32].

b. Global approach: In this approach, there is no assumption made for the posterior PDF's type. The *particle filter* is one such example where the posterior PDF can have any type and is approximated using *Monte Carlo* sampling methods [36]. This approach has the limitation of having a high computational cost associated with the sampling and is

inefficient for online state estimation in comparison with the exact Kalman filter [31].

The methods belonging to the local approach are discussed further owing to their advantages of being accurate, analytically tractable as well as having a low computational cost as compared to the global approaches. The aforementioned techniques approximate the following moment integral for the expectation of $\boldsymbol{X}$,

$$I = \int_{-\infty}^{\infty} g(\boldsymbol{x}) \cdot f(\boldsymbol{x}) d\boldsymbol{x},$$

where $g(\boldsymbol{x})$ is the nonlinear dynamic model, and $f(\boldsymbol{x})$ is a known PDF for the states which can also be Gaussian. The numerical approximation reduces the integral to the form,

$$I \approx \sum_{i=1}^{\mathtt{X}} w_i \cdot g(x_i),$$

where x_i are the samples, $\mathtt{X}$ is the number of samples, and w_i are the associated weights. In the subsequent sections, two of the main Gaussian filters used for nonlinear functions, namely the UKF and CKF, are reviewed.

Unscented Kalman Filter

The unscented Kalman filter (UKF) is a nonlinear filtering method that approximates the posterior PDF using a set of samples called the sigma points and an unscented transform (UT) [37] method for the associated weights [35]. The sigma points are generated symmetrically around the prior mean vector which has a considerably higher weight than the other points. These sigma points are propagated through the nonlinear model to estimate the posterior PDF's mean vector and covariance matrix which is accurate up to third order for the mean and up to first order for the covariance associated with any polynomial function [20,35]. The UKF is derivative-free and also a more accurate nonlinear filter than the EKF as it is not based on a linear approximation at a single point but uses a set of points to approximate the nonlinear function [20, 31].

The formulation of the UKF is described by considering a state vector $\boldsymbol{x}$ of size $\mathtt{X}$ that is transformed by a nonlinear function $g(\cdot)$ such that $f(g(\boldsymbol{x})) = \mathcal{N}(g(\boldsymbol{x}); \boldsymbol{\mu}, \boldsymbol{\Sigma})$ follows a Gaussian PDF having the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. Note that the UKF is not only restricted to the Gaussian case as it is applicable for any symmetric PDF. A sigma matrix of $2\mathtt{X} + 1$ sigma point vectors, $\boldsymbol{\mathcal{X}} = [\boldsymbol{\mathcal{X}}_0 \ \boldsymbol{\mathcal{X}}_i \ \boldsymbol{\mathcal{X}}_{\mathtt{X}+i}]^\intercal$, and their weights, $\boldsymbol{w} = [w_0 \ w_i \ w_{\mathtt{X}+i}]^\intercal$, are chosen such that the first two moments associated with the PDF

$f(g(\boldsymbol{x}))$ are matched accurately as shown by

$$\begin{aligned} \boldsymbol{\mu} &= \sum_{i=0}^{2\mathtt{X}} w_i \mathcal{X}_i, \\ \boldsymbol{\Sigma} &= \sum_{i=0}^{2\mathtt{X}} w_i (\mathcal{X}_i - \boldsymbol{\mu})(\mathcal{X}_i - \boldsymbol{\mu})^\intercal. \end{aligned} \tag{2.4}$$

The sigma vectors forming the sigma matrix $\boldsymbol{\mathcal{X}}$ and their corresponding weights $\boldsymbol{w}$ are described by

$$\begin{aligned} \boldsymbol{\mathcal{X}}_0 &= \boldsymbol{\mu}, & w_0 &= \tfrac{\kappa}{\mathtt{X}+\kappa}, \\ \boldsymbol{\mathcal{X}}_i &= \boldsymbol{\mu} + (\sqrt{(\mathtt{X}+\kappa)\boldsymbol{\Sigma}})_i, & w_i &= \tfrac{1}{2(\mathtt{X}+\kappa)}, \\ \boldsymbol{\mathcal{X}}_{\mathtt{X}+i} &= \boldsymbol{\mu} - (\sqrt{(\mathtt{X}+\kappa)\boldsymbol{\Sigma}})_i, & w_{\mathtt{X}+i} &= \tfrac{1}{2(\mathtt{X}+\kappa)}, \end{aligned}$$

where $i = 1, 2, \ldots, \mathtt{X}$, $(\sqrt{(\mathtt{X}+\kappa)\boldsymbol{\Sigma}})_i$ is the i^{th} column of the matrix $\sqrt{(\mathtt{X}+\kappa)\boldsymbol{\Sigma}}$, and the parameter $\kappa = (3 - \mathtt{X})$ is the scaling factor that controls the distribution of the sigma points around the prior mean vector $\boldsymbol{\mu}$. Using the sigma point sets and the weights, the UKF approximates the posterior PDF $f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t})$ where the mean vector and the covariance matrix are obtained using a two-step procedure as follows: In the prediction step, the sigma point set $\boldsymbol{\mathcal{X}}_{t-1|t-1}$ is propagated through the nonlinear transition function $g(\cdot)$ to obtain the transformed state vector $\boldsymbol{\mathcal{X}}_{t|t-1}$. Using this transformed vector and the associated weights, we obtain the predictive moments $\boldsymbol{\mu}_{t|t-1}$ and $\boldsymbol{\Sigma}_{t|t-1}$. Similarly, the transformed state vector $\boldsymbol{\mathcal{X}}_{t|t-1}$ is propagated through the nonlinear observation function $h(\cdot)$ to obtain the predicted observations $\hat{\boldsymbol{y}}_t$. In the update step, we first compute the innovation covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{Y},t|t-1}$ and the cross-covariance $\boldsymbol{\Sigma}_{\boldsymbol{XY},t|t-1}$ using the transformed state vector $\boldsymbol{\mathcal{X}}_{t|t-1}$ and the predicted observations $\hat{\boldsymbol{y}}_t$ using which we obtain the posterior moments for $f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$. Both steps for the UKF algorithm are summarized as follows:

Prediction step

$\boldsymbol{\mathcal{X}}_{t-1\|t-1}$	$=$	$[\boldsymbol{\mu} \quad \boldsymbol{\mathcal{X}}_i \quad \boldsymbol{\mathcal{X}}_{n+i}]_{t-1\|t-1}$,	Sigma-point set
$\boldsymbol{\mathcal{X}}_{t\|t-1}$	$=$	$g(\boldsymbol{\mathcal{X}}_{t-1\|t-1})$,	Transformed state vector
$\boldsymbol{\mu}_{t\|t-1}$	$=$	$\sum_{i=0}^{2\mathtt{X}} w_i \boldsymbol{\mathcal{X}}_{i,t\|t-1}$,	Prior mean vector
$\boldsymbol{\Sigma}_{t\|t-1}$	$=$	$\sum_{i=0}^{2\mathtt{X}} w_i [\boldsymbol{\mathcal{X}}_{i,t\|t-1} - \boldsymbol{\mu}_{t\|t-1}][\boldsymbol{\mathcal{X}}_{i,t\|t-1} - \boldsymbol{\mu}_{t\|t-1}]^\intercal + \mathbf{Q}$,	Prior covariance matrix
$\boldsymbol{\mathcal{Y}}_{t\|t-1}$	$=$	$h(\boldsymbol{\mathcal{X}}_{t\|t-1})$,	Transformed observations
$\hat{\boldsymbol{y}}_t$	$=$	$\sum_{i=0}^{2\mathtt{X}} w_i \mathcal{Y}_{i,t\|t-1}$,	Predicted observations

Update step

$$
\begin{array}{lcll}
\boldsymbol{\Sigma}_{\boldsymbol{Y},t|t-1} & = & \sum_{i=0}^{2\mathtt{X}} w_i[\mathcal{Y}_{i,t|t-1} - \hat{\boldsymbol{y}}_t][\mathcal{Y}_{i,t|t-1} - \hat{\boldsymbol{y}}_t]^\intercal + \mathbf{R}, & \text{Innovation covariance} \\
\boldsymbol{\Sigma}_{\boldsymbol{XY},t|t-1} & = & \sum_{i=0}^{2\mathtt{X}} w_i[\mathcal{X}_{i,t|t-1} - \mu_{t|t-1}][\mathcal{Y}_{i,t|t-1} - \hat{\boldsymbol{y}}_t]^\intercal, & \text{Cross-covariance} \\
\mathbf{K}_t & = & \boldsymbol{\Sigma}_{\boldsymbol{XY},t|t-1}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{Y},t|t-1}, & \text{Kalman gain} \\
\boldsymbol{\mu}_{t|t} & = & \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t), & \text{Posterior mean} \\
\boldsymbol{\Sigma}_{t|t} & = & \boldsymbol{\Sigma}_{t|t-1} - \mathbf{K}_t\boldsymbol{\Sigma}_{\boldsymbol{Y},t|t-1}\mathbf{K}_t^\intercal, & \text{Posterior covariance}
\end{array}
$$

where $\boldsymbol{\Sigma}_{\boldsymbol{Y},t|t-1} \equiv \text{cov}(\boldsymbol{Y}_t|y_{1:t-1})$ and $\boldsymbol{\Sigma}_{\boldsymbol{XY},t|t-1} \equiv \text{cov}(\boldsymbol{X}_t, \boldsymbol{Y}_t|y_{1:t-1})$.

Even though the UKF is better than the EKF considering both accuracy and efficiency, it can suffer from instabilities and numerical inaccuracies [31, 32]. The covariance matrix may result in non-positive semi-definite (non-PSD) cases owing to round-off errors introduced by sensitive numerical operations such as matrix square rooting, matrix inversion and covariance update through matrix substraction. Hence, the square-root version of the UKF is often necessary to prevent numerical ill-conditioning due to arithmetic imprecision even though the computational complexity increases [32]. However, a stable version of the square-root UKF is still not guaranteed owing to the presence of negatively weighted samples to update the posterior covariance matrix which can still result in non-PSD matrices. The *cubature Kalman filter* is deemed to be more accurate and stable compared to the UKF which is reviewed in the next section.

Cubature Kalman Filter

The cubature Kalman filter (CKF) is another nonlinear filtering method that approximates the multivariate moment integral for $f(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ using a third-order *spherical cubature rule* [32] shown by

$$\int g(\boldsymbol{x}) \cdot f(\boldsymbol{x}) d\boldsymbol{x} = \frac{1}{2\mathtt{X}} \sum_{i=1}^{2\mathtt{X}} g(\boldsymbol{\mu} + \sqrt{\boldsymbol{\Sigma}}\, \boldsymbol{\xi}_i),$$

where $2\mathtt{X}$ cubature points are generated by

$$\boldsymbol{\xi}_i = \begin{cases} \sqrt{\mathtt{X}}\boldsymbol{e}_i, & i = 1, 2, \cdots, \mathtt{X}, \\ -\sqrt{\mathtt{X}}\boldsymbol{e}_{i-\mathtt{X}}, & i = \mathtt{X}+1, \mathtt{X}+2, \cdots, 2\mathtt{X}, \end{cases}$$

where $\mathtt{X}$ is the size of the hidden state vector and $\boldsymbol{e}_i$ is the i^{th} column of the identity matrix $\boldsymbol{I}_{\mathtt{X}}$. Each cubature point is uniformly weighted by $w_i = \dfrac{1}{2\mathtt{X}}$. Also, the weights and the samples of the cubature point set $\{w_i, \boldsymbol{\xi}_i\}$ are only dependent on the size of the state vector and are

independent of the nonlinear function $g(\boldsymbol{x})$. Assuming a Gaussian PDF for the state vector $\boldsymbol{x}$ at time $t-1$ such that $f(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1})$, the prediction and the update steps in the CKF algorithm are given as follows

Prediction step

$$
\begin{aligned}
\boldsymbol{\Sigma}_{t-1|t-1} &= \boldsymbol{S}_{t-1|t-1}\boldsymbol{S}_{t-1|t-1}^{\intercal}, && \text{Factorize}\\
\boldsymbol{\mathcal{X}}_{i,t|t-1} &= \boldsymbol{\mu}_{t-1|t-1} + \boldsymbol{S}_{t-1|t-1}\boldsymbol{\xi}_i, && \text{Cubature-point set}\\
\boldsymbol{\mathcal{X}}_{t|t-1} &= g(\boldsymbol{\mathcal{X}}_{t-1|t-1}), && \text{Transformed state vector}\\
\boldsymbol{\mu}_{t|t-1} &= \tfrac{1}{2\mathtt{X}}\textstyle\sum_{i=1}^{2\mathtt{X}} \boldsymbol{\mathcal{X}}_{t|t-1}, && \text{Prior mean}\\
\boldsymbol{\Sigma}_{t|t-1} &= \tfrac{1}{2\mathtt{X}}\textstyle\sum_{i=1}^{2\mathtt{X}} \boldsymbol{\mathcal{X}}_{i,t|t-1}\boldsymbol{\mathcal{X}}_{i,t|t-1}^{\intercal} - \boldsymbol{\mu}_{t|t-1}\boldsymbol{\mu}_{t|t-1}^{\intercal} + \mathbf{Q}, && \text{Prior covariance}\\
\boldsymbol{\Sigma}_{t|t-1} &= \boldsymbol{S}_{t|t-1}\boldsymbol{S}_{t|t-1}^{\intercal}, && \text{Factorize}\\
\boldsymbol{\mathcal{X}}^{*}{}_{i,t|t-1} &= \boldsymbol{S}_{t|t-1}\boldsymbol{\xi}_i + \boldsymbol{\mu}_{t|t-1}, && \text{New cubature point set}\\
\boldsymbol{\mathcal{Y}}_{i,t|t-1} &= \boldsymbol{h}(\boldsymbol{\mathcal{X}}^{*}{}_{i,t|t-1}), && \text{Transformed observations}\\
\hat{\boldsymbol{y}}_t &= \tfrac{1}{2\mathtt{X}}\textstyle\sum_{i=1}^{2\mathtt{X}} \boldsymbol{\mathcal{Y}}_{i,t|t-1}, && \text{Predicted observations}
\end{aligned}
$$

Update step

$$
\begin{aligned}
\boldsymbol{\Sigma}_{\boldsymbol{Y},t|t-1} &= \tfrac{1}{2\mathtt{X}}\textstyle\sum_{i=1}^{2\mathtt{X}} \mathcal{Y}_{i,t|t-1}\mathcal{Y}_{i,t|t-1}^{\intercal} - \hat{\mathbf{y}}_{t|t-1}\hat{\mathbf{y}}_{t|t-1}^{\intercal} + \mathbf{R}, && \text{Innovation covariance}\\
\boldsymbol{\Sigma}_{\boldsymbol{XY},t|t-1} &= \tfrac{1}{2\mathtt{X}}\textstyle\sum_{i=1}^{2\mathtt{X}} \mathcal{X}^{*}{}_{i,t|t-1}\mathcal{Y}_{i,t|t-1}^{\intercal} - \boldsymbol{\mu}_{t|t-1}\hat{\mathbf{y}}_{t|t-1}^{\intercal}, && \text{Cross-covariance}\\
\mathbf{K}_t &= \boldsymbol{\Sigma}_{\boldsymbol{XY},t|t-1}\boldsymbol{\Sigma}_{\boldsymbol{Y},t|t-1}^{-1}, && \text{Kalman gain}\\
\boldsymbol{\mu}_{t|t} &= \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t), && \text{Posterior mean}\\
\boldsymbol{\Sigma}_{t|t} &= \boldsymbol{\Sigma}_{t|t-1} - \mathbf{K}_t\boldsymbol{\Sigma}_{\boldsymbol{Y},t|t-1}\mathbf{K}_t^{\intercal}, && \text{Posterior covariance}
\end{aligned}
$$

The CKF formulation is derivative-free, numerically more stable than the UKF and has the same computational complexity; hence the CKF is an efficient filtering method to be used for nonlinear functions under the Gaussian assumption for the hidden state vector.

2.3 Parameter Estimation in State-Space Models

Modeling dynamic systems using the state-space models involve unknown parameters $\boldsymbol{\theta} \in \mathbb{R}^{\mathtt{P}}$ in the transition and the observation equations along with the hidden states $\boldsymbol{x}$. These parameters might be employed in the linear/nonlinear functions $g(\cdot)$ and $h(\cdot)$ or the error covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ used to model the process errors $\boldsymbol{w} : \boldsymbol{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and the observation errors $\boldsymbol{v} : \boldsymbol{V} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$. Hence, identifying the optimal parameters is critical to the state estimation procedure.

The parameter estimation methods can be broadly classified into two groups: *Bayesian*

estimation (BE) [20] and *maximum likelihood* (MLE) or the *maximum a posteriori estimation* (MAP) [19, 20]. The BE methods such as the *Markov chain Monte Carlo* (MCMC) [38] and the *Laplace approximation* (LAP) [19] approximate the parameters' posterior PDF. Moreover, for specific combinations of the prior and the likelihood, the parameters' posterior PDF can be analytically computed. Such priors are known as the *conjugate priors* [19, 25]. On the other hand, the MLE and the MAP methods such as the gradient-based approaches [19,20,25] and the expectation-maximization (EM) [39] are optimization methods that determine the optimal set of parameters by either maximizing the likelihood function or the product of the likelihood and the prior. The subsequent sections reviews the parameter estimation methods.

2.3.1 Bayesian Estimation

The full Bayesian joint estimation of the parameters and hidden states, requires considering them as random variables in order to infer the joint posterior PDF such that

$$f(\boldsymbol{x}_{0:\mathrm{T}}, \boldsymbol{\theta}|\boldsymbol{y}_{1:\mathrm{T}}) = \frac{f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{x}_{0:\mathrm{T}}, \boldsymbol{\theta}) \cdot f(\boldsymbol{x}_{0:\mathrm{T}}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})}{f(\boldsymbol{y}_{1:\mathrm{T}})}, \tag{2.5}$$

where we choose the parameters' prior PDF $f(\boldsymbol{\theta})$ and evaluate the hidden states' joint prior PDF $f(\boldsymbol{x}_{0:\mathrm{T}}|\boldsymbol{\theta})$, the joint likelihood function $f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{x}_{0:\mathrm{T}}, \boldsymbol{\theta})$ and the joint evidence $f(\boldsymbol{y}_{1:\mathrm{T}})$. We can obtain the parameters' posterior PDF $f(\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathrm{T}})$ by marginalizing out the hidden states $\boldsymbol{x}_{0:\mathrm{T}}$ from the joint PDF defined in Equation 2.5 such that

$$\begin{aligned} f(\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathrm{T}}) &= \int f(\boldsymbol{x}_{0:\mathrm{T}}, \boldsymbol{\theta}|\boldsymbol{y}_{1:\mathrm{T}}) d\boldsymbol{x}_{0:\mathrm{T}}, \\ &= \frac{\int f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{x}_{0:\mathrm{T}}, \boldsymbol{\theta}) \cdot f(\boldsymbol{x}_{0:\mathrm{T}}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta}) d\boldsymbol{x}_{0:\mathrm{T}}}{f(\boldsymbol{y}_{1:\mathrm{T}})}, \\ &= \frac{f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})}{f(\boldsymbol{y}_{1:\mathrm{T}})}, \\ &\propto f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta}), \end{aligned} \tag{2.6}$$

where $f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta})$ is the joint marginal likelihood given the parameters $\boldsymbol{\theta}$ such that $f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta}) = \int f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{x}_{0:\mathrm{T}}, \boldsymbol{\theta}) \cdot f(\boldsymbol{x}_{0:\mathrm{T}}|\boldsymbol{\theta}) d\boldsymbol{x}_{0:\mathrm{T}}$. In most practical situations, evaluating the joint evidence $f(\boldsymbol{y}_{1:\mathrm{T}})$ is computationally intractable, hence it is a common practice to evaluate the unnormalized posterior PDF given by $f(\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathrm{T}}) \propto f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})$ as shown in Equation 2.6. Using the probability theory, the joint marginal likelihood defined in Equation 2.6 can be

expressed as the product of conditional PDFs shown by

$$f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta}) = \prod_{t=1}^{\mathrm{T}} f(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}), \tag{2.7}$$

where the conditional PDF $f(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ is the marginal prior predictive PDF of the observations $\boldsymbol{y}_t$ given $\boldsymbol{y}_{1:t-1}$. Using the conditional independence of the observations $\boldsymbol{y}_t$ and $\boldsymbol{y}_{1:t-1}$ given the states $\boldsymbol{x}_t$, we can recursively compute the marginal prior predictive PDF $f(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ using the filtering procedure shown by

$$f(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}) = \int f(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}) \cdot f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}) d\boldsymbol{x}_t, \tag{2.8}$$

where $f(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$ is the observation model and $f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ is the prior predictive PDF obtained by

$$f(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}) = \int f(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) \cdot f(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}) d\boldsymbol{x}_{t-1}, \tag{2.9}$$

where $f(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})$ is the transition model and $f(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ is the prior knowledge of the hidden states at time $t-1$. In the case of BDLM, the PDF $f(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ is obtained using the Kalman filter procedure such that

$$f(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}_t; \mathbf{C}\boldsymbol{\mu}_{t|t-1}, \mathbf{C}\boldsymbol{\Sigma}_{t|t-1}\mathbf{C}^{\intercal} + \mathbf{R}).$$

The posterior PDF $f(\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathrm{T}})$ defined in Equation 2.6 can be approximated using methods such as the MCMC or in specific cases, by employing conjugate priors. The Laplace approximation can also be used to approximate the posterior using the MLE or the MAP estimates. The subsequent sections review the Bayesian estimation using the MCMC method and the conjugate priors.

Markov Chain Monte Carlo

For Bayesian estimation of parameters, the Markov Chain Monte Carlo (MCMC) methods are used to generate parameter samples $\boldsymbol{\theta}_s$ that are realizations from the *target distribution* $\hat{f}(\boldsymbol{\theta})$ i.e., the parameters' un-normalized posterior PDF $f(\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathrm{T}}) \propto f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})$. The common algorithms for MCMC methods such as the *Metropolis-Hastings* (MH) [20] simulate a Markov chain by first obtaining an *initial state* $\boldsymbol{\theta}_0$ and then using a *proposal distribution* $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$ to obtain a new state $\boldsymbol{\theta}'$ from the current state $\boldsymbol{\theta}$. Depending on an acceptance ratio, the new state is either accepted or rejected. This algorithm is recursively performed for a

total of $\mathtt{S}$ steps, and once, convergence is reached we obtain parameters from a stationary distribution equal to our target distribution. All the accepted samples are used to obtain the empirical posterior moments $\hat{\mathbb{E}}[\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathtt{T}}]$ and $\hat{\text{cov}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathtt{T}})$ such that

$$\begin{aligned}\hat{\mathbb{E}}[\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathtt{T}}] &= \frac{1}{\mathtt{S}}\sum_{s=1}^{\mathtt{S}}\boldsymbol{\theta}_s \\ \hat{\text{cov}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathtt{T}}) &= \frac{1}{\mathtt{S}-1}\sum_{s=1}^{\mathtt{S}}(\boldsymbol{\theta}_s - \hat{\mathbb{E}}[\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathtt{T}}])(\boldsymbol{\theta}_s - \hat{\mathbb{E}}[\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathtt{T}}])^{\intercal}.\end{aligned}$$

The task of choosing a suitable proposal distribution is difficult as poor choices might lead to highly correlated samples or a high rejection rate [20]. The Gaussian PDF is a common choice for the proposal distribution such that $q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}'; \boldsymbol{\theta}, \boldsymbol{\Sigma}_q)$, where the mean $\boldsymbol{\theta}$ is the current location and the $\boldsymbol{\Sigma}_q$ controls the random walk within the parameter space. Methods such as the adaptive MCMC [40] and the robust MCMC [41] are capable of automatically adapting the covariance matrix while performing the MCMC run.

The Hamiltonian Monte Carlo (HMC) [42] is another method that relies on *Hamiltonian dynamics* to propose new parameters. The HMC method adds an *auxiliary momentum variable* r_i to each parameter θ_i to obtain the joint PDF of the parameters and the momentum variables such that $f(\boldsymbol{\theta}, \boldsymbol{r}|\boldsymbol{y}_{1:\mathtt{T}}) = \exp([-H(\boldsymbol{\theta}, \boldsymbol{r})])$, where using classical mechanics the Hamiltonian function $H(\boldsymbol{\theta}, \boldsymbol{r})$ is given by

$$\begin{aligned}H(\boldsymbol{\theta}, \boldsymbol{r}) &= T(\boldsymbol{r}) + V(\boldsymbol{\theta}) \\ &= -\ln f(\boldsymbol{r}) - \ln f(\boldsymbol{y}_{1:\mathtt{T}}|\boldsymbol{\theta}) - \ln f(\boldsymbol{\theta}) \qquad (2.10)\end{aligned}$$

where, assuming $\boldsymbol{r}$ and $\boldsymbol{\theta}$ are independent, $T(\boldsymbol{r}) = -\ln f(\boldsymbol{r})$ is the kinetic energy and $V(\boldsymbol{\theta}) = -\ln f(\boldsymbol{y}_{1:\mathtt{T}}|\boldsymbol{\theta}) - \ln f(\boldsymbol{\theta})$ is the potential energy. Considering that the energy terms $T(\boldsymbol{r})$ and $V(\boldsymbol{\theta})$ are known, new samples for the parameters and the momentum variables are proposed using a *leapfrog method* [20, 42] that numerically solves the Hamiltonian equations shown by

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

where ∇ is the gradient operator. The proposed samples are then passed through an acceptance criteria β^H such that

$$\beta^H = \min\left\{1, \frac{\exp([-H(\boldsymbol{\theta}^{i+1}, \boldsymbol{r}^{i+1})])}{\exp([-H(\boldsymbol{\theta}^{i}, \boldsymbol{r}^{i})])}\right\},$$

where the superscripts i and $i+1$ denote the current and the proposed samples. Compared to the Metropolis Hastings algorithm, the HMC is able to minimize correlation between the accepted samples. However, tuning the parameters such as the step size and the number of steps is critical to efficient exploration of parameters so that the method leads to accurate simulations with less autocorrelation between the consecutive samples [26]. For details regarding the various MCMC methods, the reader can refer to the work of Neal [42] and Brooks [38].

The MCMC methods have shown their efficacy in estimating the parameters' posterior PDF in high-dimensional spaces when analytical implementations for obtaining the posterior are not available. However, for specific cases, conjugate priors do exist that can be utilized to analytically obtain the posterior PDF.

Conjugate Priors

For specific combinations of the prior PDF and the likelihood function, the posterior follows the same type as the prior. Such a prior is known as the conjugate prior using which closed-form solutions are available for the parameters' posterior PDF and the posterior predictive PDF for the outputs. A well-known example for demonstrating the use of conjugate priors is for the Gaussian PDF with a *known variance* and a *random mean*. Let's consider the likelihood function to be a Gaussian PDF such that $Y \sim \mathcal{N}(y; M, \sigma^2)$ with a known variance σ^2 and a random mean $M \sim \mathcal{N}(m; \mu_M, \sigma_M^2)$, where μ_M and σ_M^2 are its hyper-parameters. The Gaussian random variables Y and M can be represented in terms of the standard Gaussian variable ϵ and ζ shown by

$$\begin{aligned} y &= m + \sigma\epsilon, \qquad \epsilon \sim \mathcal{N}(0,1), \\ m &= \mu_M + \sigma_M\zeta,\ \zeta \sim \mathcal{N}(0,1) \end{aligned} \tag{2.11}$$

Considering the joint bivariate Gaussian PDF between Y and M, the posterior moments for $M|y$ are obtained using the Gaussian conditional equations

$$\begin{aligned} \mathbb{E}[M|y] &= \mathbb{E}[M] + \frac{\mathrm{cov}(Y,M)}{\mathrm{var}(Y)}(y - \mathbb{E}[Y]), \\ &= \mu_M + \frac{\sigma_M^2}{\sigma_M^2 + \sigma^2}(y - \mu_M), \\ \mathrm{var}(M|y) &= \mathrm{var}(M) - \frac{\mathrm{cov}(Y,M)^2}{\mathrm{var}(Y)}, \\ &= \frac{\sigma_M^2 \cdot \sigma^2}{\sigma_M^2 + \sigma^2}, \end{aligned} \tag{2.12}$$

where using Equation 2.11 and the properties of random variables, the marginalized moments for Y and the covariance term $\text{cov}(Y, M)$ are derived by

$$\begin{aligned}
\mathbb{E}[Y] &= \mathbb{E}[M] + \cancelto{0}{\mathbb{E}[\sigma\epsilon]} &&= \mu_M, \\
\text{var}(Y) &= \text{var}(M) + \text{var}(\sigma\epsilon) + \cancelto{0}{2\text{cov}(M, \sigma\epsilon)} &&= \sigma_M^2 + \sigma^2, \\
\text{cov}(Y, M) &= \text{cov}(M, M) + \cancelto{0}{\text{cov}(\sigma\epsilon, M)} &&= \sigma_M^2,
\end{aligned}$$

in which the terms $\mathbb{E}[\sigma\epsilon]$ and $\text{cov}(M, \sigma\epsilon)$ are zero as ϵ has a zero mean and is independent of any other random variable. Hence, for the specific combination of the Gaussian prior $f(m) \sim \mathcal{N}(m; \mu_M, \sigma_M^2)$, and the Gaussian likelihood $f(y) \sim \mathcal{N}(y; M, \sigma^2)$, the random mean's posterior PDF is also Gaussian shown by

$$M|y \sim \mathcal{N}\Big(m; \mu_M + \frac{\sigma_M^2}{\sigma_M^2 + \sigma^2}(y - \mu_M), \frac{\sigma_M^2 \cdot \sigma^2}{\sigma_M^2 + \sigma^2}\Big).$$

Similarly for a known mean but a random variance, the conjugate prior for the variance is an inverse gamma PDF which is combined with a Gaussian likelihood function to obtain an inverse gamma posterior PDF itself. For the detailed analytical formulation involving the conjugate priors, the reader can refer to the work of Gelman [19] and Murphy [25].

2.3.2 Maximum Likelihood Estimation

For complex models, Bayesian estimation using sampling methods are computationally expensive [21], and conjugate priors are only applicable for specific cases. Hence, for most practical scenarios, we are forced to rely on point estimates for the parameters instead of a posterior PDF. The point estimates are computed by either maximizing the likelihood (MLE) or the product of the likelihood and the prior (MAP) as shown by

$$\boldsymbol{\theta}^* = \begin{cases} \arg\max_{\boldsymbol{\theta}} \ln f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta}), & \text{MLE} \\ \arg\max_{\boldsymbol{\theta}} \ln f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta}) f(\boldsymbol{\theta}), & \text{MAP} \end{cases} \tag{2.13}$$

where a common practice is to use the log-likelihood instead of the likelihood in the objective function $\tilde{f}(\boldsymbol{\theta}) = \ln f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta})$ for maintaining numerical stability [21]. Note that for brevity, we use the term log-likelihood instead of marginal log-likelihood. The MLE or the MAP estimates can be evaluated using gradient-based approaches such as the *gradient ascent* [21] and the *Newton-Raphson* [21, 25] where we identify new parameters $\boldsymbol{\theta}_{\text{new}}$ by moving towards the maximum point using the gradient $\boldsymbol{\nabla}_{\boldsymbol{\theta}}$ of the objective function evaluated at the old parameters $\boldsymbol{\theta}_{\text{old}}$. The procedure for obtaining $\boldsymbol{\theta}_{\text{new}}$ using the gradient ascent and Newton-

Raphson can be shown as follows

$$\boldsymbol{\theta}_{\text{new}} = \begin{cases} \boldsymbol{\theta}_{\text{old}} + \lambda \cdot \boldsymbol{\nabla}_{\boldsymbol{\theta}} \tilde{f}(\boldsymbol{\theta}_{\text{old}}), & \text{Gradient Ascent} \\ \boldsymbol{\theta}_{\text{old}} - \mathbf{H}[\tilde{f}(\boldsymbol{\theta}_{\text{old}})]^{-1} \cdot \boldsymbol{\nabla}_{\boldsymbol{\theta}} \tilde{f}(\boldsymbol{\theta}_{\text{old}}), & \text{Newton-Raphson} \end{cases}$$

where λ is the learning rate which, in the case of Newton-Raphson, is approximated using the inverse of the negative Hessian matrix $\mathbf{H}$ of the objective function $\tilde{f}(\boldsymbol{\theta})$ evaluated at the old parameters given by $\mathbf{H}[\tilde{f}(\boldsymbol{\theta}_{\text{old}})]^{-1}$. Compared to gradient ascent, the Newton-Raphson method converges faster but has the additional cost of computing the Hessian [21]. For further details on the various gradient-based approaches, the reader can consult the work of Kelley and Carl [43] and Goodfellow [44].

Using the MAP estimate $\boldsymbol{\theta}^*$, the Laplace approximation can be used to approximate the parameters' posterior PDF by a multivariate Gaussian such that

$$f(\boldsymbol{\theta}|\boldsymbol{y}_{1:\mathrm{T}}) \approx \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}^*, \boldsymbol{\Sigma}_{\boldsymbol{\theta}^*}).$$

for which the mean vector $\boldsymbol{\theta}^*$ is the MAP estimate and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}^*}$ is obtained by the inverse Hessian matrix of the negative log-likelihood evaluated at the MAP vector $\boldsymbol{\theta}^*$ shown by

$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}^*} = \mathbf{H}[-\ln f(\boldsymbol{y}_{1:\mathrm{T}}|\boldsymbol{\theta}^*)]^{-1}.$$

However, such approaches are computationally expensive when employed for identifying a large number of parameters as it involves computing the second-order derivatives of the objective function with respect to the model parameters. Moreover, these are offline methods that require retraining the entire model for updating the parameters as new data are collected. Furthermore they are also sensitive to the initial set of parameters and produce sub-optimal results in the presence of saddle points and local maxima [44].

Expectation-Maximization

When *complete data* is available, i.e., all the variables defining the model are fully observed, computing the MLE or the MAP estimates is straightforward [20, 25]. This is because direct optimization is feasible as the marginal likelihood of the data can be computed [20]. However, when we have *incomplete data*, i.e., the observed variables are partially known or involves unknown hidden variables [21], the marginal likelihood is intractable. For such cases, when we cannot compute the likelihood function directly, the *expectation-maximization* (EM) algorithm [45] finds its application for obtaining the MLE or MAP estimates.

Considering that the data $\mathcal{D} = \{\boldsymbol{y}_{1:\mathtt{T}}, \boldsymbol{x}_{0:\mathtt{T}}, \boldsymbol{x} \in \mathbb{R}^{\mathtt{X}}, \boldsymbol{y} \in \mathbb{R}^{\mathtt{Y}}\}$ is fully observed, the *complete data log-likelihood* given the parameters $\boldsymbol{\theta}$ is obtained by

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) &= \ln \int f(\boldsymbol{y}_{1:\mathtt{T}}, \boldsymbol{x}_{0:\mathtt{T}}|\boldsymbol{\theta}) d\boldsymbol{x}_{0:\mathtt{T}}, \\ &= \ln \int q(\boldsymbol{x}_{0:\mathtt{T}}) \cdot \frac{f(\boldsymbol{y}_{1:\mathtt{T}}, \boldsymbol{x}_{0:\mathtt{T}}|\boldsymbol{\theta})}{q(\boldsymbol{x}_{0:\mathtt{T}})} d\boldsymbol{x}_{0:\mathtt{T}}, \end{aligned} \tag{2.14}$$

where $q(\boldsymbol{x}_{0:\mathtt{T}})$ is an arbitrary PDF of the hidden variables $\boldsymbol{x}_{0:\mathtt{T}}$. Unfortunately, the complete data log-likelihood cannot be computed as the hidden variables are unknown. The EM algorithm employs the *Jensen's inequality* [25] in order to obtain a lower bound for the likelihood function $\mathcal{L}(\boldsymbol{\theta})$ defined in Equation 2.14 such that

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) &\geq \int q(\boldsymbol{x}_{0:\mathtt{T}}) \ln \left[\frac{f(\boldsymbol{y}_{1:\mathtt{T}}, \boldsymbol{x}_{0:\mathtt{T}}|\boldsymbol{\theta})}{q(\boldsymbol{x}_{0:\mathtt{T}})} \right] d\boldsymbol{x}_{0:\mathtt{T}}, \\ &\geq \int q(\boldsymbol{x}_{0:\mathtt{T}}) \cdot \ln[f(\boldsymbol{y}_{1:\mathtt{T}}, \boldsymbol{x}_{0:\mathtt{T}})|\boldsymbol{\theta}] - q(\boldsymbol{x}_{0:\mathtt{T}}) \cdot \ln[q(\boldsymbol{x}_{0:\mathtt{T}})] d\boldsymbol{x}_{0:\mathtt{T}}, \\ &\geq \int q(\boldsymbol{x}_{0:\mathtt{T}}) \cdot \ln[f(\boldsymbol{y}_{1:\mathtt{T}}, \boldsymbol{x}_{0:\mathtt{T}})|\boldsymbol{\theta}] d\boldsymbol{x}_{0:\mathtt{T}}, \end{aligned} \tag{2.15}$$

where the lower bound is simplified by removing the second term as it does not depend on $\boldsymbol{\theta}$. It can be shown that choosing a PDF such that $q(\boldsymbol{x}_{0:\mathtt{T}}) = f(\boldsymbol{x}_{0:\mathtt{T}}|\boldsymbol{y}_{1:\mathtt{T}}, \boldsymbol{\theta})$ satisfies the inequality shown in Equation 2.15 [20, 25], using which we can define the *expected complete data log-likelihood* $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$ at the current parameters $\boldsymbol{\theta}^i$ as shown by

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i) = \int f(\boldsymbol{x}_{0:\mathtt{T}}|\boldsymbol{y}_{1:\mathtt{T}}, \boldsymbol{\theta}^i) \cdot \ln[f(\boldsymbol{y}_{1:\mathtt{T}}, \boldsymbol{x}_{0:\mathtt{T}})|\boldsymbol{\theta}] d\boldsymbol{x}_{0:\mathtt{T}}, \tag{2.16}$$

where using the Markov hypothesis, the complete data log-likelihood can be formulated as

$$\ln[f(\boldsymbol{y}_{1:\mathtt{T}}, \boldsymbol{x}_{0:\mathtt{T}})|\boldsymbol{\theta}] = \ln f(\boldsymbol{x}_0|\boldsymbol{\theta}) \cdot \sum_{t=1}^{\mathtt{T}} \ln f(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) \cdot \sum_{t=1}^{\mathtt{T}} \ln f(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}).$$

In the case of BDLM, the expectations involved in Equation 2.16 can be computed in closed-form using the *Rauch-Tung-Streibel* (RTS) smoother [46] while for nonlinear dynamic models these have to be approximated using the *Gaussian smoothers* [47] or the *particle smoothers* [20]. The EM algorithm is carried out by starting from an initial set of parameters $\boldsymbol{\theta}^{(0)}$ and performing the following two steps for the i^{th} iteration:

- E-step: obtain the expected complete data likelihood $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$ defined in Equation 2.16 given the current parameters $\boldsymbol{\theta}^i$.
- M-step: find the new MLE estimates $\boldsymbol{\theta}^{i+1}$ by maximizing the expected complete data

log-likelihood shown by $\boldsymbol{\theta}^{i+1} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$.

It turns out that the EM algorithm is specifically advantageous for BDLMs as the optimization at the M-step can be performed analytically by setting the gradient to zero such that $\frac{\partial \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i)}{\partial \boldsymbol{\theta}}$. Note that for computing MAP estimates, we need to maximize $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i) + \ln f(\boldsymbol{\theta})$ instead of $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$. Even though the EM algorithm monotonically increases the complete data likelihood function, it is common to get stuck at local maxima or saddle points based on initial choice of parameters [25].

The Bayesian estimation and the maximum likelihood estimation methods discussed in this section are widely used for parameter estimation in state-space models. However, in existing literature there are methods called the *adaptive Kalman filters* (AKF) specifically designed for the estimation of the process error and the measurement error covariance matrices to be reviewed in the following section.

2.3.3 Adaptive Kalman Filters

For linear dynamic systems, the Kalman filter is an exact state estimator if the process error ($\mathbf{Q}$) and the measurement error ($\mathbf{R}$) covariance matrices are known [48]. In most practical situations, the deterministic part of the model which includes the transition and the observation models is formulated based on known system dynamics. In contrast, the stochastic part representing the process and the measurement errors is either unknown or only approximately known [48,49]. Previous studies have also shown that using incorrect error covariance matrices may result in large estimation errors or even cause divergence [48,50,51]. Hence, the accurate estimation of the error covariance matrices is necessary for the exact state estimation [50, 52].

As presented in the Section 2.3.2, the unknown error variance parameters in the $\mathbf{Q}$ and the $\mathbf{R}$ matrices can be obtained using the maximum likelihood estimation methods such as gradient-based approaches [21, 25] or the EM algorithm [45]. On the other hand, gradient-based MLE methods are sensitive towards the initialization of the parameters, are computationally demanding, and may provide sub-optimal estimates in case of small datasets and in the presence of large number of parameters [53–55]. Even though the EM algorithm can be utilized for complex models where computing the gradients is numerically infeasible [20] or when data is missing [56], it cannot guarantee to provide an optimal set of parameters because of its sensitivity to parameter initialization. Moreover such methods can only be applied offline, and as a result, the entire model needs to be retrained to obtain new parameter estimates as new data points arrive [25]. Online learning methods such as *Rao-Blackwellized*

Particle Filtering (RBPF) [57] can provide reliable estimates for small datasets, but it is computationally demanding for complex models [58].

The adaptive Kalman filters (AKF) were developed to estimate both the states and the error covariance matrices together by adaptively adjusting the Kalman filter to the measured data such that the estimation errors can be either bounded or reduced [50]. The AKFs are broadly grouped as follows: 1) *correlation methods* [48, 59–62], 2) *covariance-matching methods* [63–65], 3) *maximum likelihood methods* [66, 67], and 4) *Bayesian methods* [68–70].

Correlation Methods

The *innovation correlation method* (ICM) [48] is one of such approach which is based on the fact that the innovation sequence is white for an optimal Kalman filter or otherwise, there must be correlation between the innovations. The ICM method uses the auto-correlation function of the innovations to form a system of linear equations involving the unknown covariance matrices. A least-square method is used to solve these equations simultaneously to obtain the estimates for the $\mathbf{Q}$ and the $\mathbf{R}$ matrices. The literature contain several other correlation methods such as the measurement average correlation method (MACM) [60], the direct correlation method (DCM) [61], and the measurement correlation method (MDCM) [62]. However, these methods are strictly restricted to the case of linear dynamic models and are only capable of providing point estimates [49, 71, 72]. Moreover, in order to ensure the asymptotic convergence, the $\mathbf{Q}$ and the $\mathbf{R}$ matrices need to be updated over several iterations utilizing the entire data [48], and hence the method can only be applied offline.

Covariance-Matching Methods

The basic idea in *covariance-matching method* (CMM) is to match the actual error covariance matrices to the theoretical values computed by the filter. Considering that the process errors $\boldsymbol{W} \sim \mathcal{N}(\boldsymbol{q}, \mathbf{Q})$ and the measurement errors $\boldsymbol{V} \sim \mathcal{N}(\boldsymbol{r}, \mathbf{R})$ are independent and identically distributed having a constant mean and a covariance, the CMM method provides an unbiased estimator for the parameters $\boldsymbol{q}, \boldsymbol{r}, \mathbf{Q}$, and $\mathbf{R}$ by computing the sample mean and the sample covariance over the collected error samples. Myers and Tapley [63] proposed the *adaptive limited memory filter* (ALMF) [63] that uses such a technique where sample covariance matrices are computed at each time step for both the state prediction error and the innovation sequence using either the entire past data or over a moving window. However, such methods produce biased estimates for the covariance matrices and often fail to ensure the positive-definiteness of matrices when the sample size of the data is small [61, 73].

Maximum Likelihood Methods

The maximum likelihood methods primarily uses a gradient-based or expectation-maximization algorithm to estimate the error covariance matrices. Shumway and Stoffer [66] provided a framework that uses the EM algorithm to obtain both the states and the error covariance matrices even when the data is irregularly spaced. The main disadvantages of these methods are its computational demand and the fact that these can be only applied offline [73].

Bayesian Methods

An extensive amount of literature exists for the AKF methods under the Bayesian category. The Bayesian methods include *state augmentation methods* primarily relying on nonlinear estimation techniques such as the EKF [73], the UKF [74], or the particle filters [54, 75] for the joint estimation of both the states and the error covariance matrices (ECM). While most methods in this category identify the error variances offline, Kontoroupi and Smyth [74] provided an online estimation method by employing an approximation of the inverse gamma conjugacy. The method enables the online estimation by using the mode of the inverse gamma PDF as a point estimate to solve the intractable integral required for obtaining the marginalized posterior moments for the hidden states. The method is also applicable for multivariate cases where instead of the inverse gamma, the inverse Wishart PDF is employed to estimate the error covariance matrices. The method provides an effective technique for accurately estimating both the mean and full covariance matrix associated with the observation error, but has a limited accuracy for the moments associated with the process error [74]. The primary cause for this inaccuracy is attributed to the approximation of using the residuals from the UKF filter for updating the hyperparameters of the inverse-gamma or inverse Wishart PDF at each time step. These residuals are intrinsically uncertain while the method considers them to be deterministic values.

Another Bayesian method is the *interactive multiple models* [76] that defines multiple models each having a separate dynamic model with its own ECM as well as the transitional probabilities between one model i and another model j at any given time step t. A set of several Kalman filters are run on parallel to evaluate the state estimates for each model simultaneously. The model selection is performed according to the Bayes' rule using the likelihood obtained for each model given the ECM and the prior probabilities assigned to each model. In order to obtain the combined state estimate, the posterior state estimates from each model are weighted according to the updated model probabilities. This method is applicable for both stationary and non-stationary error variances and has the potential of providing exact estimates when an infinitely large numbers of models are considered. However, the

computational cost makes it practically infeasible [68].

The *variational Bayes* (VB) methods have been proposed to approximate the intractable joint posterior PDF of the states and the covariance matrices at a comparatively lower computational cost than using the particle filters or the multiple model methods [49, 68]. Sarkka and Nummenmaa proposed the VB-AKF method [68] that attempts to estimate the approximate joint posterior PDF of the states and the unknown diagonal $\mathbf{R}$ matrix by using an inverse gamma prior for each of the error variance terms. This is because using an inverse gamma prior combined with a Gaussian likelihood results in an inverse gamma posterior itself, hence the conjugate prior choice for the unknown variance of a Gaussian PDF [19]. Sarkaa and Hartikainen [77] extended the VB-AKF method to obtain the full $\mathbf{R}$ matrix using an inverse Wishart conjugate prior. The method can also be applied to nonlinear state-space models using the available approximate Gaussian filters [68]. However, the method requires an exact knowledge of the $\mathbf{Q}$ matrix which is not known for most practical applications [78]. Moreover, the same methodology could not be applied to obtain the $\mathbf{Q}$ matrix, since it does not appear in simple conjugate prior form, as opposed to the $\mathbf{R}$ matrix [77, 79]. Furthermore, a heuristic dynamic model is suggested for the error covariance matrix using a user-defined forgetting factor ρ and the VB method requires a fixed-point iteration to estimate the scale parameter of the inverse gamma PDF at each time step. Ardeshiri et al. [80] proposed a VB based RTS smoother to obtain both the $\mathbf{Q}$ and $\mathbf{R}$ matrices, but it can only evaluate the error covariance matrices offline [78, 79].

Huang et al. [78] proposed an online VB-AKF method, referred to as VBAKF-PR, to directly estimate the joint distribution of the states, the state prediction error covariance matrix, and the $\mathbf{R}$ matrix by using the conjugacy of the inverse Wishart prior for the covariance matrices. However, the method requires an accurate nominal $\mathbf{Q}$ matrix based on problem-specific expertise without which the performance degrades drastically. Moreover, the method has additional parameters such as the tuning parameter, the forgetting factor, and the number of fixed-point iterations per time step that needs to be tuned. The *sliding window variational adaptive Kalman filter* (SWVAKF) allows estimation of the $\mathbf{Q}$ and the $\mathbf{R}$ matrices by performing three steps: a forward Kalman filtering step to obtain the state estimates by using the error covariance matrices at the previous time step, a backward Kalman smoothing step to obtain the smoothed PDF of the state estimates over a sliding window of discrete number of time steps, and the online estimation of the $\mathbf{Q}$ and the $\mathbf{R}$ matrices using the smoothed posterior PDF of the states. The SWVAKF overcomes the limitation of the VBAKF-PR as it is robust to the initialization of the nominal $\mathbf{Q}$ matrix and proved to be computationally more efficient by avoiding the fixed-point iteration step. However, this method is only shown to be applicable for linear dynamic models [81].

As discussed in this section, there are several methods that have been proposed to estimate the error covariance matrices. In general, most methods are capable of estimating the $\mathbf{R}$ matrix [48, 68] or can be identified directly from the measuring devices [73], however the $\mathbf{Q}$ matrix is neither known in practice nor is it straightforward to estimate it using the available methods [68, 78]. Moreover, most methods are either offline in nature [48, 66], restricted to linear dynamic systems [70, 78] or are computationally demanding [20, 75, 76]. Furthermore, there is no closed-form method to obtain these matrices, and only a limited number of methods can estimate time-varying error matrices, but only through a heuristic dynamic model [78]. In addition, none of the available methods have demonstrated the capacity to estimate a high-dimensional full $\mathbf{Q}$ matrix. Hence, there is still the challenge to develop a method that performs closed-form online estimation of not only constant, but also time-varying $\mathbf{Q}$ matrices for a linear as well as a nonlinear dynamic model and that is still scalable to high-dimensional domains.

2.4 Bayesian Neural Networks

Bayesian neural networks (BNN) [82, 83] are another class of probabilistic models that provides a Bayesian treatment for the neural network's (NN) parameters. The typical NN uses a deterministic set of model parameters to provide point estimates for the model predictions. On the other hand, the BNN uses Bayesian inference to learn the posterior over the neural networks' parameters. Hence, such a model provides predictive uncertainty, robustness against over-fitting, and enables learning from smaller datasets [83].

For a supervised learning problem, neural networks are used to find a parameterized function $\boldsymbol{y} = g(\boldsymbol{x}, \boldsymbol{\theta})$ given the data $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{\mathtt{D}}$, where $\mathtt{D}$ refers to the total number of data points, $\boldsymbol{x}_i \in \mathbb{R}^{\mathtt{X}}$ are the input features, and $\boldsymbol{y}_i \in \mathbb{R}^{\mathtt{Y}}$ are the observations corresponding to the i^{th} data point. The model parameters $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{b}\}$ comprise the set of deterministic weights and biases that stores the information of the learned function $g(\cdot)$. Using this function, the model predictions $\boldsymbol{y}$ are obtained for the given set of inputs $\boldsymbol{x}$. However in BNN, prior PDFs are placed over the model parameters $f(\boldsymbol{\theta})$ and the objective is to learn the posterior PDF of the parameters $f(\boldsymbol{\theta}|\mathcal{D})$ that best describes the data $\mathcal{D}$. Using Bayes rule, $f(\boldsymbol{\theta}|\mathcal{D})$ can be obtained by

$$f(\boldsymbol{\theta}|\mathcal{D}) = \frac{f(\mathcal{D}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})}{f(\mathcal{D})}, \tag{2.17}$$

where $f(\mathcal{D}|\boldsymbol{\theta})$ is the likelihood of data given the parameters, $f(\boldsymbol{\theta})$ is the prior PDF of the model parameters, and $f(\mathcal{D}) = \int f(\mathcal{D}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta}) d\boldsymbol{\theta}$ is the model evidence. Using the posterior

PDF $f(\boldsymbol{\theta}|\mathcal{D})$, we can compute the posterior predictive PDF $f(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})$ for the output random variables $\boldsymbol{Y}$ given the input features $\boldsymbol{x}$ by marginalizing the joint PDF $f(\boldsymbol{y},\boldsymbol{\theta}|\boldsymbol{x},\mathcal{D})$ over all possible parameter values so that

$$f(\boldsymbol{y}|\boldsymbol{x},\mathcal{D}) = \int f(\boldsymbol{y},\boldsymbol{\theta}|\boldsymbol{x},\mathcal{D})d\boldsymbol{\theta} \ = \int f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}, \tag{2.18}$$

where $f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})$ is the model that generates the outputs $\boldsymbol{y}$ given the inputs $\boldsymbol{x}$ and the parameters $\boldsymbol{\theta}$.

However, solving the integrals to obtain the model evidence $f(\mathcal{D})$ or the predictive PDF $f(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})$ as defined in Equations 2.17 & 2.18 is computationally intractable. As a result, several approximate methods were proposed for quantifying the predictive uncertainty including *variational inference* [10, 11, 13], *sampling-based* methods, [82, 84] and *ensemble model combination* [8, 9]. The subsequent section provides a detail overview of the existing approximate inference methods.

2.4.1 Approximate Inference Methods

Since exact Bayesian inference is computationally intractable for neural networks (NN), many approximate inference methods have been proposed in the literature for employing *Bayesian neural networks* (BNN) [8, 9, 85]. The *Laplace approximation* [86] was used for obtaining the posterior distribution of the parameters in NN. In its original form, this method is computationally inefficient for large neural networks as it requires the computation of a full inverse Hessian matrix. The *Hamiltonian Monte Carlo* (HMC) is a Monte-Carlo sampling method that is considered as the reference for BNN [10, 82]. For most practical applications, the method lacks practical scalability and requires problem-specific parameter tuning [10].

A scalable *variational inference* (VI) method [87] was proposed for estimating the posterior distribution of the parameters in neural networks by maximizing the *evidence lower bound* (ELBO) of the marginal log-likelihood. This required computing the intractable expected log-likelihood term of the ELBO using a *Monte Carlo* approximation and using the *stochastic gradient descent* (SGD) approach to optimize the approximate lower bound [10,83]. However, VI tends to perform poorly for large datasets as the optimization procedure using SGD requires multiple passes through the entire data which is practically infeasible for large datasets [10]. *Probabilistic backpropagation* (PBP) [10] uses expectation propagation to estimate the parameters in BNN. Unlike backpropagation, PBP computes the posterior distribution over the parameters by propagating backward the gradients of the marginal likelihood with respect to the parameters. PBP is accurate, more computationally efficient than

any form of variational inference or Markov chain Monte Carlo (MCMC) method, and tunes its hyper-parameters automatically [10].

Monte Carlo dropout (MC-dropout) uses *dropout* [88] as a Bayesian approximation for deriving an approximate predictive distribution. MC-dropout computes the predictive uncertainty by averaging over an ensemble of neural networks where each network is trained using dropout [8, 9]. MC-dropout is orders of magnitudes faster than PBP but requires hyper-parameter tuning unlike PBP [8]. MC-dropout motivated researchers to look into ensemble methods in neural networks [9]. *Deep ensembles* [9] provided a simple and scalable method that combines ensembling techniques and adversarial training for obtaining improved predictive performance and out-of-distribution robustness [9].

Variational inference remains an active research area for BNN. Other notable works using variational inference include *Variational matrix Gaussian* (VMG) [12] and *probabilistic back-propagation with the matrix-variate Gaussian* (MVG) distribution (PBP-MV) [11]. VMG uses matrix-variate Gaussian (MVG) [89] priors over the weights compared to PBP which uses independent standard Gaussian priors. VMG shows a better predictive performance compared to PBP and MC-dropout despite its slow convergence. On the other hand, PBP-MV was shown to outperform VMG and converge faster in regression and classification benchmarks [11]. Moreover, Wu et al. pointed out that variational Bayes is limited in practical applications because of its computational constraints and sensitivity to the definition of the prior variances for weights [13]. *Deterministic variational inference* (DVI) [13] was proposed to counter these limitations. DVI provides analytically tractable moment computation and an empirical Bayes [90] approach to automatically assign prior variances for the weights, but has a high computational demand. Recently, the analytically *tractable approximate Gaussian inference* (TAGI) method [7] was proposed which allows for closed-form parameter inference in BNN, without relying on backpropagation. The next section reviews the TAGI method in detail.

2.4.2 Tractable Approximate Gaussian Inference

TAGI [7] performs analytical inference for the parameters in Bayesian neural networks. Here, we summarize the key principles behind TAGI through a feedforward neural network (FNN) architecture. We consider a FNN with $\mathtt{L}$ hidden layers for learning the relationship between the input covariates $\boldsymbol{x} = [x_1 \; x_2 \; \cdots \; x_{\mathtt{X}}]^\intercal, \in \mathbb{R}^{\mathtt{X}}$ and the observed system responses $\boldsymbol{y} = [y_1 \; y_2 \; \cdots \; y_{\mathtt{Y}}]^\intercal, \in \mathbb{R}^{\mathtt{Y}}$. Each of the $\mathtt{L}$ layer of this FNN consists of $\mathtt{A}$ hidden units $z_i^{(j)}, \forall i \in \{1, 2, \cdots, \mathtt{A}\}$ and $\forall j \in \{1, 2, \cdots, \mathtt{L}\}$ for which the corresponding activation units $a_i^{(j)} = \phi(z_i^{(j)})$ are obtained using an activation function $\phi(\cdot)$. The observation model describing the

relationship between the observed system responses $\boldsymbol{y}$ and the model outputs is given by

$$\boldsymbol{y} = \boldsymbol{z}^{(\mathtt{0})} + \boldsymbol{v}, \;\; \boldsymbol{v} : \boldsymbol{V} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{V}}), \tag{2.19}$$

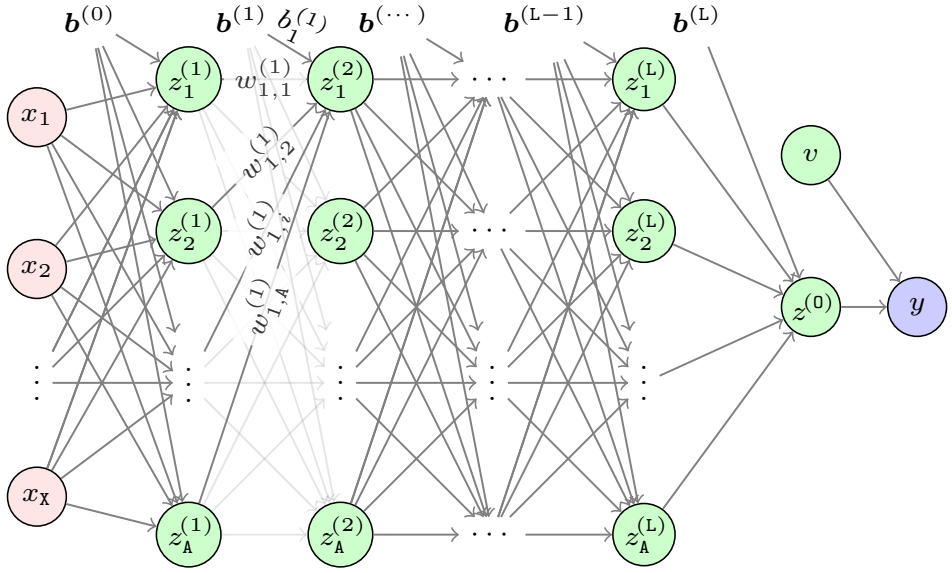
where $\boldsymbol{z}^{(\mathtt{0})} \in \mathbb{R}^{\mathtt{Y}}$ represents the vector of hidden units on the output layer, and $\boldsymbol{v}$ represents the vector of errors with mean zero and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{V}}$. Figure 2.4 shows a graphical model representing the FNN for obtaining a single model output $z^{(\mathtt{0})}$ as a function of the input covariates $\boldsymbol{x}$. The green nodes represent the vector of hidden units and the directed arrows show the flow of information from one node to another. The parameters between any two layers j and $j+1$ are represented by $\boldsymbol{\theta}^{(j)}, \; j \in \{1, 2, \cdots, \mathtt{L}\}$. The observation y, denoted by the purple node, is connected to the output unit $z^{(\mathtt{0})}$, and the error v in accordance to the observation model in Equation 2.19.

In a generic form, TAGI requires propagating uncertainties from the activation hidden units $\boldsymbol{A}^{(j)} \sim \mathcal{N}(\boldsymbol{a}^{(j)}; \boldsymbol{\mu}_{\boldsymbol{A}}^{(j)}, \boldsymbol{\Sigma}_{\boldsymbol{A}}^{(j)})$ in hidden layer j to the i^{th} hidden unit in layer $j+1$ given by

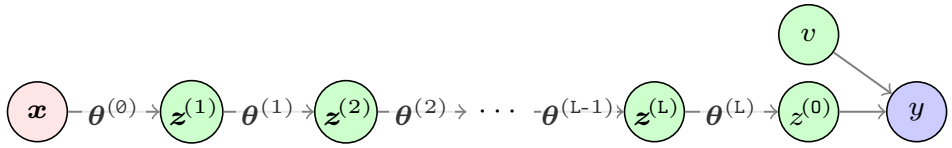
$$Z_i^{(j+1)} = \sum_{k=1}^{\mathtt{A}} W_{i,k}^{(j)} A_k^{(j)} + B_i^{(j)}, \tag{2.20}$$

where the parameters $W_{i,k}^{(j)}$ and $B_i^{(j)}$ are assumed to be Gaussian random variables. Equation 2.20 involves the product of pairs of weights W and activation units A for which the exact moments can be computed using the *Gaussian multiplicative approximation* (GMA) [7, 91] described as follows: Consider $\boldsymbol{X} = [X_1 \; X_2 \; X_3 \; X_4]^\intercal$, a vector of Gaussian random variables such that $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Using the Gaussian moment generating function [7], the following equations hold for the product of any two Gaussian random variables such that

$$\begin{aligned}
\mathbb{E}[X_1 X_2] &= \mu_1 \mu_2 + \text{cov}(X_1, X_2), && (2.21)\\
\text{var}(X_1 X_2) &= \sigma_1^2 \sigma_2^2 + \text{cov}(X_1, X_2)^2 \\
&\quad + 2\text{cov}(X_1, X_2)\mu_1 \mu_2 && (2.22)\\
&\quad + \sigma_1^2 \mu_2^2 + \sigma_2^2 \mu_1^2, \\
\text{cov}(X_3, X_1 X_2) &= \text{cov}(X_1, X_3)\mu_2 + \text{cov}(X_2, X_3)\mu_1, && (2.23)\\
\text{cov}(X_1 X_2, X_3 X_4) &= \text{cov}(X_1, X_3)\text{cov}(X_2, X_4) && (2.24)\\
&\quad + \text{cov}(X_1, X_4)\text{cov}(X_2, X_3) \\
&\quad + \text{cov}(X_1, X_3)\mu_2 \mu_4 \\
&\quad + \text{cov}(X_1, X_4)\mu_2 \mu_3 + \text{cov}(X_2, X_3)\mu_1 \mu_4 \\
&\quad + \text{cov}(X_2, X_4)\mu_1 \mu_3.
\end{aligned}$$



(a) Full representation



(b) Compact representation where $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{b}\}$

Figure 2.4 The (a) Full and (b) Compact representation of a FNN for obtaining a single model output $z^{(\mathtt{0})}$ as a function of the input covariates $\boldsymbol{x}$. The network comprises of $\mathtt{L}$ hidden layers having $\mathtt{A}$ hidden units in any layer $j \in \{1, 2, \cdots, \mathtt{L}\}$. The parameters between any two layers j and $j+1$ are represented by $\boldsymbol{\theta}^{(j)}$. The observation y, denoted by the purple node, is connected to the output unit $z^{(\mathtt{0})}$, and the error v in accordance to the observation model in Equation 2.19.

In TAGI, the GMA Equations 2.21 - 2.24 are leveraged to obtain the moments for the product of pairs of weights W and activation units A.

Even though the true distribution for the product of two Gaussian random variables is not Gaussian, TAGI considers that the sum of a large number of independent product terms approximately results into a Gaussian PDF under the central limit theorem (CLT), given that all activation units A_k are independent from each other. Moreover, using nonlinear activation functions prohibits the analytical tractability for propagating uncertainty through the network. Therefore, TAGI uses a 1^{st} order Taylor series approximation at the expected value of the hidden unit $\mu_{Z_i}^{(j+1)}$ to maintain the analytical tractability when propagating uncertainty through the activation function. In order to maintain a linear computational

complexity with respect to the number of parameters, first, the method employs a diagonal covariance matrix for both the parameters $\boldsymbol{\theta}$ and the hidden units $\boldsymbol{Z}^{(j)}$. Second, it uses a recursive layer-wise Gaussian inference approach that relies on the conditional independence between the hidden units $\boldsymbol{Z}^{(j-1)}$ and $\boldsymbol{Z}^{(j+1)}$ given that the hidden units $\boldsymbol{z}^{(j)}$ are known and that the parameters $\boldsymbol{\theta}$ are independent for each layer.

A two-fold inference step is used for obtaining the posterior moments for the parameters $\boldsymbol{\theta}$ and hidden units $\boldsymbol{Z}^{(j)}$. First, the posterior expected value and diagonal covariance matrix for the output units are obtained such that

$$
\begin{aligned}
f(\boldsymbol{z}^{(0)}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{z}^{(0)}; \boldsymbol{\mu}_{\boldsymbol{Z}^{(0)}|\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{Z}^{(0)}|\boldsymbol{y}}), && (2.25)\\
\boldsymbol{\mu}_{\boldsymbol{Z}^{(0)}|\boldsymbol{y}} &= \boldsymbol{\mu}_{\boldsymbol{Z}^{(0)}} + \boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Z}^{(0)}}^{\intercal} \boldsymbol{\Sigma}_{\boldsymbol{Y}}^{-1} \left(\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{Y}}\right), && (2.26)\\
\boldsymbol{\Sigma}_{\boldsymbol{Z}^{(0)}|\boldsymbol{y}} &= \boldsymbol{\Sigma}_{\boldsymbol{Z}^{(0)}} - \boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Z}^{(0)}}^{\intercal} \boldsymbol{\Sigma}_{\boldsymbol{Y}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Z}^{(0)}}, && (2.27)
\end{aligned}
$$

where $\boldsymbol{\mu}_{\boldsymbol{Z}^{(0)}}$ and $\boldsymbol{\mu}_{\boldsymbol{Y}}$ are the mean vectors for $\boldsymbol{Z}^{(0)}$ and $\boldsymbol{Y}$; $\boldsymbol{\Sigma}_{\boldsymbol{Z}^{(0)}}$ is the prior covariance matrix for $\boldsymbol{Z}^{(0)}$, $\boldsymbol{\Sigma}_{\boldsymbol{Y}}$ is the prior covariance matrix for $\boldsymbol{Y}$, and $\boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Z}^{(0)}}$ is the prior cross-covariance between $\boldsymbol{Z}^{(0)}$ and $\boldsymbol{Y}$. Second, the Rauch-Tung-Striebel (RTS) smoother [92] is used to perform the layer-wise backward inference pass using the posterior knowledge for the output units obtained by Equations 2.25 – 2.27. The posterior moments for the parameters $\boldsymbol{\theta}$ and the hidden units $\boldsymbol{Z}$ are obtained following

$$
\begin{aligned}
f(\boldsymbol{Z}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_{\boldsymbol{Z}|\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{Z}|\boldsymbol{y}}), & f(\boldsymbol{\theta}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\boldsymbol{\theta}|\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}|\boldsymbol{y}}),\\
\boldsymbol{\mu}_{\boldsymbol{Z}|\boldsymbol{y}} &= \boldsymbol{\mu}_{\boldsymbol{Z}} + \boldsymbol{J}_{\boldsymbol{Z}} \left(\boldsymbol{\mu}_{\boldsymbol{Z}^{+}|\boldsymbol{y}} - \boldsymbol{\mu}_{\boldsymbol{Z}^{+}}\right), & \boldsymbol{\mu}_{\boldsymbol{\theta}|\boldsymbol{y}} &= \boldsymbol{\mu}_{\boldsymbol{\theta}} + \boldsymbol{J}_{\boldsymbol{\theta}} \left(\boldsymbol{\mu}_{\boldsymbol{Z}^{+}|\boldsymbol{y}} - \boldsymbol{\mu}_{\boldsymbol{Z}^{+}}\right),\\
\boldsymbol{\Sigma}_{\boldsymbol{Z}|\boldsymbol{y}} &= \boldsymbol{\Sigma}_{\boldsymbol{Z}} + \boldsymbol{J}_{\boldsymbol{Z}} \left(\boldsymbol{\Sigma}_{\boldsymbol{Z}^{+}|\boldsymbol{y}} - \boldsymbol{\Sigma}_{\boldsymbol{Z}^{+}}\right) \boldsymbol{J}_{\boldsymbol{Z}}^{\intercal}, & \boldsymbol{\Sigma}_{\boldsymbol{\theta}|\boldsymbol{y}} &= \boldsymbol{\Sigma}_{\boldsymbol{\theta}} + \boldsymbol{J}_{\boldsymbol{\theta}} \left(\boldsymbol{\Sigma}_{\boldsymbol{Z}^{+}|\boldsymbol{y}} - \boldsymbol{\Sigma}_{\boldsymbol{Z}^{+}}\right) \boldsymbol{J}_{\boldsymbol{\theta}}^{\intercal},\\
\boldsymbol{J}_{\boldsymbol{Z}} &= \boldsymbol{\Sigma}_{\boldsymbol{Z}\boldsymbol{Z}^{+}} \boldsymbol{\Sigma}_{\boldsymbol{Z}^{+}}^{-1}, & \boldsymbol{J}_{\boldsymbol{\theta}} &= \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{Z}^{+}} \boldsymbol{\Sigma}_{\boldsymbol{Z}^{+}}^{-1},
\end{aligned}
\qquad (2.28)
$$

where the short-hand notations for the parameters and hidden units in the j^{th} and the subsequent layer are $\{\boldsymbol{\theta}^{+}, \boldsymbol{Z}^{+}\} \equiv \{\boldsymbol{\theta}^{(j+1)}, \boldsymbol{Z}^{(j+1)}\}$ and $\{\boldsymbol{\theta}, \boldsymbol{Z}\} \equiv \{\boldsymbol{\theta}^{(j)}, \boldsymbol{Z}^{(j)}\}$. The posterior inference for the parameters $\boldsymbol{\theta}$ shown in Equation 2.28 is done recursively using either a single observation or a batch of them. Moreover, this recursive inference process is done over multiple epochs $\mathtt{E} > 1$ to overcome the limitation of having weakly informative priors for the parameters $f(\boldsymbol{\theta}|\boldsymbol{\eta}^{(\emptyset)}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\boldsymbol{\theta}}^{(\emptyset)}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{(\emptyset)})$ for which the hyperparameters $\boldsymbol{\eta}^{(\emptyset)} = \{\boldsymbol{\mu}_{\boldsymbol{\theta}}^{(\emptyset)}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{(\emptyset)}\}$ are defined using either the Xavier's [93] or He's [16] approach. Hence, multiple iterations are carried out using a training set $\mathcal{D}_{\mathtt{T}}$ and a validation set $\mathcal{D}_{\mathtt{V}}$, where the posterior parameters' hyperparameter values at the i^{th} iteration $\boldsymbol{\eta}^{(i)} = \{\boldsymbol{\mu}_{\boldsymbol{\theta}|\mathcal{D}_{\mathtt{T}}}^{(i)}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}|\mathcal{D}_{\mathtt{T}}}^{(i)}\}$ are used as that of the prior's

for the $i + 1^{th}$ iteration shown by

$$\boldsymbol{\eta}^{(i+1)} = \text{TAGI}(\mathcal{D}_{\text{T}}, \boldsymbol{\eta}^{(i)}).$$

In order to avoid overfitting, the learning process is stopped using an early-stopping procedure when the marginal log-likelihood for the validation set $\mathcal{D}_{\text{V}}$ reaches its maximum value. Using multiple epochs to learn the optimal hyperparameters is equivalent to the empirical Bayes approach [94], but the maximization of the marginal log-likelihood is performed implicitly rather explicitly [7].

TAGI provides an analytically tractable method for inferring the posterior expected values and diagonal covariance matrix of the neural network's parameters. It was shown to provide a competitive performance with regard to regression and classification benchmarks [7], as well as on applications such as adversarial attacks, optimization, and continuous-action reinforcement learning [95]. However, a key limitation is that the original version of TAGI can only handle homoscedastic aleatory uncertainty for which the error variance σ_V^2 is considered as a hyperparameter that needs to be identified separately from the analytical parameter inference, which makes it computationally expensive to identify [7].

2.5 Conclusion

The literature review has covered key probabilistic models that can quantify predictive uncertainty. First, the review examined the state-space models (SSM) used for modeling linear and nonlinear dynamic systems comprising hidden states, and the filtering methods that provide a probabilistic approach for updating the knowledge of these hidden states using observations at any time. We can perform the exact inference using the Kalman filter under the Gaussian assumption for the hidden states and the linear assumption for the transition and observation models. However, we depend on nonlinear filtering methods such as the unscented Kalman filter, cubature Kalman filter, and particle filters for handling nonlinear models which are computationally expensive. Furthermore, parameter estimation is typically the most computationally demanding task in the state estimation procedure. This task relies either on Bayesian methods such as the Markov Chain Monte Carlo (MCMC) which are computationally expensive, or on maximum likelihood methods such as gradient-based approaches, which are sensitive to parameter initialization, are computationally demanding, and require model retraining with each additional data point. In particular, modeling the process and observation errors associated with the transition and the observation models are critical for an accurate hidden state estimation. These errors are quantified by the error

variance terms in the error covariance matrices. The adaptive Kalman filter (AKF) methods are specifically designed for estimating the error covariance matrices but it still remains a challenge to develop a method able to perform closed-form online estimation of the process error covariance matrix $\mathbf{Q}$.

Second, the review investigated the Bayesian neural networks (BNN), which provide a probabilistic approach to the traditional neural networks with regard to quantifying uncertainty over the parameters and model predictions using a Bayesian framework. As exact Bayesian inference is intractable in neural networks, there are many approximate methods employing either variational inference, sampling-based methods or ensemble of models for quantifying the predictive uncertainty. The tractable approximate Gaussian inference (TAGI) allows for analytical parameter inference for BNNs and was shown to be competitive in comparison to existing networks trained with backpropagation. However, in its current version, a key limitation of TAGI is that it can only model homoscedastic aleatory uncertainty as quantified by the constant error variance parameter.

Overall, the review identifies that the parameter inference step is critical for improving the applicability and the scalability of our probabilistic models. The following chapters will describe methods that can overcome each of the aforementioned limitations related to parameter inference for the state-space models and Bayesian neural networks.

CHAPTER 3 The Gaussian Multiplicative Approximation for State-Space Models

3.1 Introduction

This chapter presents how we propose to take advantage of the Gaussian multiplicative approximation (GMA), see Section 2.4.2, in the context of state-space models. Using the GMA, the closed-form solutions for the first two moments for the product of two Gaussian random variables can be obtained. The potential of combining the GMA with the Bayesian dynamic linear models (BDLM) is illustrated through the development of generic components called (1) the *online autoregressive* (`OAR`) that can estimate both the `AR` state ($x^{\texttt{AR}}$) and the `AR` parameter ($\phi^{\texttt{AR}}$) together; (2) the *trend multiplicative* (`TM`) for multiplicative seasonality model to identify a non-harmonic periodic pattern whose amplitude changes linearly with time; and (3) the *double kernel regression* (`DKR`) to identify non-harmonic periodic pattern that involves the product of two periodic kernel regression components.

The chapter includes the detailed mathematical formulation for the proposed method and the procedure to perform state estimation using the BDLM framework. Thereafter, three applied examples are presented for showcasing the capacity of using the GMA for both real and synthetic datasets. The main contributions of this chapter are to

- Provide an analytical method that is applicable to multiplicative state-space models by providing explicitly the equations for moment computation of a product term.
- Enable the online estimation of model parameters as hidden states.
- Provide generic components called the online autoregressive, trend multiplicative and double kernel regression in the BDLM framework.
- Validate and verify the proposed method with real and synthetic datasets.
- Provide a method that exceed the performance of the cubature Kalman filter in terms of accuracy and computational cost in the SHM-based case studies.

3.2 Gaussian Multiplicative Approximation

This section presents the *Gaussian multiplicative approximation* for computing the moments associated with the product of two Gaussian random variables and demonstrates its application in state estimation using BDLM.

3.2.1 Moments of Product Term

Consider the case where the variables $\boldsymbol{x} = [x_1 \;\; x_2]^\intercal$ are the input of the nonlinear function,

$$g(\boldsymbol{x}) = x_1 x_2. \tag{3.1}$$

The goal is to infer the probability density function (PDF) of $\boldsymbol{X}$ indirectly, using an observation y that is defined such that,

$$y = g(\boldsymbol{x}) + v, \;\; v : V \sim \mathcal{N}(v; 0, \sigma_V^2), \tag{3.2}$$

where V is a random variable representing the observation error with zero mean and variance σ_V^2.
The posterior PDF of $\boldsymbol{X}$ given an observation y can be estimated using Bayes theorem as in,

$$f(\boldsymbol{x}|y) = \frac{f(\boldsymbol{x}, y)}{f(y)} = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{\boldsymbol{X}|y}, \boldsymbol{\Sigma}_{\boldsymbol{X}|y}), \tag{3.3}$$

which follows a Gaussian distribution with a mean vector $\boldsymbol{\mu}_{\boldsymbol{X}|y}$ and a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}|y}$ that are given by

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{X}|y} &= \boldsymbol{\mu}_{\boldsymbol{X}} + \frac{\boldsymbol{\Sigma}_{\boldsymbol{X}Y}}{\sigma_Y^2}(y - \mu_Y), \\ \boldsymbol{\Sigma}_{\boldsymbol{X}|y} &= \boldsymbol{\Sigma}_{\boldsymbol{X}} - \frac{\boldsymbol{\Sigma}_{\boldsymbol{X}Y} \cdot \boldsymbol{\Sigma}_{\boldsymbol{X}Y}^\intercal}{\sigma_Y^2}. \end{aligned} \tag{3.4}$$

As seen in Section 2.4.2, Equation 3.3 holds when $f(\boldsymbol{x}, y)$ is Gaussian. However, in the case presented here, y is nonlinearly related to $\boldsymbol{x}$ through Equation 3.2 which makes the joint prior PDF, $f(\boldsymbol{x}, y)$, non-Gaussian. For such a situation, the GMA can provide closed-form inference. The prior moments $\boldsymbol{\mu}_{\boldsymbol{X}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ in Equation 3.4 are obtained as follows

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{X}} &= \mathbb{E}[\boldsymbol{X}] &&= \begin{bmatrix} \mathbb{E}[X_1] \\ \mathbb{E}[X_2] \end{bmatrix}, \\ \boldsymbol{\Sigma}_{\boldsymbol{X}} &= \mathrm{var}(\boldsymbol{X}) &&= \begin{bmatrix} \mathrm{var}(X_1) & \mathrm{cov}(X_1, X_2) \\ \mathrm{cov}(X_2, X_1) & \mathrm{var}(X_2) \end{bmatrix}. \end{aligned}$$

The mean and variance of Y can be obtained by propagating the uncertainty associated with $\boldsymbol{X}$ through the model described in Equation 3.2 so that

$$\begin{aligned} \mu_Y &= \mathbb{E}[Y] &&= \mathbb{E}[g(\boldsymbol{X})] + \mathbb{E}[V] &&= \mathbb{E}[X_1X_2], \\ \sigma_Y^2 &= \mathrm{var}(Y) &&= \mathrm{var}(X_1X_2) + \mathrm{var}(V) &&= \mathrm{var}(X_1X_2) + \sigma_V^2. \end{aligned}$$

Using the *moment generating function* (MGF) [96, 97] for the multivariate Gaussian distribution or 2^{nd} order Taylor series expansion (TSE) [98] of the product of the two Gaussian random variables, the first two moments associated with the product term X_1X_2 can be computed exactly using

$$\mathbb{E}[X_1X_2] = \mu_1\mu_2 + \mathrm{cov}(X_1, X_2), \tag{3.5}$$

$$\begin{aligned} \mathrm{var}(X_1X_2) = {} & \sigma_1^2\sigma_2^2 + \mathrm{cov}(X_1, X_2)^2 + 2\mathrm{cov}(X_1, X_2)\mu_1\mu_2 + \\ & \sigma_1^2\mu_2^2 + \sigma_2^2\mu_1^2, \end{aligned} \tag{3.6}$$

and the covariance between $\boldsymbol{X}$ and Y is given by

$$\boldsymbol{\Sigma}_{\boldsymbol{X}Y} = \mathrm{cov}(\boldsymbol{X}, Y) = \begin{bmatrix} \mathrm{cov}(X_1, X_1X_2) \\ \mathrm{cov}(X_2, X_1X_2) \end{bmatrix}.$$

Similarly, we can also derive the exact solution for the covariance between the product term X_1X_2 and any other Gaussian random variable X_3,

$$\mathrm{cov}(X_3, X_1X_2) = \mathrm{cov}(X_1, X_3)\mu_2 + \mathrm{cov}(X_2, X_3)\mu_1. \tag{3.7}$$

Finally, for the general case, the covariance between any two pair of product terms is given by

$$\begin{aligned} \mathrm{cov}(X_1X_2, X_3X_4) = {} & \mathrm{cov}(X_1, X_3)\mathrm{cov}(X_2, X_4) \\ & + \mathrm{cov}(X_1, X_4)\mathrm{cov}(X_2, X_3) + \mathrm{cov}(X_1, X_3)\mu_2\mu_4 \\ & + \mathrm{cov}(X_1, X_4)\mu_2\mu_3 + \mathrm{cov}(X_2, X_3)\mu_1\mu_4 \\ & + \mathrm{cov}(X_2, X_4)\mu_1\mu_3, \end{aligned} \tag{3.8}$$

where X_1X_2 and X_3X_4 are the product terms of the Gaussian random variables X_1, X_2 and X_3, X_4, respectively. The derivation of the GMA equations using both the MGF and TSE are presented in Appendix A.

Hence, the GMA approximates the distribution for the product term X_1X_2 as a Gaussian

random variable for which the expected value, variance and covariance can be calculated exactly, under the assumption that X_1 and X_2 are themselves Gaussians.

3.2.2 State Estimation

In the context of state-space models, the state estimation for cases involving product terms in the transition model can be performed by combining the linear estimation theory and the GMA. Given the vector of hidden states $\boldsymbol{x} = [x_1 \ x_2]^\intercal$, a generic multiplicative transition model involving the product of the hidden states x_1 and x_2 is given by

$$\begin{aligned} x_{1,t} &= x_{1,t-1}x_{2,t} + w_{1,t}, \quad w_1 : W_1 \sim \mathcal{N}(0, \sigma^2_{W_1}), \\ x_{2,t} &= x_{2,t-1} + w_{2,t}, \qquad\quad w_2 : W_2 \sim \mathcal{N}(0, \sigma^2_{W_2}), \end{aligned} \tag{3.9}$$

where $\boldsymbol{w} = [w_1 \ w_2]^\intercal$ is the vector of error terms associated with the transition model. The hidden states at time $t-1$ is assumed to follow a Gaussian PDF with mean vector and covariance matrix given by

$$\boldsymbol{X}_{t-1|t-1} \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1}),$$

where $\boldsymbol{\mu}_{t-1|t-1} = \mathbb{E}[\boldsymbol{X}_{t-1}|\boldsymbol{y}_{1:t-1}]$, $\boldsymbol{\Sigma}_{t-1|t-1} = \mathrm{cov}(\boldsymbol{X}_{t-1}|\boldsymbol{y}_{1:t-1})$, and $\boldsymbol{y}_{1:t-1} = \{y_1, \ldots, y_{t-1}\}$. In its current form, the transition model as given by Equation 3.9 is nonlinear. However, the nonlinear transition model can be formulated as a linear dynamic model by augmenting the state vector $\tilde{\boldsymbol{x}} = [\boldsymbol{x} \ x^{\mathtt{p}}]^\intercal$ so that

$$\tilde{\boldsymbol{X}}_{t-1|t-1} \sim \mathcal{N}(\tilde{\boldsymbol{x}}; \tilde{\boldsymbol{\mu}}_{t-1|t-1}, \tilde{\boldsymbol{\Sigma}}_{t-1|t-1}),$$

where $\tilde{\boldsymbol{\mu}}_{t-1|t-1} = \begin{bmatrix} \boldsymbol{\mu} \\ \mu^{\mathtt{p}} \end{bmatrix}_{t-1|t-1}$ and $\tilde{\boldsymbol{\Sigma}}_{t-1|t-1} = \begin{bmatrix} \boldsymbol{\Sigma} & \mathrm{cov}(\boldsymbol{X}, X^{\mathtt{p}}) \\ \mathrm{cov}(X^{\mathtt{p}}, \boldsymbol{X}) & (\sigma^{\mathtt{p}})^2 \end{bmatrix}_{t-1|t-1}$.

The hidden state variable $X^{\mathtt{p}} = X_1 X_2$ represents the product term and is assumed to be Gaussian with expected value $\mu^{\mathtt{p}} = \mathbb{E}[X_1 X_2]$ and variance $(\sigma^{\mathtt{p}})^2 = \mathrm{var}(X_1 X_2)$. The covariance terms between $\boldsymbol{X}$ and $X^{\mathtt{p}}$ in $\tilde{\boldsymbol{\Sigma}}_{t-1|t-1}$ is given by

$$\mathrm{cov}(\boldsymbol{X}, X^{\mathtt{p}}) = \begin{bmatrix} \mathrm{cov}(X_1, X_1 X_2) \\ \mathrm{cov}(X_2, X_1 X_2) \end{bmatrix}.$$

Using linear algebra, the transition model in Equation 3.9 can be written as

$$\begin{aligned} \tilde{\boldsymbol{x}}_t &= \mathbf{A}\tilde{\boldsymbol{x}}_{t-1}, \\ \begin{bmatrix} x_1 \\ x_2 \\ x^{\mathtt{p}} \end{bmatrix}_t &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x^{\mathtt{p}} \end{bmatrix}_{t-1}, \end{aligned} \tag{3.10}$$

where $\tilde{\boldsymbol{x}} = [x_1\ x_2\ x^{\mathtt{p}}]^{\intercal}$. The augmented state vector $\tilde{\boldsymbol{X}}_{t|t-1}$ follows a Gaussian PDF given by

$$\tilde{\boldsymbol{X}}_{t|t-1} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_{t|t-1}, \tilde{\boldsymbol{\Sigma}}_{t|t-1}), \tag{3.11}$$

where $\tilde{\boldsymbol{\mu}}_{t|t-1} = \mathbf{A}\tilde{\boldsymbol{\mu}}_{t-1|t-1}$ and $\tilde{\boldsymbol{\Sigma}}_{t|t-1} = \mathbf{A}\tilde{\boldsymbol{\Sigma}}_{t-1|t-1}\mathbf{A}^{\intercal} + \mathbf{Q}$, considering that $\mathbf{Q}$ is the process error covariance matrix. The variance terms in the $\mathbf{Q}$ matrix can be estimated using the methods described in Section 2.3. The observation model is defined as

$$y_t = \mathbf{C}\tilde{\boldsymbol{x}}_t + v_t, \quad v : V \sim \mathcal{N}(0, \mathbf{R}), \tag{3.12}$$

where $\mathbf{C}$ is the observation matrix, v_t is the observation error, and $\mathbf{R} = \sigma_V^2$ is the observation error covariance matrix. Using Equations 3.11 and 3.12, both the prediction and the update steps in the Kalman filter can be carried out for a nonlinear system having product terms. Note that the application of the GMA equations is shown explicitly using a product term in the state vector to simplify the use of the method, hence the last row in the $\mathbf{A}$ matrix has 0's as the product term $x^{\mathtt{p}}$ is a placeholder. However, the GMA equations can also be applied implicitly without the need to store the information specifically in a variable. At each step $t-1$ of the recursive procedure, the GMA Equations 3.5 – 3.8 are applied to compute the moments of the product term X_1X_2 using the moments of X_1 and X_2 obtained from the Kalman filter. The method can also be extended to more than one product terms in the state vector either by placing more placeholders as shown by Equation 3.10 or by computing the moments implicitly. The case studies 2 and 3 in this chapter are examples of more than one product terms in the state vector.

The computational complexity of using the GMA for estimating a state vector of size n is $\mathcal{O}(3n^3)$. When using either the UKF or the CKF (see Section 2.2.2) to perform the same state estimation, the computational complexity is to the order $\mathcal{O}(14n^3)$ [99]. Hence, using the GMA for state estimation of product terms is more than four times faster than using a state-of-the-art nonlinear filter like the CKF. The derivation of the computational complexity for the GMA is provided in Appendix A.

3.3 Applied Examples

This section presents three case studies comparing the performance of the GMA and the CKF for the task of estimating the state variables in SSM having product terms.

3.3.1 Case Study 1: First-Order Online Autoregressive Process (OAR)

This case study presents the application of the proposed online estimation method for the state and parameter of a first-order autoregressive process.

Model Formulation

Consider the transition model for a first-order autoregressive process (AR) given by

$$\underbrace{x_t^{\mathtt{AR}} = \phi^{\mathtt{AR}} x_{t-1}^{\mathtt{AR}} + w_t^{\mathtt{AR}}}_{\text{transition model}},\ \overbrace{w_t^{\mathtt{AR}} : W^{\mathtt{AR}} \sim \mathcal{N}(w^{\mathtt{AR}}; 0, (\sigma^{\mathtt{AR}})^2)}^{\text{process error}},$$

where $x^{\mathtt{AR}}$ is the AR hidden state, $\phi^{\mathtt{AR}}$ is the AR coefficient, and $W^{\mathtt{AR}}$ is the zero-mean Gaussian process error. The AR coefficient $\phi^{\mathtt{AR}}$ can be estimated online by considering it as a hidden state x^{ϕ}. The new transition model is defined as

$$\begin{aligned} x_t^{\mathtt{AR}} &= x_t^{\phi} x_{t-1}^{\mathtt{AR}} + w_t^{\mathtt{AR}}, \\ x_t^{\phi} &= x_{t-1}^{\phi}, \end{aligned} \tag{3.13}$$

where the hidden states at time step $t-1$ are $\boldsymbol{x}_{t-1} = [x^{\mathtt{AR}}\ x^{\phi}]_{t-1}^{\intercal}$. The linear transition model for this case is given by Equation 3.10. The augmented mean vector $\tilde{\boldsymbol{\mu}}_{t-1|t-1}$ and the covariance matrix $\tilde{\boldsymbol{\Sigma}}_{t-1|t-1}$ of $\tilde{\boldsymbol{x}}_{t-1}$ are given by

$$\tilde{\boldsymbol{\mu}}_{t-1|t-1} = \begin{bmatrix} \mathbb{E}[X^{\mathtt{AR}}] \\ \mathbb{E}[X^{\phi}] \\ \mathbb{E}[X^{\phi} X^{\mathtt{AR}}] \end{bmatrix}_{t-1|t-1},$$

$$\tilde{\boldsymbol{\Sigma}}_{t-1|t-1} = \begin{bmatrix} \mathrm{var}(X^{\mathtt{AR}}) & \mathrm{cov}(X^{\mathtt{AR}}, X^{\phi}) & \mathrm{cov}(X^{\mathtt{AR}}, X^{\phi} X^{\mathtt{AR}}) \\ \vdots & \mathrm{var}(X^{\phi}) & \mathrm{cov}(X^{\phi}, X^{\phi} X^{\mathtt{AR}}) \\ \text{sym.} & \dots & \mathrm{var}(X^{\phi} X^{\mathtt{AR}}) \end{bmatrix}_{t-1|t-1},$$

where the elements of $\tilde{\boldsymbol{\mu}}_{t-1|t-1}$ and $\tilde{\boldsymbol{\Sigma}}_{t-1|t-1}$ can be computed analytically using Equations 3.5-3.8. The observation model is given by

$$y_t = \mathbf{C}\tilde{\boldsymbol{x}}_t + v_t, v_t : V \sim \mathcal{N}(v; 0, \sigma_V^2),$$

where the observation matrix is $\mathbf{C} = [1\ 0\ 0]$.

Numerical Example

Simulated data is generated from a first-order AR process using the following parameters: $\sigma_v = 0.1$, $\sigma^{\mathtt{AR}} = 0.05$ and $\phi^{\mathtt{AR}} = 0.9$. Five datasets containing 1000 data points are generated using these parameters with a uniform time step of one unit. The prior knowledge of the hidden states are initialized by

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_0 &= [0\ 0\ 0]^\intercal, \\ \tilde{\boldsymbol{\Sigma}}_0 &= \mathrm{diag}([100\ 100\ 0]). \end{aligned}$$

Both $\tilde{\boldsymbol{\mu}}_{t|t}$ and $\tilde{\boldsymbol{\Sigma}}_{t|t}$ of $\tilde{\boldsymbol{x}}_t$ are estimated using the GMA and the CKF methods. Table 3.1 shows the average results along with their standard deviation for the mean square error (MSE) [25] and log-likelihood (LL) [18, 25] values for $x^{\mathtt{AR}}$, x^{ϕ}, as well as their joint log-likelihood. Figure 3.1 compares the actual and estimated hidden state values obtained using the GMA and the CKF methods, which shows the convergence of the estimated states to the true values while there is a slight difference early on in the state estimation. The results presented in Table 3.1 show that the predictive performance of both the methods using MSE and LL values have negligible discrepancies. The LL values using the CKF are slightly higher than the GMA owing to the difference in the state estimation at the initial stage, which disappears as the state estimates merge together and move towards the true values.

Table 3.1 Comparison of the mean square error and log-likelihood estimates for the GMA and the CKF

metric	MSE		Log-likelihood, $\sum_{t=1}^{T} \ln f(\boldsymbol{x}_t \vert y_{1:t})$		
	$x^{\mathtt{AR}}$	x^{ϕ}	$x^{\mathtt{AR}}$	x^{ϕ}	$(x^{\mathtt{AR}}, x^{\phi})$
GMA	3.3e-03 $\pm$ 1.9e-04	**2.9e-02 $\pm$ 1.6e-02**	1181.3 $\pm$ 34.2	2081.9 $\pm$ 250.2	3281.4 $\pm$ 263.2
CKF	3.3e-03 $\pm$ 1.6e-04	3.3e-02 $\pm$ 1.2e-02	**1199.5 $\pm$ 30.15**	**2175.3 $\pm$ 198.8**	**3386.4 $\pm$ 201.8**

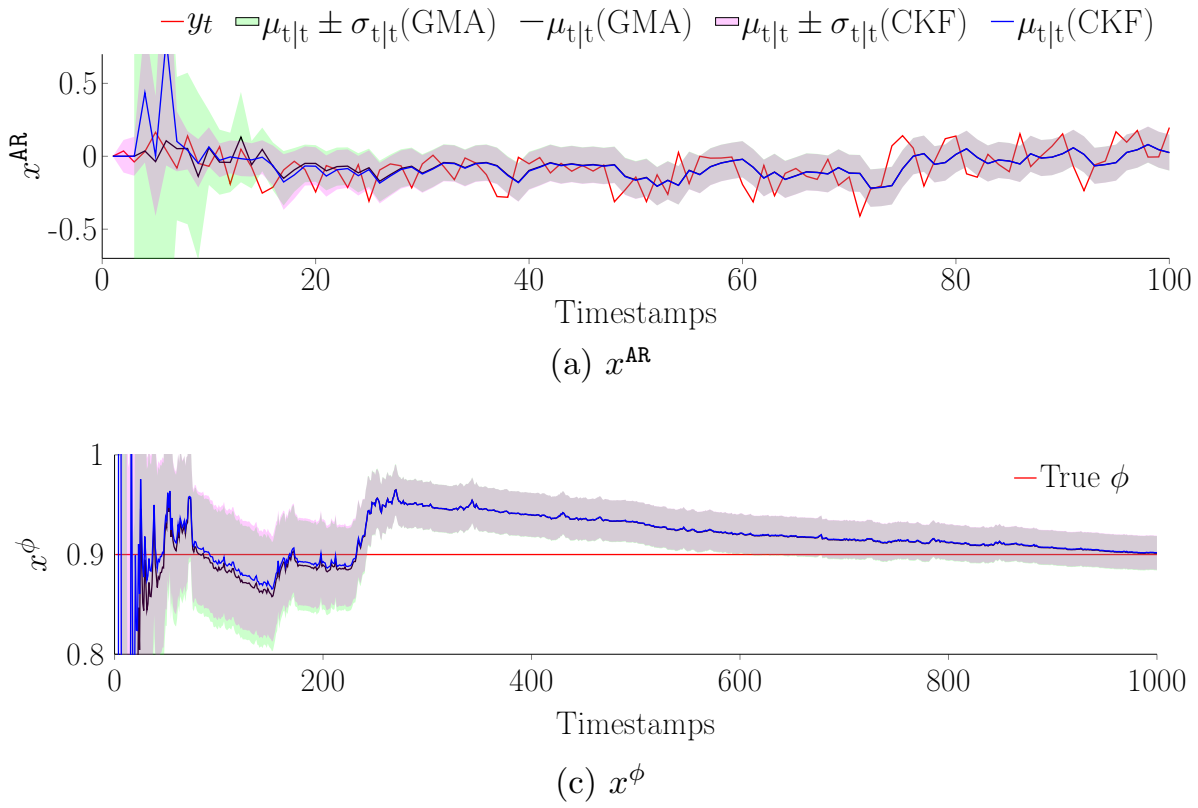


Figure 3.1 Comparison of the GMA and the CKF method for estimating a) $x^{\mathtt{AR}}$ and b) x^{ϕ}. The red solid line shows the observations, the black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions using the GMA, and the blue solid line and the pink shaded region shows the predictions and their $\pm 1\sigma$ confidence regions using the CKF. Note that Figure (a) is a close-up view from the actual plot showing the first 100 time steps.

3.3.2 Case Study 2: Trend Multiplicative Model (TM)

This case study is conducted on water infiltration flow-rate data [100–102] recorded on a concrete gravity dam in Canada. Such data is employed by engineers as a proxy for a dam's health.

Data Description

The flow-rate data ranges from September 26^{th} 2006 to December 31^{st} 2012. The raw data is averaged daily to have 2289 data points. The data have an increasing baseline along with a periodic component whose amplitude is increasing with time. A *multiplicative model* is the classical approach [103] to handle periodicity that varies with time, which can be performed

by the product of the baseline component with the static periodic component. The data are divided into a *training set* (1618 points) and a *test set* (671 points) to evaluate the predictive performance of the model. Figure 3.2 shows the entire dataset where the test set is represented by the shaded region.

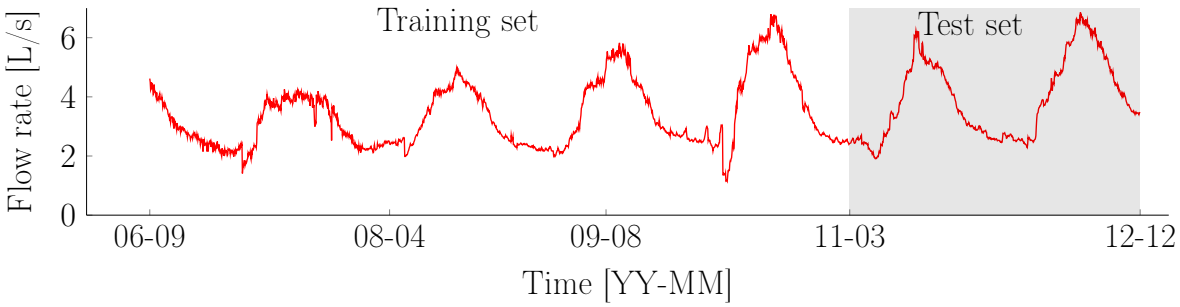


Figure 3.2 Plot showing the flow-rate data recorded on a concrete gravity dam. The test set is represented by the shaded region.

Model Formulation

The components used for this model are a *local trend* (`LT`) that includes the *level* (`L`) and the *local trend* (`LT`) hidden states to model the baseline of the time series, a *periodic component* (`S`) that includes the amplitude ($\mathtt{S}_1$) to model the periodic pattern with a periodicity of one year, an online first-order autoregressive component (`OAR`) to model the residual, and a new component called *trend multiplicative* (`TM`) to model the increasing amplitude with time. The `TM` component includes a new set of level (`LP`) and local trend (`TP`) hidden states to capture the constant rate of change in the level of the periodic component. In this case, the transition model has two placeholders for the two product terms represented by $x_{t-1}^{\mathtt{P}_1} = \{x_{t-1}^{\mathtt{AR}} \cdot x_{t-1}^{\phi}\}$ for the `OAR` and $x_t^{\mathtt{P}_2} = \{x_t^{\mathtt{LP}} \cdot x_t^{\mathtt{S}_1}\}$ for the `TM`. The transition model for the new component `TM` is given by concatenating the product term $x^{\mathtt{P}_2}$ to model the time-varying amplitude, with the local trend component provided by

$$\begin{bmatrix} x^{\mathtt{LP}} \\ x^{\mathtt{TP}} \\ x^{\mathtt{P}_2} \end{bmatrix}_t = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x^{\mathtt{LP}} \\ x^{\mathtt{TP}} \\ x^{\mathtt{P}_2} \end{bmatrix}_{t-1}.$$

The vector of 10 hidden states at time $t-1$ for all the components combined together is defined as

$$\boldsymbol{x}_{t-1} = [\overbrace{\underbrace{x^{\mathtt{L}}\ x^{\mathtt{LT}}\ x^{\mathtt{S}_1}\ x^{\mathtt{S}_2}\ x^{\mathtt{AR}}\ x^{\phi}}_{\boldsymbol{x}^{\mathtt{S}_3}}\ x^{\mathtt{P}_1}}^{\boldsymbol{x}^{\mathtt{S}_1}}\ \overbrace{x^{\mathtt{LP}}\ x^{\mathtt{TP}}\ x^{\mathtt{P}_2}}^{\boldsymbol{x}^{\mathtt{S}_2}}]^{\intercal}_{t-1},$$

where $\boldsymbol{x}^{\mathtt{S}_1}_{t-1}$, $\boldsymbol{x}^{\mathtt{S}_2}_{t-1}$ and $\boldsymbol{x}^{\mathtt{S}_3}_{t-1}$ are subsets of the vector $\boldsymbol{x}_{t-1}$. The prediction step in the Kalman filter is carried out sequentially using the mean vector $\boldsymbol{\mu}_{t-1|t-1}$ and the covariance matrix $\boldsymbol{\Sigma}_{t-1|t-1}$ of $\boldsymbol{X}_{t-1}$ given by

1. $\tilde{\boldsymbol{\mu}}_{1,t-1|t-1} = \boldsymbol{\mu}_{t-1|t-1} + \begin{bmatrix} \mathbf{0}_{n_1\times 1} \\ \mathbb{E}[X^{\mathtt{P}_1}] \\ \mathbf{0}_{n_2\times 1} \end{bmatrix}_{t-1|t-1},$

   $\tilde{\boldsymbol{\Sigma}}_{1,t-1|t-1} = \boldsymbol{\Sigma}_{t-1|t-1} + \begin{bmatrix} \mathbf{0}_{n\times n_1} & \mathrm{cov}(\boldsymbol{X}, X^{\mathtt{P}_1}) & \mathbf{0}_{n\times n_2} \end{bmatrix}_{t-1|t-1},$

2. $\tilde{\boldsymbol{\mu}}_{1,t|t-1} = \mathbf{A}\tilde{\boldsymbol{\mu}}_{1,t-1|t-1},$

   $\tilde{\boldsymbol{\Sigma}}_{1,t|t-1} = \mathbf{A}\tilde{\boldsymbol{\Sigma}}_{1,t-1|t-1}\mathbf{A}^{\intercal} + \mathbf{Q},$

3. $\tilde{\boldsymbol{\mu}}_{t|t-1} = \tilde{\boldsymbol{\mu}}_{1,t|t-1} + \begin{bmatrix} \mathbf{0}_{n_3\times 1} \\ \mathbb{E}[X^{\mathtt{P}_2}] \end{bmatrix}_{t|t-1},$

   $\tilde{\boldsymbol{\Sigma}}_{t|t-1} = \tilde{\boldsymbol{\Sigma}}_{1,t|t-1} + \begin{bmatrix} \mathbf{0}_{n_3\times n_3} & \mathrm{cov}(\boldsymbol{X}^{\mathtt{S}_3}, X^{\mathtt{P}_2}) \\ \mathrm{cov}(X^{\mathtt{P}_2}, \boldsymbol{X}^{\mathtt{S}_3}) & \mathrm{var}(X^{\mathtt{P}_2}) \end{bmatrix}_{t|t-1},$

where in Step 1 we explicitly compute the expected value $\mathbb{E}[X^{\mathtt{P}_1}]$ and the covariance matrix $\mathrm{cov}(\boldsymbol{X}, X^{\mathtt{P}_1})$ associated with the first product term $x^{\mathtt{P}_1}_{t-1}$, thereby computing the augmented mean vector $\tilde{\boldsymbol{\mu}}_{1,t-1|t-1}$ and the covariance matrix $\tilde{\boldsymbol{\Sigma}}_{1,t-1|t-1}$, in Step 2 we carry out the prediction step using $\tilde{\boldsymbol{\mu}}_{1,t-1|t-1}$, $\tilde{\boldsymbol{\Sigma}}_{1,t-1|t-1}$, and the matrices $\mathbf{A}$ and $\mathbf{Q}$, and finally in Step 3, we compute the moments for the second product term $x^{\mathtt{P}_2}_t$ to obtain the predicted mean vector $\tilde{\boldsymbol{\mu}}_{t|t-1}$ and the covariance matrix $\tilde{\boldsymbol{\Sigma}}_{t|t-1}$. Using $\tilde{\boldsymbol{\mu}}_{t|t-1}$, $\tilde{\boldsymbol{\Sigma}}_{t|t-1}$, and Equation 3.12, the update step of the Kalman filter is performed. The complete model matrices $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$ and $\mathbf{R}$ required for the state estimation using the Kalman filter are described in Appendix A. The vector of unknown parameters which need to be estimated using an optimization algorithm [18,25,104] are given by

$$\boldsymbol{\theta} = [\sigma^{\mathtt{LT}}_W\ \sigma^{\mathtt{AR}}_W\ \sigma^{\mathtt{TP}}_W\ \sigma_V]^{\intercal},$$

where $\sigma^{\mathtt{LT}}_W$ is the standard deviation of the local trend, $\sigma^{\mathtt{AR}}_W$ is the standard deviation of the $\mathtt{AR}$ process, $\sigma^{\mathtt{TP}}_W$ is the standard deviation of the local trend ($\mathtt{TP}$) in the $\mathtt{TM}$, and σ_V is the standard

deviation for the observation error. Using the initial parameters $\boldsymbol{\theta}_0 = [10^{-6} \quad 0.1 \quad 10^{-6} \quad 1]^\intercal$, the optimized parameters $\boldsymbol{\theta}^* = [2.16 \times 10^{-6} \quad 0.092 \quad 6.5 \times 10^{-7} \quad 0.054]^\intercal$ are obtained by maximizing the joint log-likelihood [21] using the Newton-Raphson method [25].

State Estimation

Figure 3.3 shows the observed flow-rate data in red, the black solid line and the green shaded region shows the predictions $\boldsymbol{\mu}_{t|t}$ and the uncertainty bounds $\boldsymbol{\mu}_{t|t} \pm \boldsymbol{\sigma}_{t|t}$ using the GMA, and in blue solid line and pink shaded region using the CKF, for both the training set and the test set. The grey region shows the forecast period. Figure 3.4 shows the hidden state estimation of the flow-rate data; where (a) represents the product of the level associated with the `TM` and the periodic pattern $x^{\mathtt{S}_1}$, (b) represents the level component $x^{\mathtt{LP}}$ associated with the `TM` component, and (c) represents the periodic pattern $x^{\mathtt{S}_1}$. Figure 3.4(d) represents the online estimation of x^{ϕ} associated with the `OAR`. The black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions.The test set mean square error (MSE) and log-likelihood (LL) values obtained using the CKF and the GMA are $\{0.28, -541.5\}$ and $\{0.28, -541.6\}$ respectively. These results show that the proposed methodology has the same predictive capacity as that of the CKF and also provide interpretable sub-components of the time-varying amplitude hidden state.

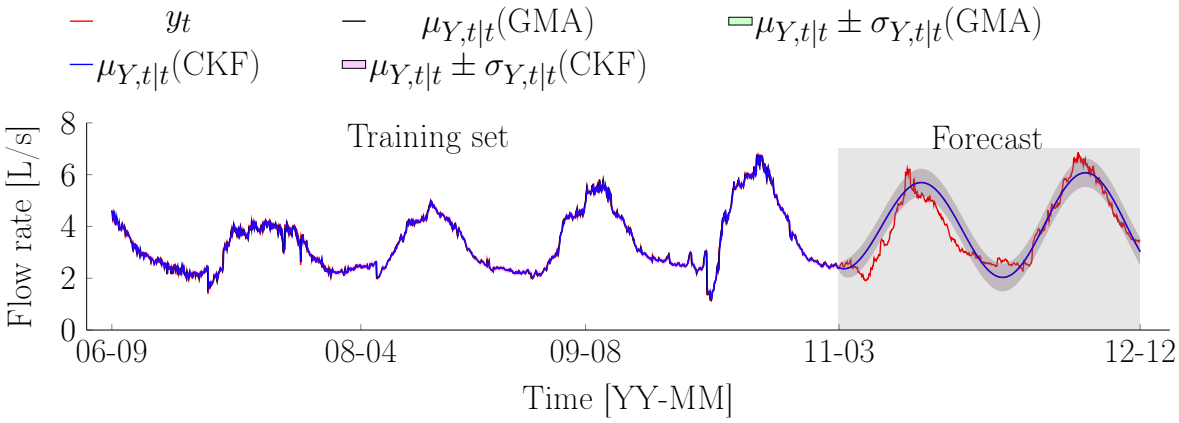


Figure 3.3 Plot showing the estimated values for the flow-rate data using the GMA and the CKF. The red solid line shows the observations, the black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions using the GMA, and the blue solid line and the pink shaded region shows the predictions and their $\pm 1\sigma$ confidence regions using the CKF.

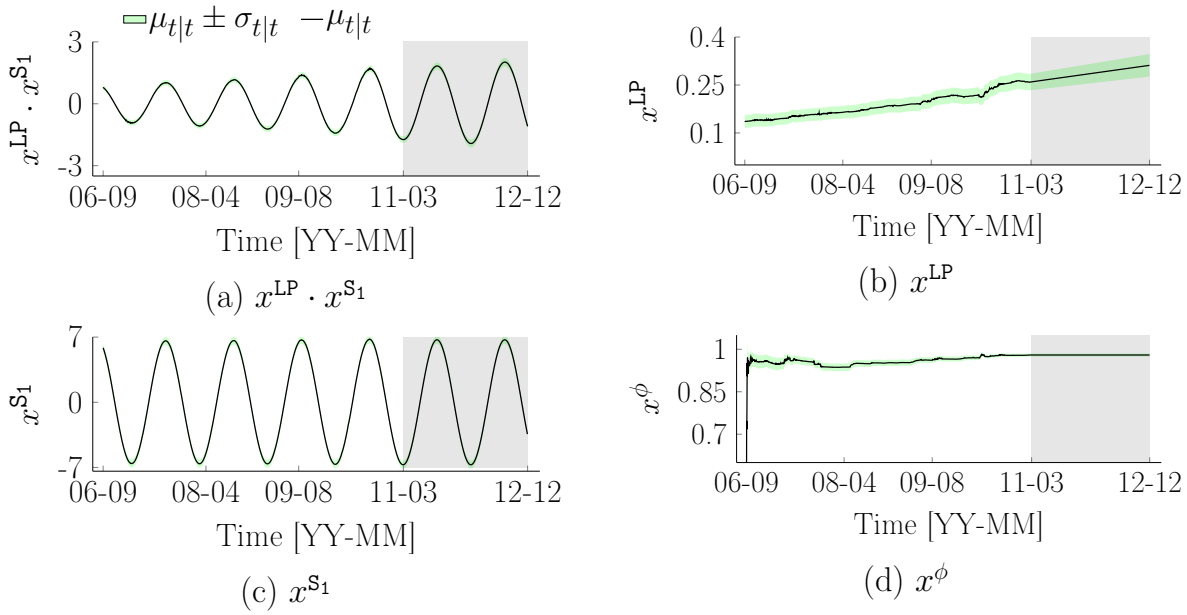


Figure 3.4 Illustration of the hidden state estimation for the flow-rate data. Figures (a)-(c) represents the hidden states of the `TM` component; where (a) represents the product of the level associated with the `TM` and the periodic pattern $x^{\mathtt{S}_1}$, (b) represents the level component $x^{\mathtt{LP}}$ associated with the `TM` component, and (c) represents the periodic pattern $x^{\mathtt{S}_1}$. Figure (d) represents the online estimation of x^{ϕ} associated with the `OAR`. The black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions.

3.3.3 Case Study 3: Double Kernel Regression (`DKR`)

This case study is conducted on traffic-load data [29, 105, 106] recorded on the Tamar bridge in the UK. In the context of structural health monitoring, modeling traffic data correctly is important for removing its effect on structural responses.

Data Description

The data ranges from September 01 to October 21, 2007. The raw data have 2409 data points with a uniform time steps of 30 minutes. The raw data shows a constant baseline and two periodic components having a daily and a weekly periodicity. A multiplicative model is used to capture the dual periodicity using the product of the two periodic components. The data is divided into a *training set* (1649 points) and a *test set* (760). The entire dataset is shown in Figure 3.5.

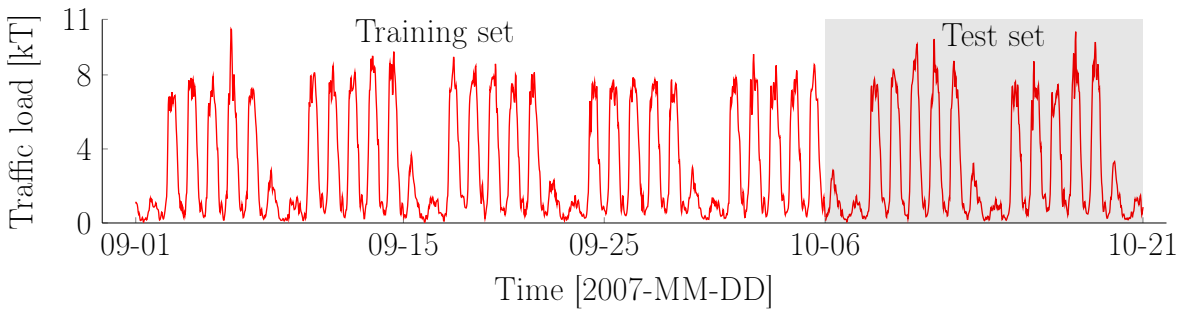


Figure 3.5 Plot showing traffic-load data recorded on the Tamar bridge in the UK. The test set is represented by the shaded region.

Model Formulation

The BDLM components used are the *local level* (LL) to model the constant baseline, two *kernel regression* (KR) components [29] each having 50 non-uniform and 30 uniform control-points to model the periodic patterns with periodicity of 7 days and 1 day respectively, the online autoregressive component (OAR), and a new component called the *double kernel regression* (DKR). The DKR is used to model the product of two periodic patterns represented by the hidden states $x_0^{\mathtt{KR}_1}$ and $x_0^{\mathtt{KR}_2}$. The KR component for modeling the 7 day periodic pattern requires more control points in the first two days due to a higher complexity in the sub-daily pattern compared to the rest of the week. Note that increasing the number of control points can further improve accuracy at the cost of increasing computational cost. In this case, the transition model has two product terms represented by $x_{t-1}^{\mathtt{p}_1} = \{x_{t-1}^{\mathtt{AR}} \cdot x_{t-1}^{\phi}\}$ for the OAR and $x_t^{\mathtt{p}_2} = \{x_0^{\mathtt{KR}_1} \cdot x_0^{\mathtt{KR}_2}\}_t$ for the DKR. The vector of hidden states at time $t-1$ for all the components is defined as

$$\boldsymbol{x}_{t-1} = [\overbrace{\underbrace{x^{\mathtt{LL}}\ x^{\mathtt{AR}}\ x^{\phi}}_{\boldsymbol{x}^{\mathtt{s}_1}}\ x^{\mathtt{p}_1}\ \overbrace{x_0^{\mathtt{KR}_1} \ldots x_{50}^{\mathtt{KR}_1}\ x_0^{\mathtt{KR}_2} \ldots x_{30}^{\mathtt{KR}_2}}^{\boldsymbol{x}^{\mathtt{s}_2}}\ x^{\mathtt{p}_2}}^{\boldsymbol{x}^{\mathtt{s}_3}}]_{t-1}^{\intercal}. \tag{3.14}$$

The prediction step in the Kalman filter is carried out sequentially using the mean vector $\boldsymbol{\mu}_{t-1|t-1}$ and the covariance matrix $\boldsymbol{\Sigma}_{t-1|t-1}$ of $\boldsymbol{X}_{t-1}$ as shown in Section 3.3.2. The complete model matrices $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$ and $\mathbf{R}$ are described in Appendix A. The vector of unknown parameters is given by

$$\boldsymbol{\theta} = [\sigma_W^{\mathtt{LL}}\ \ell^{\mathtt{KR}_1}\ \ell^{\mathtt{KR}_2}\ \sigma_W^{\mathtt{AR}}\ \sigma_V]^{\intercal},$$

where $\sigma_W^{\mathtt{LL}}$ is the standard deviation of the local level, $\ell^{\mathtt{KR}_1}$ is the kernel length for the KR component with a period of 1 day, $\ell^{\mathtt{KR}_2}$ is the kernel length for the KR component with a period

of 7 days, $\sigma_W^{\texttt{AR}}$ is the standard deviation of the AR process, and σ_V is the standard deviation for the observation error. Using the initial parameter values $\boldsymbol{\theta}_0 = [10^{-6} \;\; 0.05 \;\; 0.5 \; 0.1 \;\; 0.1]^\intercal$, the optimized values $\boldsymbol{\theta}^* = [1.01 \times 10^{-6} \;\; 0.359 \;\; 0.24 \;\; 0.275 \;\; 1.93 \times 10^{-7}]^\intercal$ are obtained using the Newton-Raphson method [25].

State Estimation

Figure 3.6 shows the observed traffic-load data in red, the black solid line and the green shaded region shows the predictions $\boldsymbol{\mu}_{t|t}$ and the uncertainty bounds $\boldsymbol{\mu}_{t|t} \pm \boldsymbol{\sigma}_{t|t}$ using the GMA, and in blue solid line and pink shaded region using the CKF, for both the training set and the test set. Figure 3.7 presents the hidden state estimation for the traffic-load data; where (a) represents the product of the two product terms $x_0^{\texttt{KR}_1} \cdot x_0^{\texttt{KR}_2}$, (b) represents the periodic pattern $x_0^{\texttt{KR}_1}$ with a 7 day periodicity, and (c) represents the periodic pattern $x_0^{\texttt{KR}_2}$ with a 1 day periodicity. Figure 3.7(d) represents the online estimation of x^ϕ associated with the OAR. The black solid line and the green shaded region show the predictions and their $\pm 1\sigma$ confidence regions. These results show that using the GMA in BDLM has better predictive capacity than the CKF. The predictive capacity is also compared to the results presented in Nguyen et al. [29] for the same dataset while using a single KR component with 101 control-points having a periodicity of 7 days. Table 3.2 presents the test set mean square error and the log-likelihood values as well as the training time using the DKR and the KR. The results demonstrate that DKR has better predictive capacity than the KR component, fewer

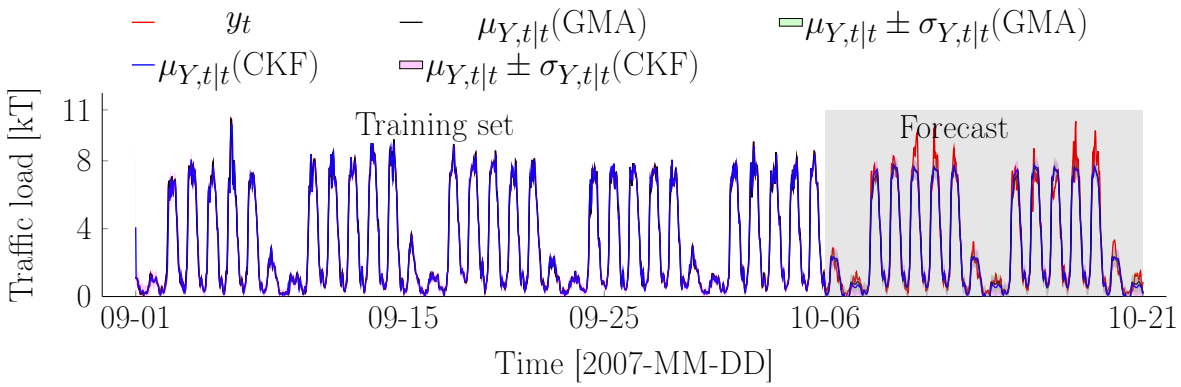


Figure 3.6 Plot showing the estimated values of traffic-load data using the GMA and the CKF. The red solid line shows the observations, the black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions using the GMA, and the blue solid line and the pink shaded region shows the predictions and their $\pm 1\sigma$ confidence regions using the CKF.

hidden states, and also has fewer parameters to optimize, which makes it computationally faster.

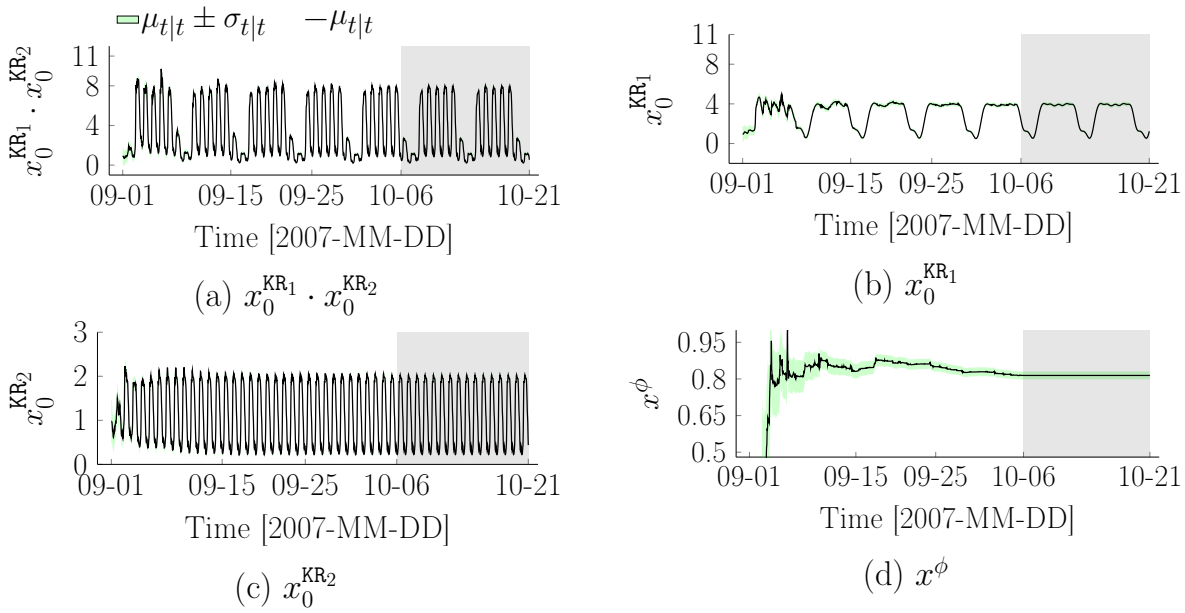


Figure 3.7 Illustration of the hidden state estimation for the traffic-load data. Figures (a)-(c) represents the hidden states of the `DKR` component; where (a) represents the product of the two product terms $x_0^{\mathtt{KR}_1} \cdot x_0^{\mathtt{KR}_2}$, (b) represents the periodic pattern $x_0^{\mathtt{KR}_1}$ with a 7 day periodicity, and (c) represents the periodic pattern $x_0^{\mathtt{KR}_2}$ with a 1 day periodicity. Figure (d) represents the online estimation of x^{ϕ} associated with the `OAR`. The black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions.

Table 3.2 Comparison of mean square error and log-likelihood values for `DKR` and `KR` on the traffic-load dataset.

metric	MSE	Log-likelihood, $\sum_{t=1}^{T} \ln f(\boldsymbol{x}_t \vert y_{1:t})$
`DKR`	**0.30**	**−616.96**
`KR`	0.34	−656.30
`CKF`	0.32	−629.59

3.4 Conclusion

This chapter presented an analytical approach to handle multiplicative state-space models by leveraging the Gaussian multiplicative approximation (GMA). The method enables: (1) the analytical inference of the mean vector and the covariance matrix for the product of two hidden states in the transition and/or observation models using linear estimation theory, and (2) analytically tractable online estimation of model parameters as hidden states. In the first case study, the proposed method was validated and verified using synthetic data for a first-order autoregressive process. A new generic component developed was called the online autoregressive (`OAR`) to evaluate both the `AR` state and the `AR` parameter together. For the second and third case studies, the proposed method was applied to SHM-based real data for which new components were developed, namely the trend multiplicative (`TM`) and the double kernel regression (`DKR`), in order to handle specific multiplicative models. The three case studies confirm that the method match or exceeds the performance of the existing nonlinear Kalman filter methods such as the cubature Kalman filter in terms of both predictive capacity and computational complexity.

CHAPTER 4 Modeling Nonlinear Dependency Using State-Based Regression

4.1 Introduction

Bayesian dynamic linear models (BDLM) [18] are probabilistic approaches used for time series analysis and that are capable of online learning (see Section 2.2.1). These models consist of generic components that each capture a specific pattern and that can be grouped together to model time series. With the existing BDLM framework, linear relationships between independent and the dependent time series can be described by a constant regression coefficient [21]. Nevertheless, with the existing method, it is not possible to model a nonlinear relationship between two time series in such a way that the regression coefficient is not a constant parameter but a function of the current value of the independent time series. This is particularly important in the context of dam health monitoring where the displacement at various locations is most often nonlinearly related to the reservoir water level [17]. Building upon the Gaussian multiplicative approximation described in Section 3.2, this chapter presents the *state-based regression* (`SR`) method to model the nonlinear dependency between any two time series such that the state-dependent regression coefficient is inferred using a new `SR` component within the BDLM framework. This component provides an interpretable representation of how each nonlinear dependency explains specific patterns in the interdependent time series. In addition, the chapter also describes the methodology for performing state estimation using the `SR` method along with two case studies derived from the 16^{th} *International Commission on Large Dams* (ICOLD) Benchmark 2022 on dam behavior prediction [107]. The main contributions of this chapter are to

- Provide the methodology for applying the state-based regression method to model the nonlinear dependency between any two time series.
- Validate the method using two inverted pendulum datasets from the ICOLD workshop.

4.2 Methodology

This section provides the methodology for applying the state-based regression method to model nonlinear dependency between two time series. First, a kernel method is employed to obtain the probabilistic weights for a set of regression coefficients associated with the value of an independent time series at a particular time instant. Thereafter, a weighted summation of these regression coefficients is carried out in order to obtain the predicted regression

coefficient using which the dependent value is obtained by multiplying the coefficient with the independent value. Second, the SR component is created within the BDLM framework that allows the regression coefficient and the dependent time series to be learned probabilistically while maintaining analytical tractability.

4.2.1 Kernel Method

The state-based regression is a kernel method relying on a set of control-points, where each consists of a *reference variable* $x^{\texttt{cp}}$ that is associated with a *regression coefficient* represented by the hidden state $x^{\phi^{\texttt{R}}}$. The reference variables are defined as a fixed set of values covering the entire output range of the *independent time series*. The hidden state associated with each control-point is modeled as a Gaussian random variable such that $X^{\phi^{\texttt{R}}} \sim \mathcal{N}(x^{\phi^{\texttt{R}}}; \mu^{\phi^{\texttt{R}}}, (\sigma^{\phi^{\texttt{R}}})^2)$, where $\mu^{\phi^{\texttt{R}}}$ is the expected value and $(\sigma^{\phi^{\texttt{R}}})^2$ is the variance. Figure 4.1 shows the schematic plot for a set of five control-points, $(x^{\texttt{cp}}, x^{\phi^{\texttt{R}}})_i, \forall i = 1:5$, where each point is associated with a value for $x^{\texttt{cp}}$ as well as the expected value $\mu^{\phi^{\texttt{R}}}$and uncertainty bound given by $\mu^{\phi^{\texttt{R}}} \pm \sigma^{\phi^{\texttt{R}}}$.

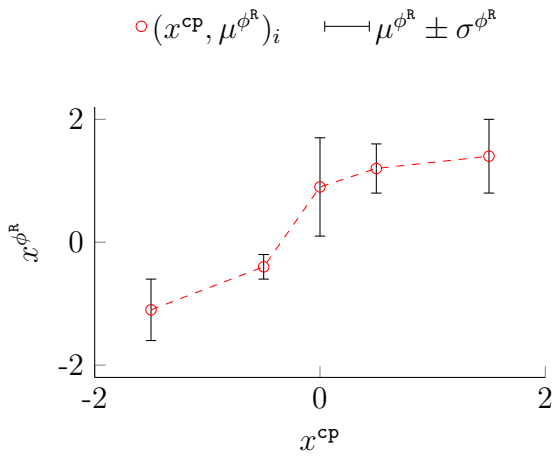


Figure 4.1 Illustration of the set of control-points where each point marked in red circle $(x^{\texttt{cp}}, \mu^{\phi^{\texttt{R}}})_i$ is associated with a value for the reference variable $x^{\texttt{cp}}$ as well as the expected value of the hidden state $\mu^{\phi^{\texttt{R}}}$. The uncertainty bounds for the hidden state $\mu^{\phi^{\texttt{R}}} \pm \sigma^{\phi^{\texttt{R}}}$ are shown by the black error bars.

The Gaussian radial basis function (RBF) kernel [25] is used to measure the similarity between the hidden state associated with the independent time series $x^{\texttt{ref}}$ and the reference variable $\boldsymbol{x}^{\texttt{cp}}$ given by

$$k(x^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}}) = \exp\left[\frac{-1}{2(\ell^{\texttt{SK}})^2}(x^{\texttt{ref}} - \boldsymbol{x}^{\texttt{cp}})^2\right], \tag{4.1}$$

where $k(x^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}})$ gives the kernel outputs as a function of the Euclidean distance between the two covariates $x^{\texttt{ref}}$ and $\boldsymbol{x}^{\texttt{cp}}$, and the kernel length $\ell^{\texttt{SK}}$. Figure 4.2 shows an illustrative example of the kernel outputs $k(x^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}})$ obtained using the independent time series $x^{\texttt{ref}}$ marked in solid blue line and the set of control-points $\boldsymbol{x}^{\texttt{cp}}$ marked in red circles that cover the entire output range of $x^{\texttt{ref}}$, i.e., $[-1.5, 1.5]$. In Figure 4.2, $x_t^{\texttt{ref}}$ is the value of the independent time series at the time instant t denoted by the black asterisk. Following Equation 4.1, the kernel outputs at time t which are marked by red crosses are obtained using $x_t^{\texttt{ref}}$ and the control-points $\boldsymbol{x}^{\texttt{cp}}$.

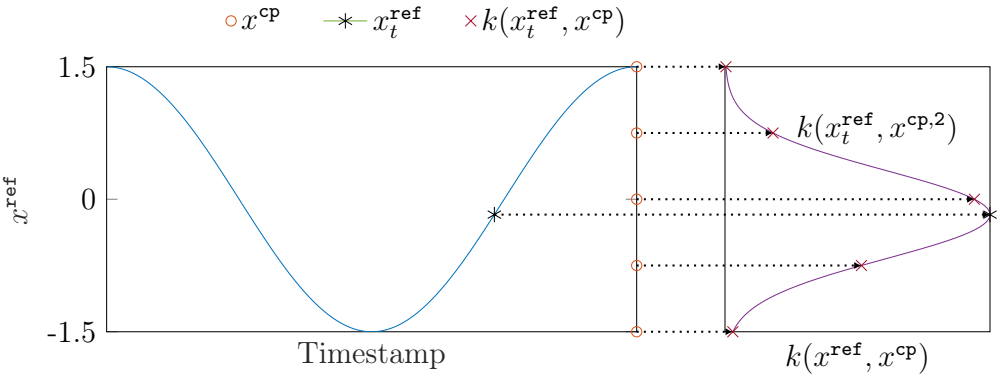


Figure 4.2 Illustrative example showing the process for obtaining the kernel outputs using the independent time series $x^{\texttt{ref}}$ and the set of control-points $x^{\texttt{cp}}$ at a given instant of time t. The control-points are marked by red circles that cover the entire output range of $x^{\texttt{ref}}$, i.e., $[-1.5, 1.5]$, the independent time series $x^{\texttt{ref}}$ is marked in solid blue line, the value of the independent time series $x_t^{\texttt{ref}}$ at time t is marked by the black asterisk, and the kernel outputs $k(x^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}})$ at time t are denoted by red crosses. The Gaussian radial basis function (RBF) is represented by the purple solid line as defined in Equation 4.1.

Furthermore, these kernel outputs are normalized in order to obtain probabilistic weights for each hidden state $x^{\phi^{\texttt{R}}}$. Thereafter, the hidden state associated with the state-dependent regression coefficient $x_0^{\phi^{\texttt{R}}}$ is computed using a weighted summation of the normalized kernel outputs $\tilde{k}(x^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}})$ and the control-point's hidden states $\boldsymbol{x}^{\phi^{\texttt{R}}}$. Finally, the hidden state associated with the interdependent time series $x^{\texttt{D}}$ is obtained by multiplying the reference variable $x^{\texttt{ref}}$ with the state-dependent regression coefficient $x_0^{\phi^{\texttt{R}}}$ such that $x^{\texttt{D}} = x_0^{\phi^{\texttt{R}}} \cdot x^{\texttt{ref}}$.

Given that the input $X^{\texttt{ref}} \sim \mathcal{N}(x^{\texttt{ref}}; \mu^{\texttt{ref}}, (\sigma^{\texttt{ref}})^2)$ is a Gaussian random variable due to the fact that it is a hidden state inferred from the observations, the output of $k(x^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}})$ from Equation 4.1 is also a random variable. For maintaining the analytical tractability, the kernel function is linearised using a 1^{st} order Taylor series expansion at $\mu^{\texttt{ref}}$ such that

$$k(x^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}}) \approx \boldsymbol{x}^{\texttt{SK}} = k(\mu^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}}) + k'(\mu^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}})(x^{\texttt{ref}} - \mu^{\texttt{ref}}), \tag{4.2}$$

where the random kernel outputs $k(x^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}})$ are represented by the hidden state vector $\boldsymbol{x}^{\texttt{SK}}$, $k(\mu^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}})$, and $k'(\mu^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}})$ represents the kernel output and the first-order derivative at $\mu^{\texttt{ref}}$, where $\mu^{\texttt{ref}} \equiv \mathbb{E}[X^{\texttt{ref}}]$. Using Equation 4.2, the raw and the normalized expected kernel outputs are obtained by

$$\begin{aligned} \boldsymbol{\mu}^{\texttt{SK}} &= \exp\left[\frac{-1}{2(\ell^{\texttt{SK}})^2}(\mu^{\texttt{ref}} - \boldsymbol{x}^{\texttt{cp}})^2\right], \\ \tilde{\boldsymbol{\mu}}^{\texttt{SK}} &= \frac{\boldsymbol{\mu}^{\texttt{SK}}}{\sum \boldsymbol{\mu}^{\texttt{SK}}}, \end{aligned} \tag{4.3}$$

where $\boldsymbol{\mu}^{\texttt{SK}}$ and $\tilde{\boldsymbol{\mu}}^{\texttt{SK}}$ represents the raw and the normalized expected kernel outputs for the hidden state vector $\boldsymbol{x}^{\texttt{SK}}$ and $\tilde{\boldsymbol{x}}^{\texttt{SK}}$. The normalized 1^{st} order derivative of $x^{\texttt{SK}}$ evaluated at $\mu^{\texttt{ref}}$ is given by

$$\tilde{k'}(\mu^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}}) = \exp\left[\frac{-1}{2(\ell^{\texttt{SK}})^2}(\mu^{\texttt{ref}} - \boldsymbol{x}^{\texttt{cp}})^2\right] \cdot \frac{-(\mu^{\texttt{ref}} - \boldsymbol{x}^{\texttt{cp}})}{(\ell^{\texttt{SK}})^2} \cdot \frac{\boldsymbol{\mu}^{\texttt{SK}}}{\tilde{\boldsymbol{\mu}}^{\texttt{SK}}}. \tag{4.4}$$

Using Equations 4.2-4.4 and the variance of $X^{\texttt{ref}}$, the variance and covariance terms in the covariance matrix of $\tilde{\boldsymbol{X}}^{\texttt{SK}}$ are given by

$$\begin{aligned} \text{var}(\tilde{\boldsymbol{X}}_i^{\texttt{SK}}) &= (\tilde{k'}(\mu^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}}))^2 \cdot (\sigma^{\texttt{ref}})^2, \\ \text{cov}(\tilde{\boldsymbol{X}}_i^{\texttt{SK}}, \tilde{\boldsymbol{X}}_j^{\texttt{SK}}) &= \tilde{k'}(\mu_i^{\texttt{ref}}, x_i^{\texttt{cp}}) \cdot \tilde{k'}(\mu_j^{\texttt{ref}}, x_j^{\texttt{cp}}) \cdot (\sigma^{\texttt{ref}})^2. \end{aligned}$$

The cross-covariance between $\tilde{\boldsymbol{X}}^{\texttt{SK}}$ and any other Gaussian random variable X is given by

$$\text{cov}(X, \tilde{\boldsymbol{X}}^{\texttt{SK}}) = \tilde{k'}(\mu^{\texttt{ref}}, \boldsymbol{x}^{\texttt{cp}}) \cdot \text{cov}(X, X^{\texttt{ref}}).$$

The SR method also allows for modeling multiple pair of dependencies where more than one independent time series can be used to model a dependent time series. In such cases, a separate random variable is considered for the normalized kernel outputs associated with each independent time series. The cross-covariance between the random variables $\tilde{\boldsymbol{X}}_k^{\texttt{SK}}$ and $\tilde{\boldsymbol{X}}_l^{\texttt{SK}}$ for any k^{th} and l^{th} time series is given by

$$\text{cov}(\tilde{\boldsymbol{X}}_{k,i}^{\texttt{SK}}, \tilde{\boldsymbol{X}}_{l,j}^{\texttt{SK}}) = \tilde{k'}(\mu_{k,i}^{\texttt{ref}}, x_{k,i}^{\texttt{cp}}) \cdot \tilde{k'}(\mu_{l,j}^{\texttt{ref}}, x_{l,j}^{\texttt{cp}}) \cdot \text{cov}(X_k^{\texttt{ref}}, X_l^{\texttt{ref}}).$$

where $\{\mu_{k,i}^{\texttt{ref}}, x_{k,i}^{\texttt{cp}}\}$ and $\{\mu_{l,i}^{\texttt{ref}}, x_{l,i}^{\texttt{cp}}\}$ represents the expected values and the fixed set of points for the two independent hidden states $X_k^{\texttt{ref}}$ and $X_l^{\texttt{ref}}$, and $\text{cov}(X_k^{\texttt{ref}}, X_l^{\texttt{ref}})$ represents the covariance between them.

4.2.2 State Regression Component

Within the BDLM framework, a new generic component called the *state regression* (SR) is created so that nonlinear relationships between time series can be modeled and can also be combined with other generic components while maintaining the analytical tractability. The SR component provides the estimated values $\mu_{t|t}$ and their uncertainty bounds $\mu_{t|t} \pm \sigma_{t|t}$ for the predicted regression coefficient $X_0^{\phi^{\mathtt{R}}}$ and the predicted pattern for the dependent time series $X^{\mathtt{D}}$. The SR component includes N hidden states for the normalized kernel outputs such that $\tilde{\boldsymbol{x}}^{\mathtt{SK}} = [\tilde{x}_1^{\mathtt{SK}} \ \tilde{x}_2^{\mathtt{SK}} \ \ldots \tilde{x}_{\mathtt{N}}^{\mathtt{SK}}]^{\intercal}$; $\mathtt{N}+1$ hidden states for the regression coefficient that includes N hidden states associated with the control-points, $\boldsymbol{x}^{\phi^{\mathtt{R}}} = [x_1^{\phi^{\mathtt{R}}} \ x_2^{\phi^{\mathtt{R}}} \ \ldots x_{\mathtt{N}}^{\phi^{\mathtt{R}}}]^{\intercal}$ and the hidden state for the predicted regression coefficient $x_0^{\phi^{\mathtt{R}}}$; the hidden state for the dependent time series, $x^{\mathtt{D}} = (x_0^{\phi^{\mathtt{R}}} \cdot x^{\mathtt{ref}})$, and N product terms, $\mathbf{x}^{\mathtt{P}} = [x^{\mathtt{P}_1} \ x^{\mathtt{P}_2} \ \ldots x^{\mathtt{P}_{\mathtt{N}}}]^{\intercal}$, where, $x^{\mathtt{P}_i} = (\tilde{x}_i^{\mathtt{SK}} \cdot x_i^{\phi^{\mathtt{R}}}); \forall i = 1:\mathtt{N}$. The GMA (see Section 3.2) is used to compute the moments associated with the product terms. The predicted regression coefficient is computed using the weighted summation of the normalized kernel outputs, $\tilde{\boldsymbol{x}}^{\mathtt{SK}}$ and the hidden states for the control-points $\boldsymbol{x}^{\phi^{\mathtt{R}}}$ so that,

$$x_0^{\phi^{\mathtt{R}}} = \tilde{x}_1^{\mathtt{SK}} x_1^{\phi^{\mathtt{R}}} + \tilde{x}_2^{\mathtt{SK}} x_2^{\phi^{\mathtt{R}}} + \ldots + \tilde{x}_{\mathtt{N}}^{\mathtt{SK}} x_{\mathtt{N}}^{\phi^{\mathtt{R}}}. \tag{4.5}$$

Finally, the hidden states for the SR component can be grouped together as

$$\boldsymbol{x}^{\mathtt{SR}} = [(\tilde{\boldsymbol{x}}^{\mathtt{SK}})^{\intercal} \ (\boldsymbol{x}^{\phi^{\mathtt{R}}})^{\intercal} \ x_0^{\phi^{\mathtt{R}}} \ x^{\mathtt{D}} \ (\boldsymbol{x}^{\mathtt{P}})^{\intercal}]^{\intercal}. \tag{4.6}$$

The transition matrix for the SR component of size $3\mathtt{N}+2$ is formulated as

$$\mathbf{A}^{\mathtt{SR}} = \begin{bmatrix} \mathbf{0}_{\mathtt{N}} & 0_{1\times\mathtt{N}} & 0 & 0 & 0_{1\times\mathtt{N}} \\ \vdots & \mathbf{I}_{\mathtt{N}} & 0 & 0 & 0_{1\times\mathtt{N}} \\ \vdots & \ldots & 0 & 0 & \mathbf{1}_{1\times\mathtt{N}} \\ \vdots & \ldots & \ldots & 0 & \mathbf{0}_{1\times\mathtt{N}} \\ sym. & \ldots & \ldots & \ldots & \mathbf{0}_{\mathtt{N}} \end{bmatrix}.$$

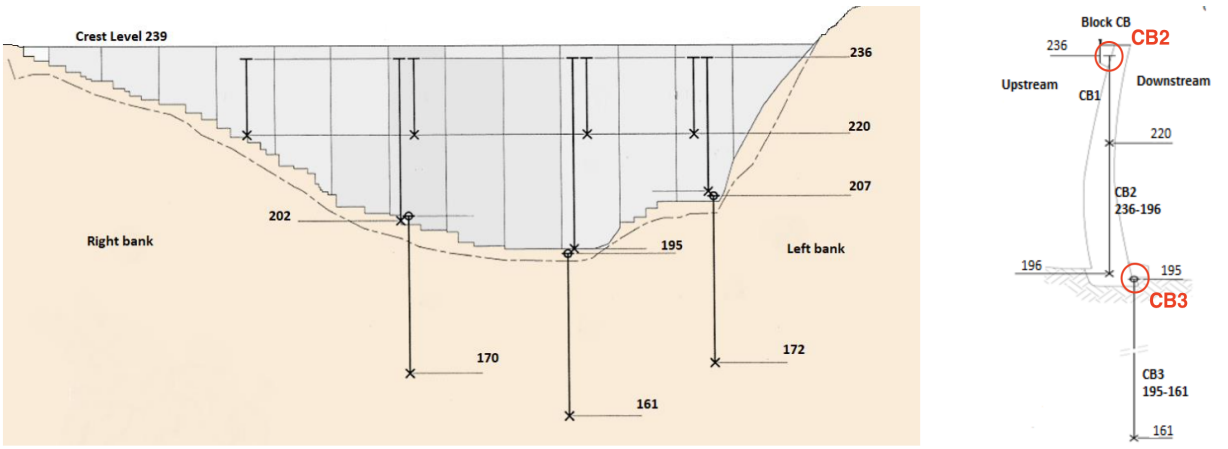
The hidden states for the control-points transition from $t-1$ to t as a random walk, i.e. $\boldsymbol{x}_t^{\phi^{\mathtt{R}}} = \boldsymbol{x}_{t-1}^{\phi^{\mathtt{R}}} + \boldsymbol{w}_t^{\phi^{\mathtt{R}}}$, as represented by $\boldsymbol{I}_{\mathtt{N}}$ in $\mathbf{A}^{\mathtt{SR}}$. Similarly to the procedure in Section 3.2.2, the kernel outputs and the product terms are merely placeholders as represented by $\mathbf{0}_{\mathtt{N}}$ to simplify the use of the method. The observation matrix $\mathbf{C}^{\mathtt{SR}}$ is given by

$$\mathbf{C}^{\mathtt{SR}} = [\mathbf{0}_{\mathtt{N}\times 1}^{\intercal} \ \mathbf{0}_{\mathtt{N}\times 1}^{\intercal} \ 0 \ 1 \ \mathbf{0}_{\mathtt{N}\times 1}^{\intercal}], \tag{4.7}$$

where the only hidden state observable is the one associated with the dependent time series i.e., $x^{\mathtt{D}}$.

4.3 Applied Examples

This section presents two case studies illustrating the application of the `SR` method for each of the two displacement datasets obtained from the 16^{th} ICOLD Benchmark 2022 [14, 107]. The data provides the radial displacement measurements in mm from inverted pendulums placed within the dam. Figure 4.3(a) shows the front view of the concrete arch dam and Figure 4.3(b) presents the elevation view showing the two inverted pendulums placed in the dam's central blocks 2 and 3 referred to as `CB2` and `CB3` [14]. In each case study, the datasets involved and the pre-processing steps are described, followed by model forecasting and interpretation.



(a) Front view (b) Elevation view

Figure 4.3 Illustration showing (a) the front view of the dam and (b) the inverted pendulums placed in the dam's central blocks 2 and 3 (`CB2` and `CB3`) that measures the dam's radial displacement reproduced from the ICOLD Benchmark [14].

4.3.1 Case Study 1 – `CB2` Time Series

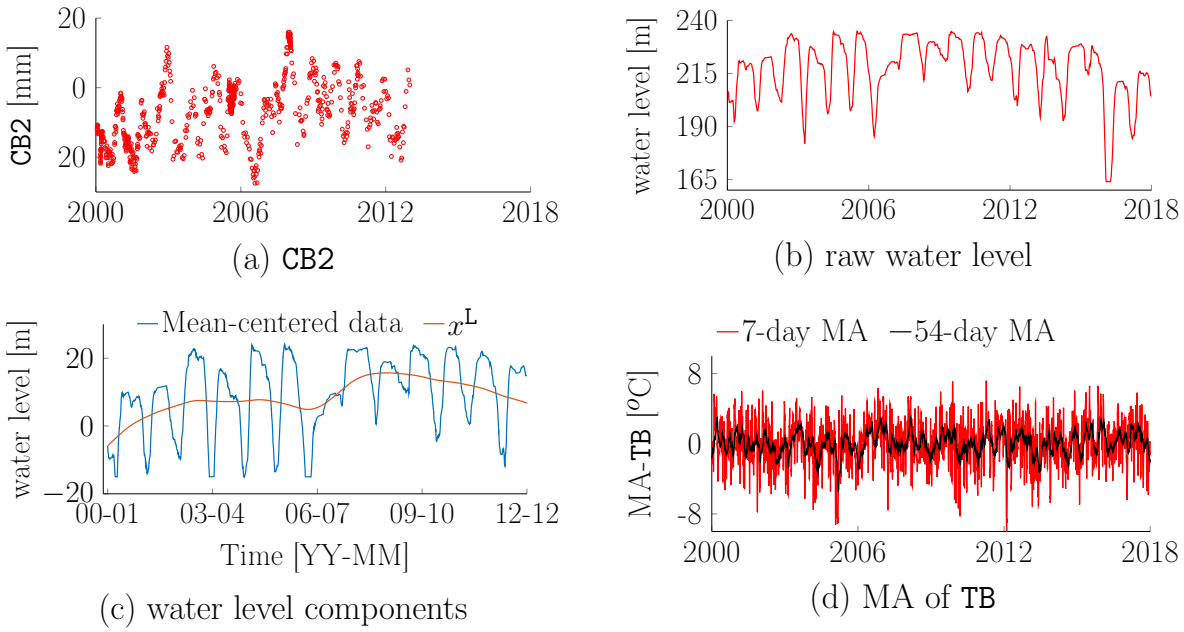
This case study is conducted on the `CB2` time series which measures the dam's radial displacement between the altitudes 236 m and 196 m, i.e., between under the crest and the toe of the dam.

Data Description

The CB2 dataset is available from the year 2000 to 2012 with an average data acquisition frequency of one data point every 1.5 week. Moreover, the daily reservoir water level and the air temperature time series are also available from the period 2000 to 2018; these two are used as *explanatory variables* to build a prediction model for CB2. The TB dataset is selected for the temperature as it takes into account the altitude of the dam and is also calculated by interpolating from several measuring stations. The stationary seasonal pattern is removed from the temperature time series and several moving averages such as $\{1, 7, 14, 28, 54\}$ days of the residuals are considered for taking into account the thermal inertia of the dam [17,108]. Furthermore, the water level time series is divided into two separate datasets to isolate the effect of the average long-term trend and its short-term periodic pattern. Figure 4.4(a) shows the raw CB2 data using red dotted points, Figure 4.4(b) presents the raw daily dataset for water level in red solid line, and Figure 4.4(c) provides the mean-centered data showing the short-term periodic pattern in blue solid line and the average long-term trend ($x^{\mathtt{L}}$) in red solid line. Figure 4.4(d) presents examples of moving averages (MA) for the residuals of temperature such as 7 and 54 days. In addition, the raw water level data presented in Figure 4.4(b) is truncated to 196 m in order to account for the physical constrain associated with the bottom of the dam. This step is carried out before centering the dataset as shown by the lower flat regions in Figure 4.4(c).

Model Formulation

Each of the four time series, i.e., displacement, mean-centered water level, average long-term trend, and temperature moving averages are modeled using set of BDLM components. The mean-centered water level is modeled using the AR component that captures the short-term periodic pattern while the average long-term trend ($x^{\mathtt{L}}$) is modeled using a local trend component with a non-zero $\mathbf{Q}^{\mathtt{LT}}$ matrix to allow the model to capture the non-stationary pattern. The residuals for temperature are also modeled using the AR component with which linear dependencies are considered through the regression coefficients $\boldsymbol{\beta}^{\mathtt{T-MA}}$ in order to model the daily fluctuations in the CB2 data. Note that nonlinear dependencies with the temperature did not improve the predictions and hence, are not considered in this study. Finally, the CB2 dataset is modeled using a local level to model the constant baseline, a kernel regression to model the periodic pattern, two state regression (SR) components, i.e., $\mathtt{SR}_1$ and $\mathtt{SR}_2$, to model nonlinear dependencies of displacement on 1) mean-centered water level (WL1) and 2) average long-term trend (WL2), and the AR to model the residuals. In this case study, 20 control-points are considered for both the SR and the KR components. The vector of hidden



(a) CB2 (b) raw water level

(c) water level components (d) MA of TB

Figure 4.4 Illustration showing the available CB2 dataset along with the reservoir water level and examples of moving averages (MA) for the residuals of temperature (TB);(a) shows the raw CB2 data using red dotted points, (b) presents the raw daily dataset for water level in red solid line, (c) provides the mean-centered data showing the short-term periodic but non-harmonic fluctuations in blue solid line along with the average long-term trend (x^{L}) in red solid line, and (d) presents 7 and 54 days MA for TB.

states for all the components combined together at time t are

$$\boldsymbol{x}_t = [\overbrace{\underbrace{x^{\mathrm{LL}}\ (\boldsymbol{x}^{\mathrm{KR}})^\intercal\ x^{\mathrm{AR}}}_{\mathrm{N_1}}\ x^{\mathrm{L,WL1}}\ x^{\mathrm{LT}}}^{\mathrm{CB2}}\ \overbrace{\underbrace{(\tilde{\boldsymbol{x}}^{\mathrm{SK_1}})^\intercal\ (\boldsymbol{x}^{\phi_1^{\mathrm{R}}})^\intercal\ x_0^{\phi_1^{\mathrm{R}}}\ x^{\mathrm{D_1}}\ (\boldsymbol{x}^{\mathrm{P_1}})^\intercal}_{\boldsymbol{x}^{\mathrm{SR_1}}}}^{\mathrm{WL1}}\ \overbrace{x^{\mathrm{AR,WL2}}\ \underbrace{(\tilde{\boldsymbol{x}}^{\mathrm{SK_2}})^\intercal\ (\boldsymbol{x}^{\phi_2^{\mathrm{R}}})^\intercal\ x_0^{\phi_2^{\mathrm{R}}}\ x^{\mathrm{D_2}}\ (\boldsymbol{x}^{\mathrm{P_2}})^\intercal}_{\boldsymbol{x}^{\mathrm{SR_2}}}}^{\mathrm{WL2}}\ \overbrace{(\boldsymbol{x}^{\mathrm{AR}})^\intercal]^\intercal_t}^{\mathrm{T-MA}},$$

$\mathrm{N_3}$ and $\mathrm{N_2}$ (4.8)

where the hidden state vectors for the SR components $\boldsymbol{x}^{\mathrm{SR_1}} = [(\tilde{\boldsymbol{x}}^{\mathrm{SK_1}})^\intercal\ (\boldsymbol{x}^{\phi_1^{\mathrm{R}}})^\intercal\ x_0^{\phi_1^{\mathrm{R}}}\ x^{\mathrm{D_1}}\ (\boldsymbol{x}^{\mathrm{P_1}})^\intercal]^\intercal$ and $\boldsymbol{x}^{\mathrm{SR_2}} = [(\tilde{\boldsymbol{x}}^{\mathrm{SK_2}})^\intercal\ (\boldsymbol{x}^{\phi_2^{\mathrm{R}}})^\intercal\ x_0^{\phi_2^{\mathrm{R}}}\ x^{\mathrm{D_2}}\ (\boldsymbol{x}^{\mathrm{P_2}})^\intercal]^\intercal$, are defined using Equation 4.6 for which the reference hidden states are $\boldsymbol{x}^{\mathrm{L,WL1}}$ and $\boldsymbol{x}^{\mathrm{AR,WL2}}$ for the nonlinear dependency with WL1 and WL2, respectively.

State Estimation

Using the prior knowledge of the hidden state vector $\boldsymbol{x}$ defined in Equation 4.8, the objective is to obtain its posterior knowledge using the Kalman filter. This procedure can be done analytically and is summarized as follows:

The first task is to obtain the prior moments for the normalized kernel outputs $\tilde{\boldsymbol{x}}^{\mathtt{SK}}$ associated with the SR component. This step is carried out using the procedure described in Section 4.2.1 that uses the moments for the hidden states $\boldsymbol{X}^{\mathtt{L,WL1}}_{t-1|t-1}$ and $\boldsymbol{X}^{\mathtt{AR,WL2}}_{t-1|t-1}$ defined in the prior moments $\boldsymbol{\mu}_{t-1|t-1}$ and $\boldsymbol{\Sigma}_{t-1|t-1}$. The updated prior mean vector $\boldsymbol{\mu}_{1,t-1|t-1}$ and covariance matrix $\boldsymbol{\Sigma}_{1,t-1|t-1}$ are obtained by

$$\begin{aligned}
\boldsymbol{\mu}_{1,t-1|t-1} &= \boldsymbol{\mu}_{t-1|t-1} + \begin{bmatrix} \mathbf{0}_{1\times \mathtt{N}_1} & (\tilde{\boldsymbol{\mu}}^{\mathtt{SK}_1})^{\intercal} & \mathbf{0}_{1\times(2\mathtt{N}+3)} & (\tilde{\boldsymbol{\mu}}^{\mathtt{SK}_2})^{\intercal} & \mathbf{0}_{1\times(2\mathtt{N}+7)} \end{bmatrix}^{\intercal}_{t-1|t-1}, \\
\boldsymbol{\Sigma}_{1,t-1|t-1} &= \boldsymbol{\Sigma}_{t-1|t-1} + \boldsymbol{\Sigma}^{\mathtt{A}}_{t-1|t-1} + (\boldsymbol{\Sigma}^{\mathtt{A}}_{t-1|t-1})^{\intercal},
\end{aligned} \tag{4.9}$$

where $\mathtt{N}_1$ represents the total number of hidden states from $x^{\mathtt{LL}}$ of CB2 up to $x^{\mathtt{LT}}$ of WL1 in the hidden state vector $\boldsymbol{x}$ as shown in Equation 4.8, N is the total number of control-points for the SR component, and the matrix $\boldsymbol{\Sigma}^{\mathtt{A}}$ is given by

$$\boldsymbol{\Sigma}^{\mathtt{A}}_{t-1|t-1} = \begin{bmatrix} \mathbf{0}_{\mathtt{X}\times \mathtt{N}_1} & \mathrm{cov}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\mathtt{SK}_1}) & \mathbf{0}_{1\times(2\mathtt{N}+3)} & \mathrm{cov}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\mathtt{SK}_2}) & \mathbf{0}_{\mathtt{X}\times(2\mathtt{N}+7)} \end{bmatrix}_{t-1|t-1},$$

which consists of only the covariance terms between the state vector $\boldsymbol{X}$ and the hidden states for the normalized kernel outputs, i.e., $\mathrm{cov}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\mathtt{SK}_1})$ and $\mathrm{cov}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\mathtt{SK}_2})$, along with sub-matrices of zeros to make the matrix addition compatible. Note that X represents the total number of hidden states in $\boldsymbol{x}$.

Using the moments for the normalized kernel outputs as shown in Equation 4.9 and the Gaussian multiplicative approximation (GMA), the second task is obtain the moments for the product terms $\boldsymbol{x}^{\mathtt{P}_1} = \boldsymbol{x}^{\mathtt{SK}_1} \cdot \boldsymbol{x}^{\phi^{\mathtt{R}}_1}$ and $\boldsymbol{x}^{\mathtt{P}_2} = \boldsymbol{x}^{\mathtt{SK}_2} \cdot \boldsymbol{x}^{\phi^{\mathtt{R}}_2}$ in the two SR components. The updated prior moments $\boldsymbol{\mu}_{2,t-1|t-1}$ and $\boldsymbol{\Sigma}_{2,t-1|t-1}$ are

$$\begin{aligned}
\boldsymbol{\mu}_{2,t-1|t-1} &= \boldsymbol{\mu}_{1,t-1|t-1} + \begin{bmatrix} \mathbf{0}_{1\times \mathtt{N}_2} & (\boldsymbol{\mu}^{\mathtt{P}_1})^{\intercal} & \mathbf{0}_{1\times(2\mathtt{N}+3)} & (\boldsymbol{\mu}^{\mathtt{P}_2})^{\intercal} & \mathbf{0}_{1\times 5} \end{bmatrix}^{\intercal}_{t-1|t-1}, \\
\boldsymbol{\Sigma}_{2,t-1|t-1} &= \boldsymbol{\Sigma}_{t-1|t-1} + \boldsymbol{\Sigma}^{\mathtt{B}}_{t-1|t-1} + (\boldsymbol{\Sigma}^{\mathtt{B}}_{t-1|t-1})^{\intercal},
\end{aligned} \tag{4.10}$$

where $\mathtt{N}_2$ represents the total number of hidden states from $x^{\mathtt{LL}}$ of CB2 up to $x^{\mathtt{D}_1}$ of WL1 in the hidden state vector $\boldsymbol{x}$ as shown in Equation 4.8 and the matrix $\boldsymbol{\Sigma}^{\mathtt{B}}_{t-1|t-1}$ is given by

$$\boldsymbol{\Sigma}^{\mathtt{B}}_{t-1|t-1} = \begin{bmatrix} \mathbf{0}_{\mathtt{X}\times \mathtt{N}_2} & \mathrm{cov}(\boldsymbol{X}, \boldsymbol{X}^{\mathtt{P}_1}) & \mathbf{0}_{1\times(2\mathtt{N}+3)} & \mathrm{cov}(\boldsymbol{X}, \boldsymbol{X}^{\mathtt{P}_2}) & \mathbf{0}_{\mathtt{X}\times 5} \end{bmatrix}_{t-1|t-1}.$$

Using the prior mean vector $\boldsymbol{\mu}_{2,t-1|t-1}$ and covariance matrix $\boldsymbol{\Sigma}_{2,t-1|t-1}$ defined in Equation 4.10, and the matrices $\mathbf{A}$ and $\mathbf{Q}$, the next task is to perform the prediction step of the Kalman filtering procedure (see Section 2.2.2). This task is further divided into two steps where first, the predicted mean vector $\tilde{\boldsymbol{\mu}}_{1,t|t-1}$ and covariance matrix $\tilde{\boldsymbol{\Sigma}}_{1,t|t-1}$ at time step t are obtained from

$$\tilde{\boldsymbol{\mu}}_{1,t|t-1} = \mathbf{A}\boldsymbol{\mu}_{2,t-1|t-1}, \tag{4.11}$$

$$\tilde{\boldsymbol{\Sigma}}_{1,t|t-1} = \mathbf{A}\boldsymbol{\Sigma}_{2,t-1|t-1}\mathbf{A}^{\intercal} + \mathbf{Q}, \tag{4.12}$$

where $\mathbf{A}$ is the global transition matrix and $\mathbf{Q}$ is the process error covariance matrix associated with the process error terms. The mathematical formulation for the model matrices $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$, and $\mathbf{R}$ defining the transition and observation equations are presented in Appendix B.1. By performing this step, the predicted regression coefficients, i.e., $x_0^{\phi_1^{\mathtt{R}}}$ and $x_0^{\phi_2^{\mathtt{R}}}$, defined in Equation 4.5 are automatically obtained because this information is embedded in the matrix $\mathbf{A}^{\mathtt{SR}}$ within the global $\mathbf{A}$ matrix.

The final task in the prediction step is to leverage GMA for obtaining the moments of the hidden states associated with the two interdependent time series, namely $x^{\mathtt{D}_1} = (x_0^{\phi_1^{\mathtt{R}}} \cdot x^{\mathtt{L,WL1}})$ and $x^{\mathtt{D}_2} = (x_0^{\phi_2^{\mathtt{R}}} \cdot x^{\mathtt{AR,WL2}})$,

$$\begin{aligned}
\tilde{\boldsymbol{\mu}}_{t|t-1} &= \tilde{\boldsymbol{\mu}}_{1,t|t-1} + \begin{bmatrix} \mathbf{0}_{1\times \mathtt{N}_3} & (\boldsymbol{\mu}^{\mathtt{D}_1})^{\intercal} & \mathbf{0}_{1\times(2\mathtt{N}+2)} & (\boldsymbol{\mu}^{\mathtt{D}_2})^{\intercal} & \mathbf{0}_{1\times(\mathtt{N}+5)} \end{bmatrix}_{t|t-1}^{\intercal}, \\
\tilde{\boldsymbol{\Sigma}}_{t|t-1} &= \tilde{\boldsymbol{\Sigma}}_{1,t|t-1} + \boldsymbol{\Sigma}_{t|t-1}^{\mathtt{C}} + (\boldsymbol{\Sigma}_{t|t-1}^{\mathtt{C}})^{\intercal},
\end{aligned} \tag{4.13}$$

where $\mathtt{N}_3$ represents the total number of hidden states from $x^{\mathtt{LL}}$ of $\mathtt{CB2}$ up to $x_0^{\phi_1^{\mathtt{R}}}$ in the hidden state vector $\boldsymbol{x}$ as shown in Equation 4.8 and the matrix $\boldsymbol{\Sigma}_{t|t-1}^{\mathtt{C}}$ is given by

$$\boldsymbol{\Sigma}_{t|t-1}^{\mathtt{C}} = \begin{bmatrix} \mathbf{0}_{\mathtt{X}\times \mathtt{N}_3} & \mathrm{cov}(\boldsymbol{X}, \boldsymbol{X}^{\mathtt{D}_1}) & \mathbf{0}_{1\times(2\mathtt{N}+2)} & \mathrm{cov}(\boldsymbol{X}, \boldsymbol{X}^{\mathtt{D}_2}) & \mathbf{0}_{\mathtt{X}\times(\mathtt{N}+5)} \end{bmatrix}_{t|t-1}.$$

Using the predicted moments $\tilde{\boldsymbol{\mu}}_{t|t-1}$ and $\tilde{\boldsymbol{\Sigma}}_{t|t-1}$ defined in Equation 4.13, the update step of the Kalman filter is performed as described in Section 2.2.2. All the steps carried out at a particular time step t for the proposed state-based regression method are summarized in Algorithm 1.

Model Forecast

Figure 4.5(a) shows the observed data in red, the estimated values $\boldsymbol{\mu}_{t|t}$ in black solid line and their uncertainty bounds $\boldsymbol{\mu}_{t|t} \pm \boldsymbol{\sigma}_{t|t}$ are shown by a green shaded region as obtained using

Algorithm 1 One-time step of the state-based regression method

Input: $\boldsymbol{\mu}_{t-1|t-1}$, $\boldsymbol{\Sigma}_{t-1|t-1}$, $\boldsymbol{y}_t$, $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$, and $\mathbf{R}$

Prior moments for Normalized Kernel Outputs $\tilde{\boldsymbol{x}}^{\mathtt{SK}}$:

$$\boldsymbol{\mu}_{1,t-1|t-1} = \boldsymbol{\mu}_{t-1|t-1} + \left[\begin{array}{cccccc} \mathbf{0}_{1\times \mathtt{N}_1} & (\tilde{\boldsymbol{\mu}}^{\mathtt{SK}_1})^\intercal & \mathbf{0}_{1\times(2\mathtt{N}+3)} & (\tilde{\boldsymbol{\mu}}^{\mathtt{SK}_2})^\intercal & \mathbf{0}_{1\times(2\mathtt{N}+7)} \end{array}\right]^\intercal_{t-1|t-1},$$

$$\boldsymbol{\Sigma}_{1,t-1|t-1} = \boldsymbol{\Sigma}_{t-1|t-1} + \boldsymbol{\Sigma}^{\mathtt{A}}_{t-1|t-1} + (\boldsymbol{\Sigma}^{\mathtt{A}}_{t-1|t-1})^\intercal.$$

Moments for the Product Terms $\boldsymbol{x}^{\mathtt{P}_1} = \boldsymbol{x}^{\mathtt{SK}_1} \cdot \boldsymbol{x}^{\phi^{\mathtt{R}}_1}$ and $\boldsymbol{x}^{\mathtt{P}_2} = \boldsymbol{x}^{\mathtt{SK}_2} \cdot \boldsymbol{x}^{\phi^{\mathtt{R}}_2}$:

$$\boldsymbol{\mu}_{2,t-1|t-1} = \boldsymbol{\mu}_{1,t-1|t-1} + \left[\begin{array}{cccccc} \mathbf{0}_{1\times \mathtt{N}_2} & (\boldsymbol{\mu}^{\mathtt{P}_1})^\intercal & \mathbf{0}_{1\times(2\mathtt{N}+3)} & (\boldsymbol{\mu}^{\mathtt{P}_2})^\intercal & \mathbf{0}_{1\times 5} \end{array}\right]^\intercal_{t-1|t-1},$$

$$\boldsymbol{\Sigma}_{2,t-1|t-1} = \boldsymbol{\Sigma}_{t-1|t-1} + \boldsymbol{\Sigma}^{\mathtt{B}}_{t-1|t-1} + (\boldsymbol{\Sigma}^{\mathtt{B}}_{t-1|t-1})^\intercal.$$

1^{st} **Prediction Step:**

$$\tilde{\boldsymbol{\mu}}_{1,t|t-1} = \mathbf{A}\boldsymbol{\mu}_{2,t-1|t-1},$$

$$\tilde{\boldsymbol{\Sigma}}_{1,t|t-1} = \mathbf{A}\boldsymbol{\Sigma}_{2,t-1|t-1}\mathbf{A}^\intercal + \mathbf{Q}.$$

2^{nd} **Prediction Step:**

$$\tilde{\boldsymbol{\mu}}_{t|t-1} = \tilde{\boldsymbol{\mu}}_{1,t|t-1} + \left[\begin{array}{cccccc} \mathbf{0}_{1\times \mathtt{N}_3} & (\boldsymbol{\mu}^{\mathtt{D}_1})^\intercal & \mathbf{0}_{1\times(2\mathtt{N}+2)} & (\boldsymbol{\mu}^{\mathtt{D}_2})^\intercal & \mathbf{0}_{1\times(\mathtt{N}+5)} \end{array}\right]^\intercal_{t|t-1},$$

$$\tilde{\boldsymbol{\Sigma}}_{t|t-1} = \tilde{\boldsymbol{\Sigma}}_{1,t|t-1} + \boldsymbol{\Sigma}^{\mathtt{C}}_{t|t-1} + (\boldsymbol{\Sigma}^{\mathtt{C}}_{t|t-1})^\intercal,$$

$$\boldsymbol{r}_t = \boldsymbol{y}_t - \mathbf{C}\tilde{\boldsymbol{\mu}}_{t|t-1},$$

$$\mathbf{K} = \tilde{\boldsymbol{\Sigma}}_{2,t-1|t-1}\mathbf{C}^\intercal(\mathbf{C}\boldsymbol{\Sigma}_{2,t-1|t-1}\mathbf{C}^\intercal + \mathbf{R})^{-1}.$$

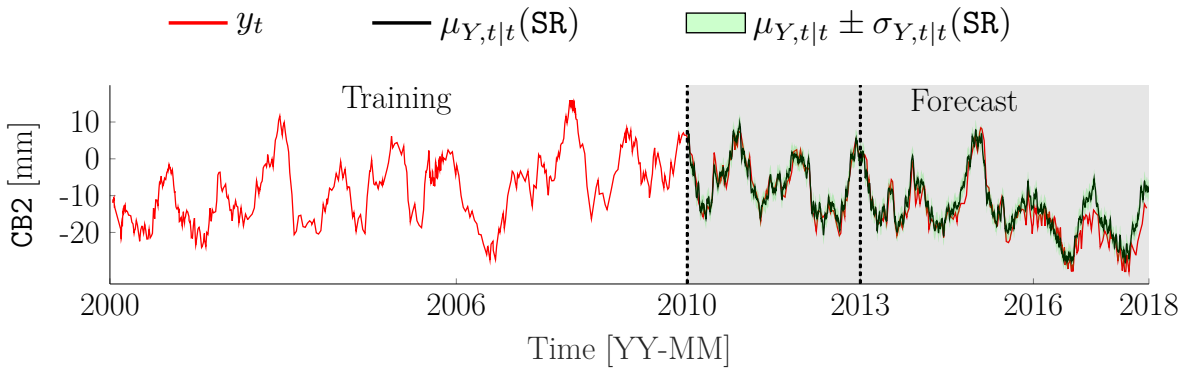
Update Step:

$$\boldsymbol{\mu}_{t|t} = \tilde{\boldsymbol{\mu}}_{t|t-1} + \mathbf{K}\boldsymbol{r}_t,$$

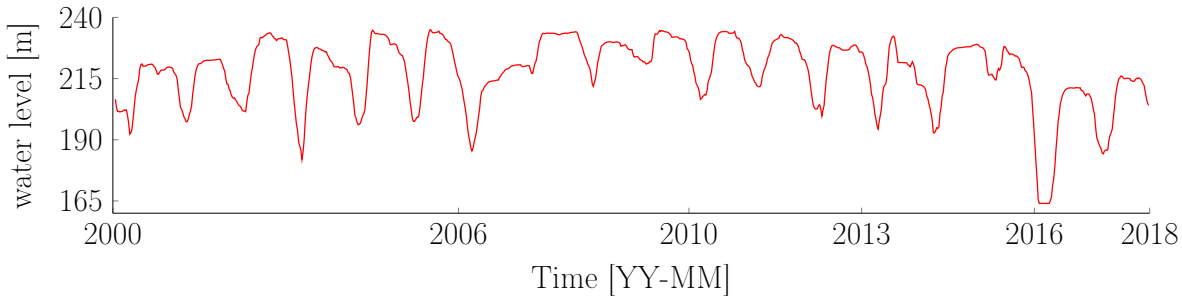
$$\boldsymbol{\Sigma}_{t|t} = (\mathbf{I} - \mathbf{KC})\tilde{\boldsymbol{\Sigma}}_{t|t-1}.$$

return $\boldsymbol{\mu}_{t|t}$, $\boldsymbol{\Sigma}_{t|t}$

the SR method. Figure 4.5(b) shows the raw water level time series which is available for the entire duration. The validation and the test data are shown by the gray region; The training data is from 2000 to 2010, the validation data is from 2010 to 2013 which is marked by the region between the two dashed lines, and the test data is from 2013 to 2018. The observations for the period 2013 to 2018 are retrieved from the summary results provided by the ICOLD Benchmark formulators [107]. The forecasts are produced using the hidden states for the local level, kernel regression, and autoregressive components associated with the CB2 dataset, as well as the hidden states for the nonlinear dependency with the two



(a) CB2 forecast



(b) water level data

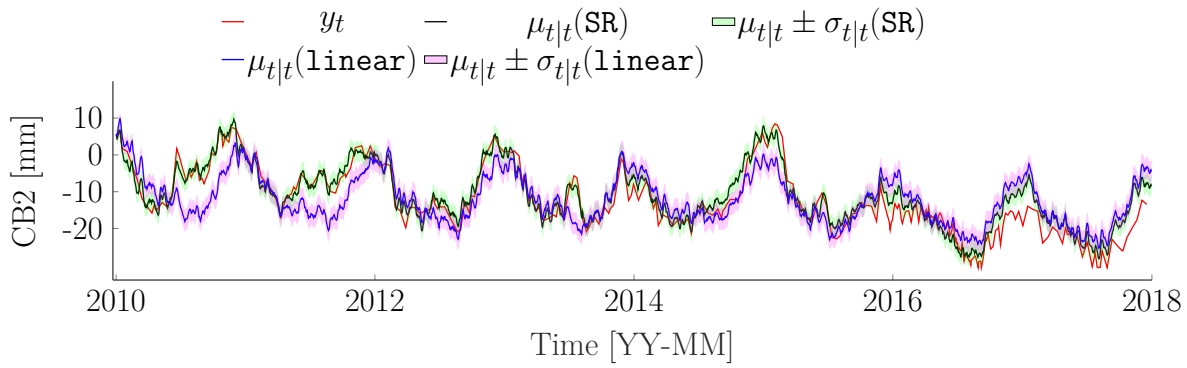
Figure 4.5 Plots showing (a) the estimated values for the CB2 time series using the state-based regression method and (b) the water level time series. The red solid line shows the observations, the black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions. The validation and the test data are shown by the gray region; The training data is from 2000 to 2010, the validation data is from 2010 to 2013 which is marked by the region between the two dashed lines, and the test data is from 2013 to 2018.

water level time series and also through the linear regression coefficients associated with the temperature's residuals.

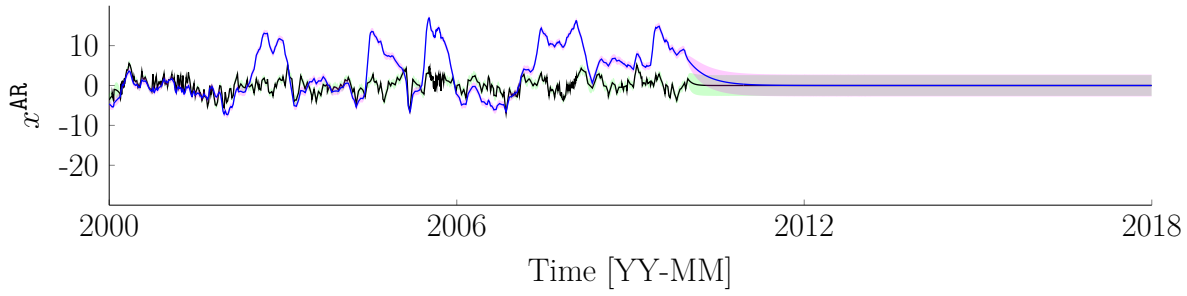
The results for the validation and the forecast period show that the method is capable of accurately identifying the short-term as well as the long-term patterns in the CB2 time series. However, there is bias in the predictions beyond the year 2016 as the observations have lower values compared to the predictions. This behavior may be attributed to lower water levels in the forecast period that are not seen during the training period. Moreover, the nonlinear relationship between the water level and the displacement time series might also be different in the forecast period than the one learned during training.

To better evaluate the advantage of identifying nonlinear dependencies, the method proposed

is compared with a linear dependency model (`linear`) in BDLM, where constant regression coefficients are learned in relation to both water level time series. The `linear` model uses the same BDLM components as the proposed method excluding the `SR` component. Figure 4.6(a) shows the forecast values for the `CB2` time series using the `SR` method as well as the `linear` model in BDLM for the period 2010 to 2018 and Figure 4.6(b) shows the residuals collected by the `AR` component in each of the method. Table 4.1 shows the test RMSE and the log-likelihood values obtained with the `SR` method and the `linear` model in BDLM for the entire forecast period. Note that these metrics are evaluated using only the observations available from the period 2010 to 2018. The results show that the `linear` model has a poor predictive performance compared to the `SR` method in terms of both RMSE and log-



(a) forecast



(b) residuals

Figure 4.6 Plots showing (a) the forecast values for the `CB2` time series using the state-based regression (`SR`) method as well as the linear dependency (`linear`) model in BDLM for the period 2010 to 2018 and (b) the residuals collected by the `AR` component in each of the method. The red solid line shows the observations, the black solid line and the green shaded region shows the estimated values and their $\pm 1\sigma$ confidence regions obtained using the `SR` method, while the blue solid line and the pink shaded region shows the predictions and their $\pm 1\sigma$ confidence regions obtained using the `linear` model.

Table 4.1 Root mean square error (RMSE) and log-likelihood values obtained with the state-based regression (SR) method and the linear dependency (linear) model in BDLM for the CB2 dataset.

metric	RMSE	Log-likelihood
SR	**2.80**	**−793.38**
Linear	5.92	−1302.90

likelihood. Moreover, there are significant patterns that remain in the residuals from the linear model as opposed to the ones obtained from the proposed method. This shows that the linear model is limited for identifying complex behavior in dam's displacement that may arise due to nonlinear dependencies with other explanatory variables such as the water level.

Model Interpretation

Following the BDLM structure used in this case study, the hidden states that contribute to the observations are $x^{\mathtt{LL}}$, $x^{\mathtt{D}_1}$, $x_0^{\mathtt{KR}}$, $x^{\mathtt{D}_2}$, and $x^{\mathtt{AR}}$. Figure 4.7 summarizes the contribution of these hidden states to the prediction shown in Figure 4.5(a); where (a) demonstrates the constant average value of the time series shown by the hidden state $x^{\mathtt{LL}}$, (b) represents the pattern obtained by adding $x^{\mathtt{LL}}$ and the interdependent hidden state $x^{\mathtt{D}_1}$ associated with the $\mathtt{SR}_1$ component, (c) represents the pattern obtained by adding the kernel regression hidden state representing the stationary periodic pattern $x_0^{\mathtt{KR}}$ with the hidden states $x^{\mathtt{LL}}$ and $x^{\mathtt{D}_1}$, (d) represents total pattern captured by the addition of the hidden states $x^{\mathtt{LL}}$, $x^{\mathtt{D}_1}$, $x_0^{\mathtt{KR}}$ and the interdependent hidden state $x^{\mathtt{D}_2}$ associated with the $\mathtt{SR}_2$ component, and (e) represents the residuals captured by the AR component. The red solid line presents the observed data, the black solid line and the green shaded region show the predictions and their $\pm 1\sigma$ confidence regions. The residuals from the CB2 dataset shown in Figure 4.7(e) demonstrate that the SR method is able to capture the patterns from the data. Moreover, both the nonlinear dependencies have a significant contribution to the predictions for CB2 in identifying the non-stationary long-term trend as well as the periodic but non-harmonic fluctuations in the time series.

Figure 4.8(a)-(d) present the hidden state estimations of the predicted regression coefficient $x_0^{\phi^{\mathtt{R}}}$ and the interdependent time series $x^{\mathtt{D}}$ for the two SR components. Figure 4.8(e) and (f) show that the estimated regression coefficients are state-dependent as they vary from one

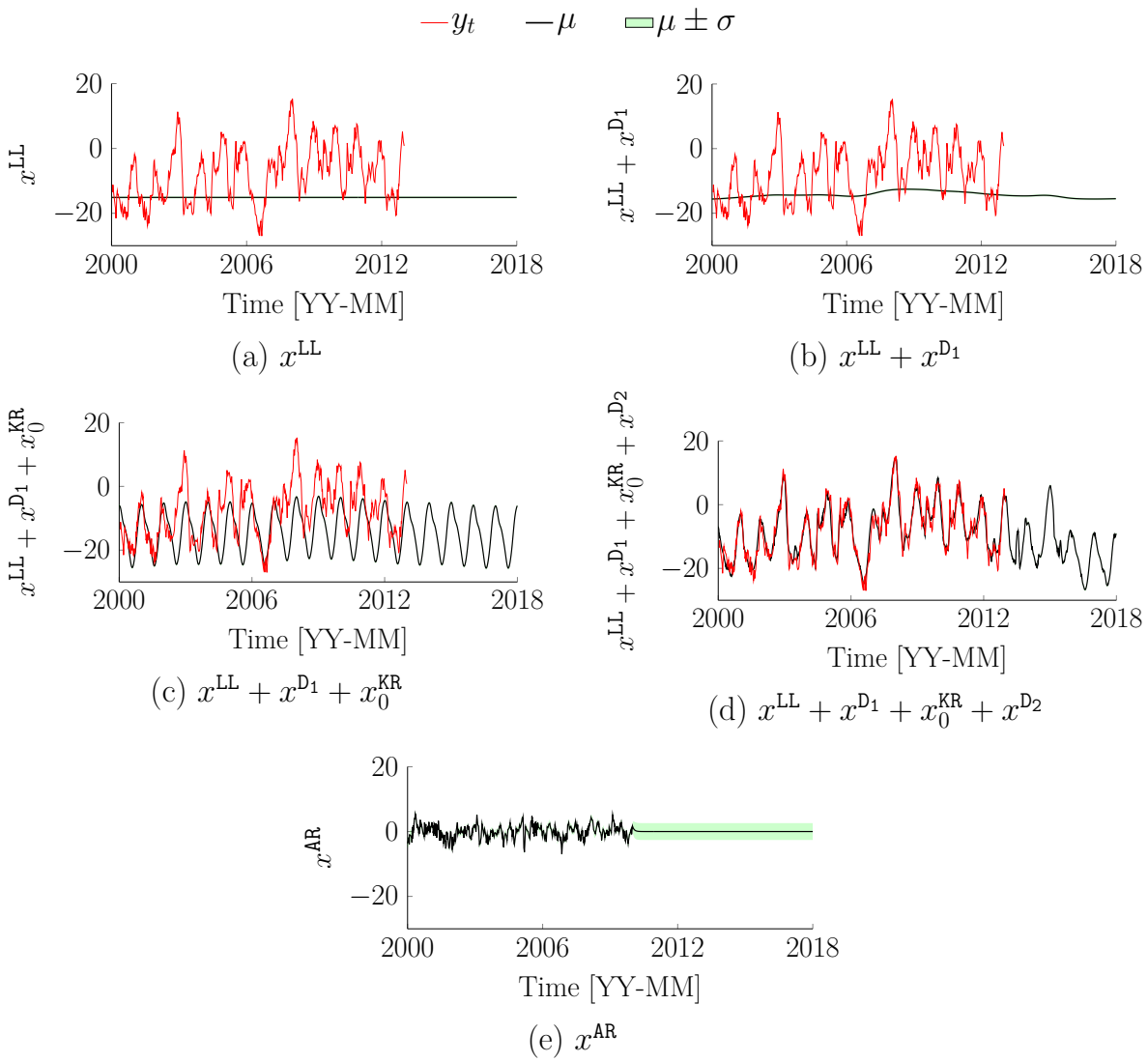
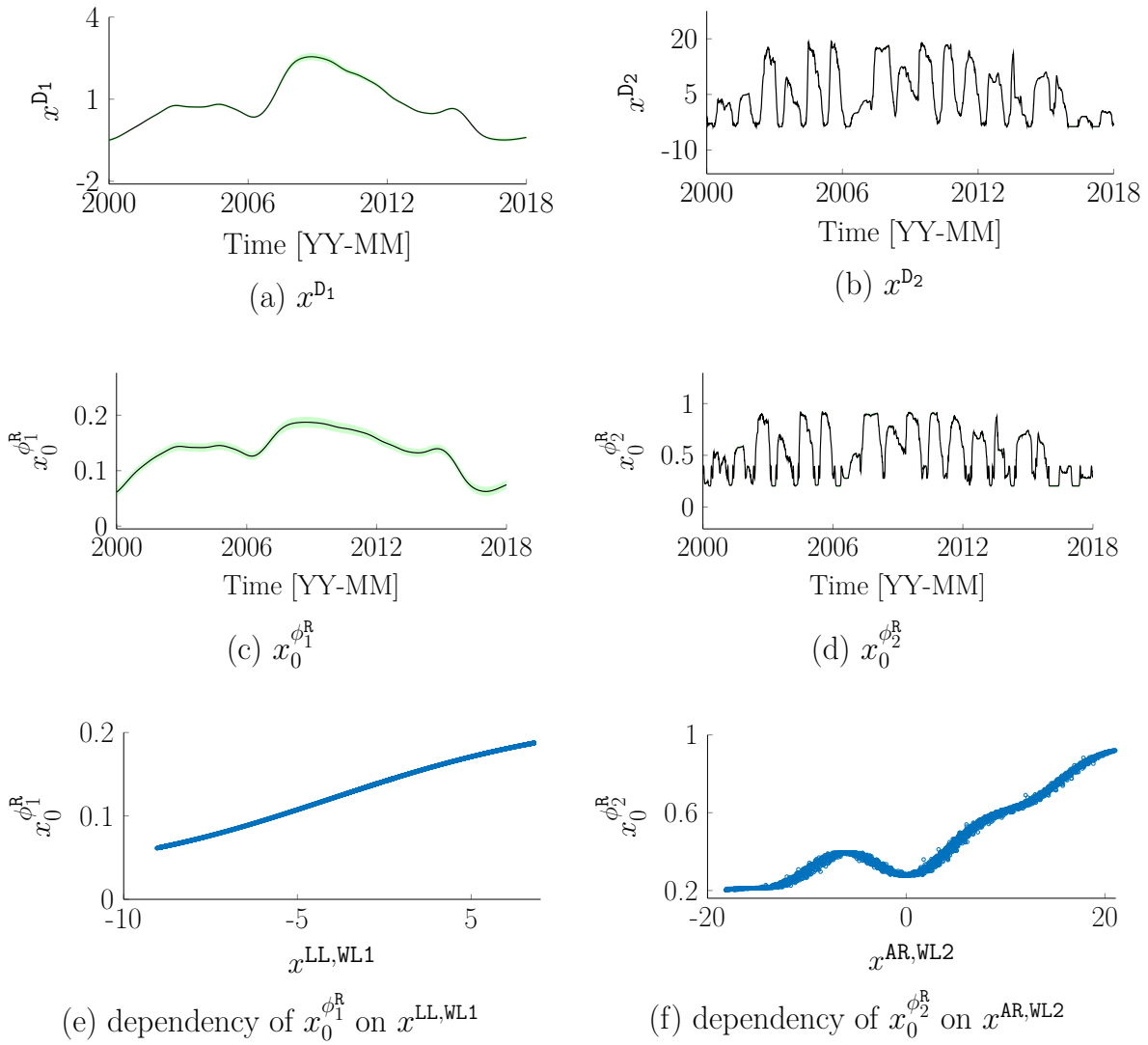


Figure 4.7 Plot showing the contribution of each of the hidden states to the `CB2` predictions where (a) demonstrates the constant average value of the time series shown by the hidden state $x^{\mathtt{LL}}$, (b) represents the pattern obtained by adding $x^{\mathtt{LL}}$ and the interdependent hidden state $x^{\mathtt{D}_1}$ associated with the $\mathtt{SR}_1$ component, (c) represents the pattern obtained by adding the kernel regression hidden state representing the stationary periodic pattern $x_0^{\mathtt{KR}}$ with the hidden states $x^{\mathtt{LL}}$ and $x^{\mathtt{D}_1}$, (d) represents total pattern captured by the addition of the hidden states $x^{\mathtt{LL}}$, $x^{\mathtt{D}_1}$, $x_0^{\mathtt{KR}}$ and the interdependent hidden state $x^{\mathtt{D}_2}$ associated with the $\mathtt{SR}_2$ component, and (e) represents the residuals captured by the `AR` component. The red solid line shows the observations, the black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions.

time step to another based on the values of the reference variables $x^{\mathtt{LL,WL1}}$ and $x^{\mathtt{AR,WL2}}$. These coefficients allow us to identify the non-stationary trend as well as periodic fluctuations in the

CB2 dataset as illustrated in Figures 4.8(a) and (b) that arise due to nonlinear dependencies on the water level time series.



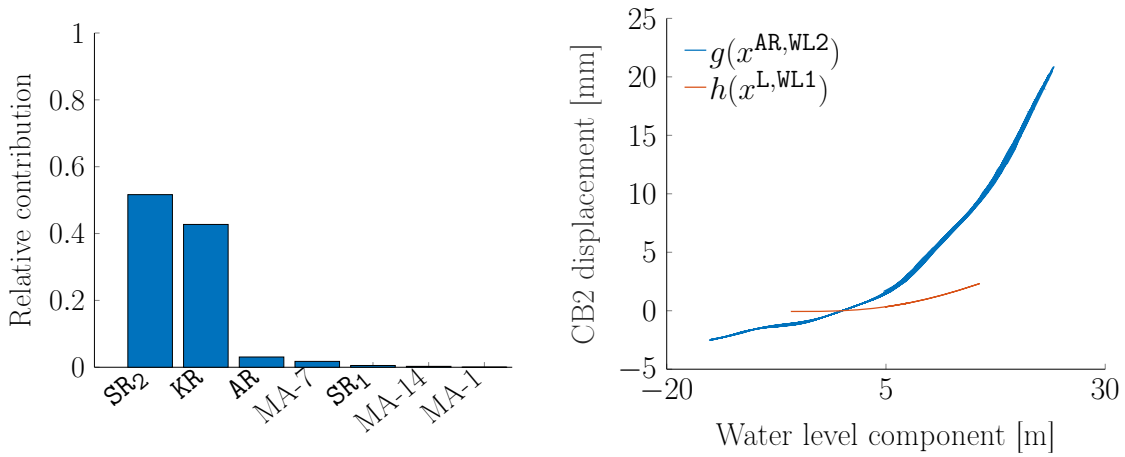
(a) $x^{\mathtt{D}_1}$ (b) $x^{\mathtt{D}_2}$

(c) $x_0^{\phi_1^{\mathtt{R}}}$ (d) $x_0^{\phi_2^{\mathtt{R}}}$

(e) dependency of $x_0^{\phi_1^{\mathtt{R}}}$ on $x^{\mathtt{LL,WL1}}$ (f) dependency of $x_0^{\phi_2^{\mathtt{R}}}$ on $x^{\mathtt{AR,WL2}}$

Figure 4.8 Plot showing the hidden state estimation of the predicted regression coefficient $x_0^{\phi^{\mathtt{R}}}$ and the interdependent time series $x^{\mathtt{D}}$ for the two SR components. The red solid line shows the observations, the black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions.

Figure 4.9 presents additional information that can be extracted from BDLM; (a) it presents the relative importance of each component measured by the relative variance of each component used and (b) it shows the extracted nonlinear relationships between the interdependent time series $x^{\mathtt{D}_1}$ and the long-term trend $x^{\mathtt{L,WL1}}$ shown by $h(x^{\mathtt{L,WL1}})$, and between $x^{\mathtt{D}_2}$ and the mean-centered water level $x^{\mathtt{AR,WL2}}$ shown by $g(x^{\mathtt{AR,WL2}})$. The results in Figure 4.9(a) shows



(a) relative contribution

(b) nonlinear dependencies

Figure 4.9 Illustration showing (a) the relative importance of each component used for modeling CB2 time series, and (b) the extracted nonlinear relationship between the interdependent time series $x^{\mathtt{D}_1}$ and the long-term trend $x^{\mathtt{L,WL1}}$ represented by $h(x^{\mathtt{L,WL1}})$ in blue solid line, and between $x^{\mathtt{D}_2}$ and the mean-centered water level $x^{\mathtt{AR,WL2}}$ represented by $g(x^{\mathtt{AR,WL2}})$ in red solid line.

the dominant relative importance of the mean-centered water level through the non-linear dependency $g(x_{\mathtt{AR,WL2}})$, and followed secondly by the periodic pattern $x_0^{\mathtt{KR}}$. The third most important contributor is the autoregressive component $x^{\mathtt{AR}}$ which represents the residual pattern that cannot be explained by other components. Even though the relative importance of the long-term trend in the water level $x^{\mathtt{L,WL1}}$ is significantly less than even the residual term, it provides the non-stationary baseline for the CB2 time series.

4.3.2 Case Study 2 – CB3 Time Series

This case study is conducted on the CB3 time series which measures the dam's radial displacement in the foundation between the altitudes 195 m and 161 m as shown in Figure 4.3.

Data Description

Similarly to the CB2 dataset as described in Section 4.3.1, the CB3 dataset is also available from the year 2000 to 2012 with an average data acquisition frequency of one data point every 1.5 week. The water level and the air temperature time series are also used as explanatory variables. The raw water level time series is further divided into two components to model the nonlinear dependency of displacement with its average long-term trend and the short-term

periodic pattern. Moreover, the water level is capped below 196 m to account for the bottom of the dam. Also, moving averages of $\{1, 7, 14, 28, 54\}$ days of the temperature's residuals are considered to model the daily fluctuations in the displacement data that occurs as a result of thermal inertia of the dam [17, 108].

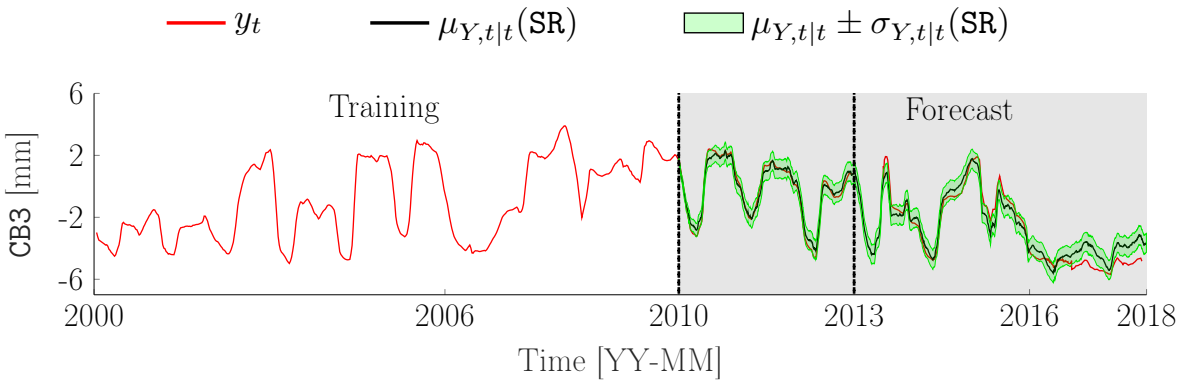
Model Formulation

As described in Section 4.3.1, the same BDLM components are used to model the CB3 dataset as used to formulate the prediction model for the CB2 dataset. The mean-centered water level is modeled using the AR component that captures the short-term periodic pattern while the average long-term trend ($x^{\mathtt{L}}$) is modeled using a local trend component with a non-zero $\mathbf{Q}^{\mathtt{LT}}$ matrix. The residuals for temperature are also modeled using the AR component with which linear dependencies are considered through the regression coefficients $\boldsymbol{\beta}^{\mathtt{T\text{-}MA}}$ in order to model the daily fluctuations in the CB3 data. Finally, the CB3 dataset is modeled using a local level to model the constant baseline, a kernel regression to model the periodic pattern, two state regression (SR) components, i.e., $\mathtt{SR}_1$ and $\mathtt{SR}_2$, to model nonlinear dependencies of displacement on 1) mean-centered water level (WL1) and 2) average long-term trend (WL2), and the AR to model the residuals.

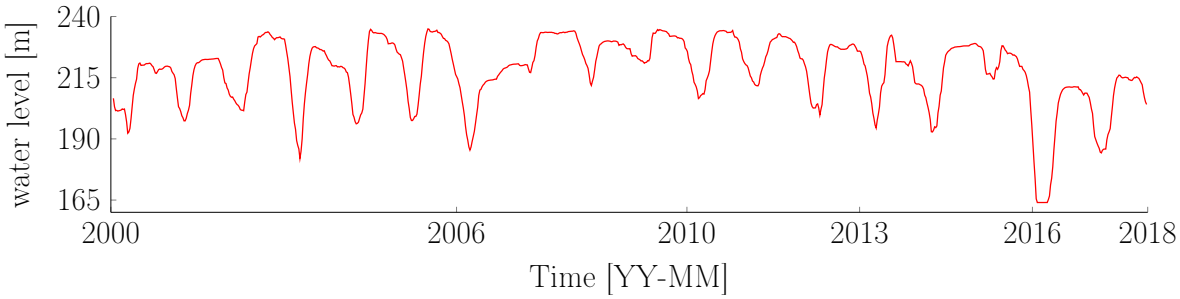
Model Forecast

Figure 4.10(a) shows the observed data in red, the estimated values $\boldsymbol{\mu}_{t|t}$ in black solid line and their uncertainty bounds $\boldsymbol{\mu}_{t|t} \pm \boldsymbol{\sigma}_{t|t}$ are shown by green shaded regions as obtained using the SR method. Figure 4.10(b) shows the raw water level time series which is available from the year 2000 to 2018. The validation and the test data are shown by the gray region; The training data is from 2000 to 2010, the validation data is from 2010 to 2013 which is marked by the region between the two dashed lines, and the test data is from 2013 to 2018. The test data is retrieved from the results provided by the ICOLD Benchmark formulators [107]. The forecasts are produced using the hidden states for the local level, kernel regression, and autoregressive components associated with the CB3 dataset, as well as the hidden states for the nonlinear dependency with the two water level time series and also through the linear regression coefficients associated with the temperature's residuals.

The results for the validation and the forecast period show that the method is capable of accurately identifying the short-term as well as the long-term patterns in the CB2 time series. However, as seen in the CB2 dataset, there is a considerable bias in the predictions beyond the year 2016 as a result of lower water levels in the forecast period that is not seen during the trained period. Figure 4.11 illustrates the model predictions and the residuals obtained



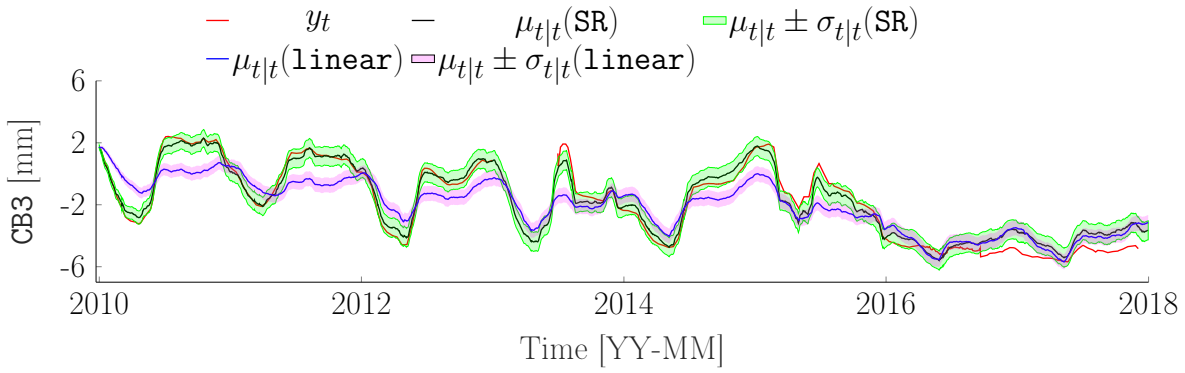
(a) CB3 forecast



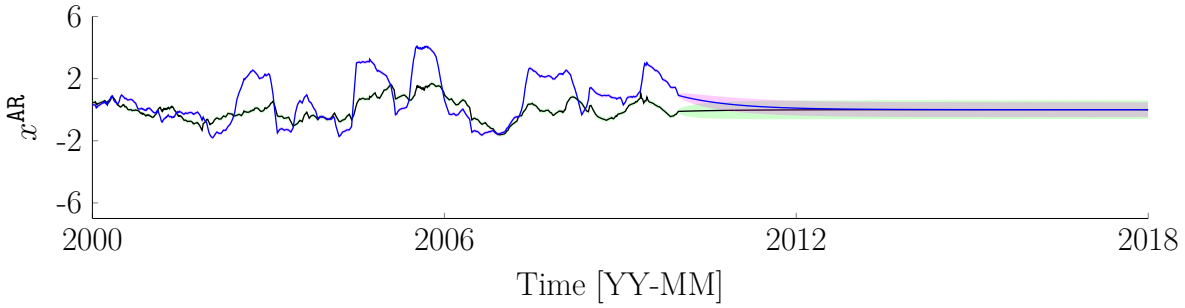
(b) water level data

Figure 4.10 Plots showing (a) the estimated values for the CB3 time series using the state-based regression method and (b) the water level time series. The red solid line shows the observations, the black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions. The validation and the test data are shown by the gray region; The training data is from 2000 to 2010, the validation data is from 2010 to 2013 which is marked by the region between the two dashed lines, and the test data is from 2013 to 2018.

using the SR method and the linear model for the CB3 time series where Figure 4.11(a) shows the forecast values for the period 2010 to 2018 and Figure 4.11(b) shows the residuals collected by the AR component in each of the method. Table 4.2 shows the test RMSE and the log-likelihood values obtained with the SR method and the linear model. Note that these metrics are evaluated using only the observations available from the period 2010 to 2018. The results show that the linear model has a poor predictive performance compared to the SR method in terms of both RMSE and log-likelihood. Moreover, there are significant patterns that remain in the residuals from the linear model compared to the ones obtained from the proposed method. Figure 4.11 illustrates the model predictions and the residuals obtained using the SR method and the linear model for the CB3 time series where Figure 4.11(a)



(a) CB3 forecast



(b) residuals

Figure 4.11 Plots showing (a) the forecast values for the CB3 time series using the state-based regression (SR) method as well as the linear dependency (linear) model in BDLM for the period 2010 to 2018 and (b) the residuals collected by the AR component in each of the method. The red solid line shows the observations, the black solid line and the green shaded region shows the estimated values and their $\pm 1\sigma$ confidence regions obtained using the SR method, while the blue solid line and the pink shaded region shows the predictions and their $\pm 1\sigma$ confidence regions obtained using the linear model.

Table 4.2 Root mean square error (RMSE) and log-likelihood values obtained with the state-based regression (SR) method and the linear dependency (linear) model in BDLM for the CB3 dataset.

metric	RMSE	Log-likelihood
SR	**1.49**	**−1079.8**
Linear	1.51	−1787.2

shows the forecast values for the period 2010 to 2018 and Figure 4.11(b) shows the residuals collected by the AR component in each of the method. Table 4.2 shows the test RMSE and the log-likelihood values obtained with the SR method and the linear model. The results show that the linear model has a poor predictive performance compared to the SR method in terms of both RMSE and log-likelihood. Moreover, there are significant patterns that remain in the residuals from the linear model compared to the ones obtained from the proposed method. Even though the SR method has a better predictive capacity than the linear model, the residuals still have non-stationary patterns which cannot be explained by the existing model setup for the CB3 data. Hence, the prediction model provided by the SR method needs to be combined with a regime switching approach [28] to detect such non-stationary patterns in real time.

Model Interpretation

Figure 4.12 summarize the contribution of the hidden states to the predictions shown in Figure 4.10(a); where (a) demonstrates the constant average value of the time series shown by the hidden state $x^{\mathtt{LL}}$, (b) represents the pattern obtained by adding $x^{\mathtt{LL}}$ and the interdependent hidden state $x^{\mathtt{D}_1}$ associated with the $\mathtt{SR}_1$ component, (c) represents the pattern obtained by adding the kernel regression hidden state representing the stationary periodic pattern $x_0^{\mathtt{KR}}$ with the hidden states $x^{\mathtt{LL}}$ and $x^{\mathtt{D}_1}$, (d) represents total pattern captured by the addition of the hidden states $x^{\mathtt{LL}}$, $x^{\mathtt{D}_1}$, $x_0^{\mathtt{KR}}$ and the interdependent hidden state $x^{\mathtt{D}_2}$ associated with the $\mathtt{SR}_2$ component, and (e) represents the residuals captured by the AR component. The red solid line presents the observed data, the black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions.

The results in Figure 4.12(d) and (e) show that even though significant patterns in the CB3 dataset could be captured using the SR method, the residuals still have non-stationary patterns which cannot be explained by the existing model setup. Hence, the prediction model provided by the SR method needs to be combined with a regime switching approach [28] to detect such non-stationary patterns in real time.

Figure 4.13 presents additional information that can be extracted from BDLM; (a) presents the relative importance of each component measured by the relative variance of each component used, (b) shows the extracted nonlinear relationships between the interdependent time series $x^{\mathtt{D}_1}$ and the long-term trend $x^{\mathtt{L,WL1}}$ shown by $h(x^{\mathtt{L,WL1}})$, and between $x^{\mathtt{D}_2}$ and the mean-centered water level $x^{\mathtt{AR,WL2}}$ shown by $g(x^{\mathtt{AR,WL2}})$, (c)-(d) shows the state-dependent regression coefficients that vary based on the values of the reference variables $x^{\mathtt{LL,WL1}}$ and $x^{\mathtt{AR,WL2}}$. The results in Figure 4.13(a) shows the dominant relative importance of the mean-centered

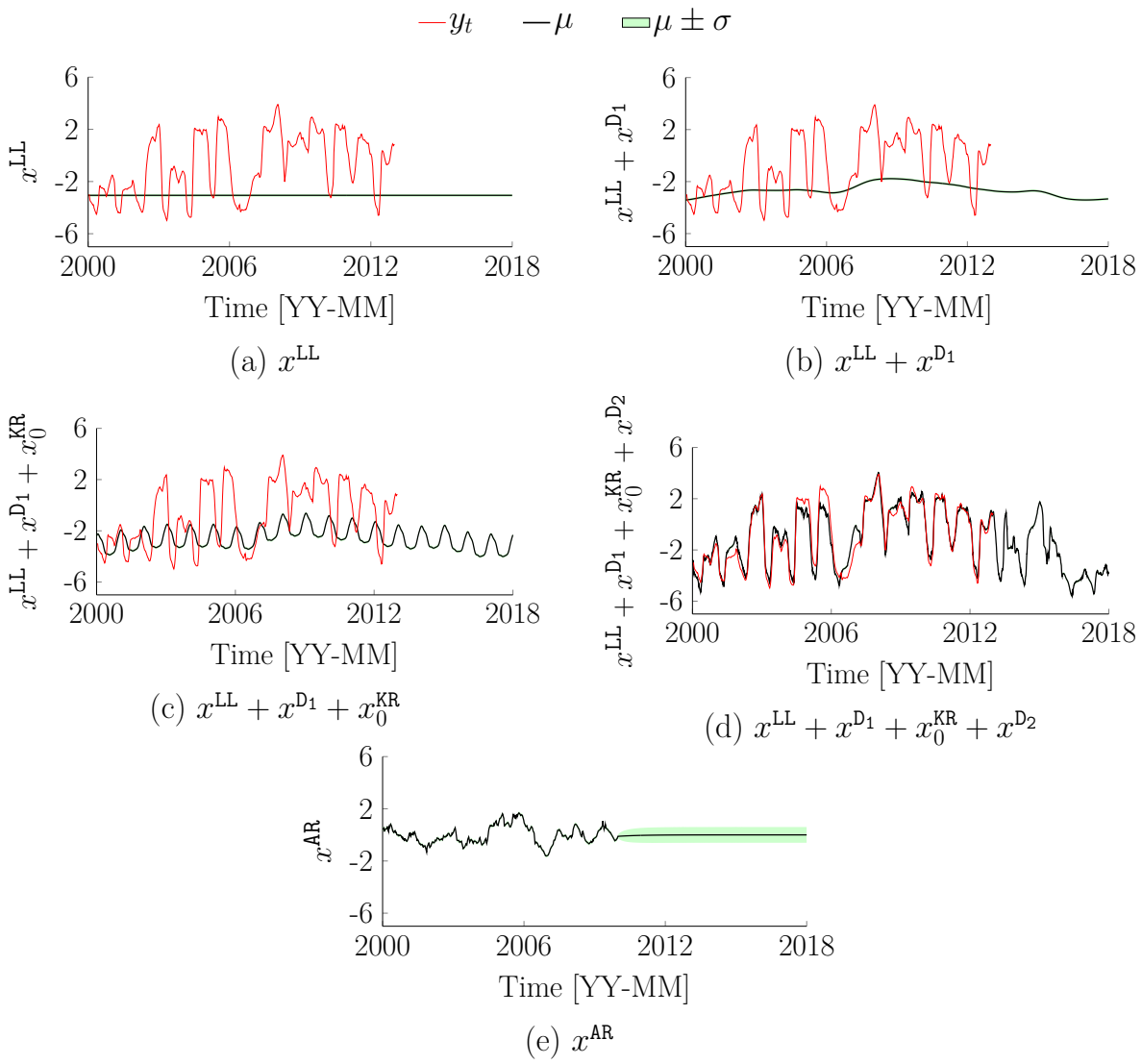
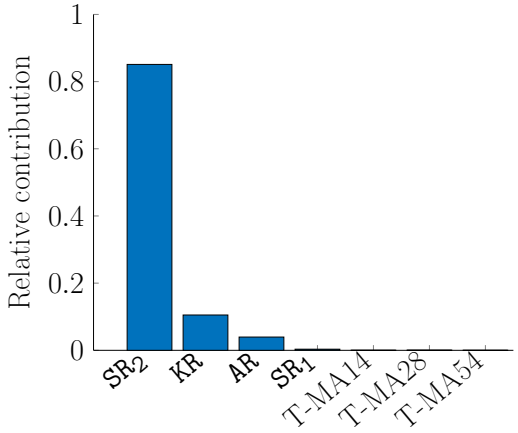


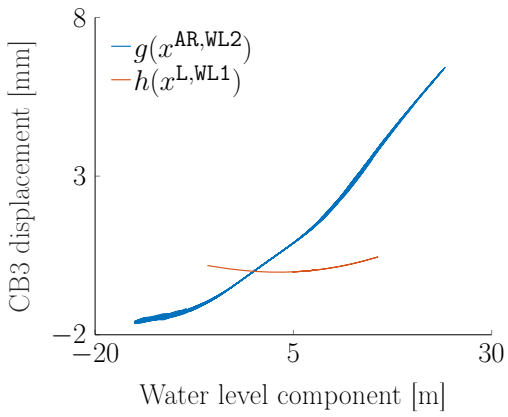
Figure 4.12 Plot showing the contribution of each of the hidden states to the `CB3` predictions where (a) demonstrates the constant average value of the time series shown by the hidden state $x^{\mathtt{LL}}$, (b) represents the pattern obtained by adding $x^{\mathtt{LL}}$ and the interdependent hidden state $x^{\mathtt{D}_1}$ associated with the $\mathtt{SR}_1$ component, (c) represents the pattern obtained by adding the kernel regression hidden state representing the stationary periodic pattern $x_0^{\mathtt{KR}}$ with the hidden states $x^{\mathtt{LL}}$ and $x^{\mathtt{D}_1}$, (d) represents total pattern captured by the addition of the hidden states $x^{\mathtt{LL}}$, $x^{\mathtt{D}_1}$, $x_0^{\mathtt{KR}}$ and the interdependent hidden state $x^{\mathtt{D}_2}$ associated with the $\mathtt{SR}_2$ component, and (e) represents the residuals captured by the `AR` component. The red solid line shows the observations, the black solid line and the green shaded region shows the predictions and their $\pm 1\sigma$ confidence regions.

water level through the non-linear dependency $g(x_{\mathtt{AR},\mathtt{WL2}})$, and followed secondly by the periodic pattern $x_0^{\mathtt{KR}}$. The third most important contributor is the autoregressive component $x^{\mathtt{AR}}$

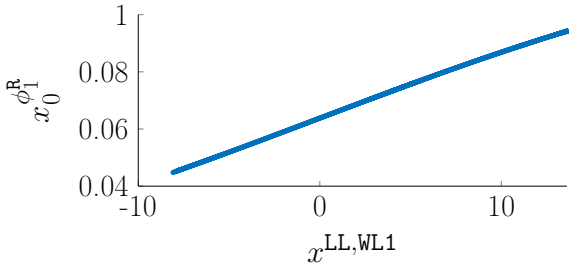
which represents the residual pattern that cannot be explained by other components. The relative importance of the long-term trend in the water level $x^{\mathtt{L,WL1}}$ is comparatively less, but it provides the non-stationary baseline for the CB3 time series.



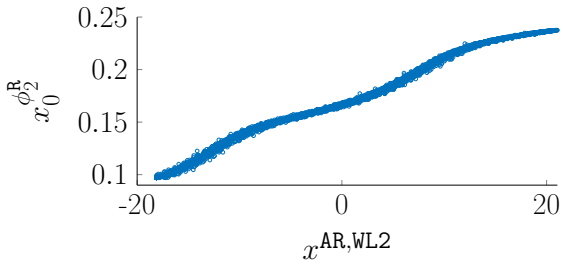
(a) relative contribution



(b) nonlinear dependencies



(c) dependency of $x_0^{\phi_1^{\mathtt{R}}}$ on $x^{\mathtt{LL,WL1}}$



(d) dependency of $x_0^{\phi_2^{\mathtt{R}}}$ on $x^{\mathtt{AR,WL2}}$

Figure 4.13 Illustration showing (a) the relative importance of each component used for modeling CB3 time series, (b) the extracted nonlinear relationship between the interdependent time series $x^{\mathtt{D}_1}$ and the long-term trend $x^{\mathtt{L,WL1}}$ represented by $h(x^{\mathtt{L,WL1}})$ in blue solid line, and between $x^{\mathtt{D}_2}$ and the mean-centered water level $x^{\mathtt{AR,WL2}}$ represented by $g(x^{\mathtt{AR,WL2}})$ in red solid line, and (c)-(d) shows the state-dependent regression coefficients that vary based on the values of the reference variables $x^{\mathtt{LL,WL1}}$ and $x^{\mathtt{AR,WL2}}$.

4.4 Conclusion

The state-based regression (SR) method proposed in this chapter enables modeling nonlinear dependency between any two time series within the Bayesian dynamic linear model (BDLM) framework. The SR method employs a Gaussian radial-basis kernel function to obtain the probabilistic weights for a set of regression coefficients associated with the hidden state

representing an independent time series at a particular time instant. Thereafter, a weighted summation of the regression coefficients is carried out in order to infer the predicted regression coefficient. Finally, the interdependent time series is obtained by the product of the hidden state associated with the predicted regression coefficient and the independent time series.

A new `SR` component is developed in the BDLM framework using which the state-dependent regression coefficient and the interdependent time series can be analytically inferred online as hidden states by leveraging the Gaussian multiplicative approximation as described in Section 3.2. This component provides an interpretation of how each nonlinear dependency explains specific patterns in the interdependent time series. Two case studies were conducted using two of the dam's radial displacement datasets obtained from the 16^{th} ICOLD Benchmark 2022 [14, 107]. The results for both the case studies show that the method is capable of accurately identifying the short-term as well as long-term stationary patterns. The method provides interpretable components such that the estimates and the uncertainties for the hidden states involved in the prediction model are available at each instant of time. In comparison to the linear dependency model available in the exiting BDLM framework, the predictive performance for the `SR` method is superior in terms of both test RMSE and log-likelihood.

However, the presence of non-stationary patterns in the residual for the `CB3` dataset suggests that the predictive model would require a regime switching framework to identify such non-stationary patterns in real-time. Moreover, it might be worth investigating additional explanatory variables that can identify specific dam behavior such as creep or creep-relief effects [109]. Furthermore, a key limitation in the proposed method is the need for feature engineering to pre-select the explanatory variables such as the average long-term trend, mean-centered water level, and the moving averages of the air temperature's residuals as shown in the case studies that require domain specific knowledge and also can be time consuming. In addition, the error variances representing the prediction model's aleatory uncertainties need to be obtained using offline gradient-based methods such as Newton-Raphson [25] which are typically computationally demanding.

CHAPTER 5 Approximate Gaussian Variance Inference for Univariate Process Error in the Context of State-Space Models

5.1 Introduction

In the context of state-space models, the process error represents the inaccuracy of perfectly modeling the change in a physical quantity over time. This chapter presents a novel method called the *approximate Gaussian variance inference* (AGVI) that enables analytical Bayesian inference of the variance term σ_W^2 associated with the univariate process error $W \sim \mathcal{N}(w; 0, \sigma_W^2)$. By definition, the expected value of the square of the process error W^2 is equal to the error variance parameter, i.e., $\mathbb{E}[W^2] = \sigma_W^2$. With the approximation that W^2 is Gaussian such that $W^2 \sim \mathcal{N}(w^2; \mathbb{E}[W^2], \text{var}(W^2))$, the error variance parameter is same as the mean parameter for the probability density function (PDF) of W^2. Subsequently, considering that this mean parameter $\mathbb{E}[W^2]$ is a random variable itself, inferring its posterior will be analogous to computing the posterior for the error variance term.

The method proposed utilizes this definition and formulates the relationship between the process error W, the square of process error W^2, and $\mathbb{E}[W^2]$ by leveraging the Gaussian multiplicative approximation (GMA) (see Section 3.2) that provides the exact moments for W^2. Thereafter, the Gaussian conjugate prior (see Section 2.3.1) is used to analytically infer the unknown mean parameter for W^2, i.e., $\mathbb{E}[W^2] = \sigma_W^2$, using closed-form equations. This chapter provides the detailed mathematical formulation and the methodology for applying AGVI in the case of a univariate process error along with two applied examples. The main contributions of this chapter are to

- Provide an analytical Bayesian inference method for performing closed-form online estimation of the univariate process error variance term.
- Provide the mathematical formulation and the methodology of applying AGVI in the Bayesian dynamic linear model (BDLM) framework.
- Validate and verify the performance of the AGVI using synthetic and real data.

5.2 Approximate Gaussian Variance Inference

This section presents the mathematical formulation of the AGVI for inferring the variance parameter σ_W^2 associated with the univariate process error $W \sim \mathcal{N}(w; 0, \sigma_W^2)$ in the context of state-space models.

5.2.1 Problem Formulation

Let us consider an $\mathtt{N}$-dimensional hidden state vector at time $t-1$, $\boldsymbol{x}_{t-1} = [x_1\ x_2\ \cdots x_{\mathtt{N}}]^{\intercal}_{t-1}$, having a Gaussian PDF such that $\boldsymbol{X}_{t-1|t-1} \sim \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1})$ where $\boldsymbol{\mu}_{t-1|t-1} = \mathbb{E}[\boldsymbol{X}_{t-1}|\boldsymbol{y}_{1:t-1}]$ is the prior mean and $\boldsymbol{\Sigma}_{t-1|t-1} = \text{var}(\boldsymbol{X}_{t-1}|\boldsymbol{y}_{1:t-1})$ is the prior covariance matrix. Note that for brevity, the notation $\boldsymbol{X}_{t-1|t-1}$ is used as a shorthand for $\boldsymbol{X}_{t-1}|\boldsymbol{y}_{1:t-1}$. The transition and the observation equations for the Bayesian dynamic linear models (BDLM) (see Section 2.2.1) are given by

$$\begin{array}{rclcrcl} \boldsymbol{x}_t &=& \mathbf{A}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t, & \boldsymbol{w} &:& \boldsymbol{W} &\sim \mathcal{N}(0, \mathbf{Q}), \\ y_t &=& \mathbf{C}\boldsymbol{x}_t + v_t, & v &:& V &\sim \mathcal{N}(0, \mathbf{R}), \end{array} \tag{5.1}$$

where $\mathbf{A}$ is the transition matrix, $\boldsymbol{w}_t = [w_1\ w_2\ \cdots w_{\mathtt{N}}]^{\intercal}_t$ is a vector of process error terms for which $\mathbf{Q}$ is the process error covariance matrix, y_t is the observation, $\mathbf{C}$ is the observation matrix, and v_t is the observation error for which the observation error variance is $\mathbf{R} = \sigma^2_V$. The $\mathbf{A}$ and $\mathbf{Q}$ matrices are constructed by assembling $\mathtt{S}$ specific components given by

$$\begin{aligned} \mathbf{A} &= \text{blkdiag}(\mathbf{A}^1, \mathbf{A}^2, \cdots, \mathbf{A}^{\mathtt{S}}), \\ \mathbf{Q} &= \text{blkdiag}(\mathbf{Q}^1, \mathbf{Q}^2, \cdots, \mathbf{Q}^{\mathtt{S}}), \end{aligned} \tag{5.2}$$

where blkdiag$(\cdot,\cdot)$ refers to block diagonal assembly of the individual components. The $\mathbf{Q}$ matrix in Equation 5.2 can be further described by

$$\mathbf{Q} = \text{blkdiag}(\mathbf{Q}^1(\sigma^2_{W_1}, \Delta t), \mathbf{Q}^2(\sigma^2_{W_2}, \Delta t), \cdots, \mathbf{Q}^{\mathtt{S}}(\sigma^2_{W_{\mathtt{S}}}, \Delta t)), \tag{5.3}$$

where each component $\mathbf{Q}^i(\sigma^2_{W_i}, \Delta t)$ can be represented as a function of the error variance parameter $\sigma^2_{W_i}$ and Δt. For example, consider that the BDLM comprises two generic components, namely the local trend ($\mathtt{LT}$) and the autoregressive ($\mathtt{AR}$), for which the global $\mathbf{Q}$ matrix is

$$\mathbf{Q} = \begin{bmatrix} \sigma^2_{\mathtt{LT}} \cdot \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \sigma^2_{\mathtt{AR}} \end{bmatrix}, \tag{5.4}$$

where the $\mathbf{Q}$ matrices associated with the local trend $\mathbf{Q}^1$ and the autoregressive component $\mathbf{Q}^2$ are

$$\mathbf{Q}^1 = \sigma^2_{\mathtt{LT}} \cdot \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix}, \quad \mathbf{Q}^2 = \sigma^2_{\mathtt{AR}},$$

where $\sigma^2_{\texttt{LT}}$ and $\sigma^2_{\texttt{AR}}$ are the process error variance terms associated with the `LT` and the `AR` components, respectively [21]. Both matrices $\mathbf{Q}^1$ and $\mathbf{Q}^2$ are assembled in a block diagonal arrangement to get the $\mathbf{Q}$ matrix as shown by Equation 5.4. Moreover, for a time series consisting of a single observation variable y_t, it is only possible to infer σ^2_W for one component $\mathbf{Q}^i$, while all other should be either known or 0. This is because only a single unknown variable can be uniquely solved per equation. Hence, for each time series there is one unique process error variance that can be inferred. The next section describes the various steps for performing AGVI in order to obtain the posterior PDF for the error variance parameter.

5.2.2 Methodology

The proposed method considers the process error variance term $\sigma^2_W = \mathbb{E}[W^2]$ as a random variable represented by $\overline{W^2}$ having a Gaussian PDF such that

$$\overline{W^2} \sim \mathcal{N}(\overline{w^2}; \mu^{\overline{W^2}}, (\sigma^{\overline{W^2}})^2), \tag{5.5}$$

where $\mu^{\overline{W^2}}$ and $(\sigma^{\overline{W^2}})^2$ are the hyper-prior mean and variance for $\overline{W^2}$. Using Equation 5.5, the PDF of W can be re-written as

$$W \sim \mathcal{N}(w; 0, \overline{w^2}). \tag{5.6}$$

Hence, the first objective is to obtain the marginal PDF of W such that the random variance can be marginalized out. The following lemmas are invoked to show that the marginal PDF of W can be obtained using the marginal PDF of W^2. The subsequent proposition uses these lemmas to provide the prior predictive PDF for W at a time t.

Lemma 1. *Given that W^2 is approximated as a Gaussian random variable given by $W^2 \sim \mathcal{N}(w^2; \mu^{W^2}, (\sigma^{W^2})^2)$ for which the exact moments are provided by the GMA (see Section 3.2), it can be shown that the PDF of W^2 is dependent only on the mean parameter μ^{W^2} so that*

$$W^2 \sim \mathcal{N}(w^2, \mu^{W^2}, 2(\mu^{W^2})^2)),$$

where the variance term $(\sigma^{W^2})^2$ is equal to $2(\mu^{W^2})^2$. As a result, the PDF $f(w^2|\mu^{W^2}, (\sigma^{W^2})^2)$ can be shown by $f(w^2|\mu^{W^2})$.

Proof. See Appendix C.1. □

Lemma 2. *Given that the parameter μ^{W^2} in $f(w^2|\mu^{W^2})$ is considered as a Gaussian random variable $\overline{W^2} \sim \mathcal{N}(\overline{w^2}; \mu^{\overline{W^2}}, (\sigma^{\overline{W^2}})^2)$, the mean and variance of the prior predictive PDF of*

$W^2_{t|t-1}$ *are given by*

$$\begin{aligned} \mu^{W^2}_{t|t-1} &= \mu^{\overline{W^2}}_{t-1|t-1}, \\ (\sigma^{W^2}_{t|t-1})^2 &= 3(\sigma^{\overline{W^2}}_{t-1|t-1})^2 + 2(\mu^{\overline{W^2}}_{t-1|t-1})^2, \end{aligned}$$

where $\mu^{\overline{W^2}}_{t-1|t-1}$ *and* $(\sigma^{\overline{W^2}}_{t-1|t-1})^2$ *are the prior moments for* $\overline{W^2}_{t-1|t-1}$.

Proof. See Appendix C.2. □

Proposition 1. *Considering that the mean parameter* μ^{W^2} *is itself a random variable* $\overline{W^2}$ *so that*

$$\overline{W^2_{t-1|t-1}} \sim \mathcal{N}(\overline{w^2_{t-1}}\ ; \mu^{\overline{W^2}}_{t-1|t-1}, (\sigma^{\overline{W^2}}_{t-1|t-1})^2),$$

where $\mu^{\overline{W^2}}_{t-1|t-1}$ *and* $(\sigma^{\overline{W^2}}_{t-1|t-1})^2$ *are the hyper-prior mean and variance for* $\overline{W^2_{t-1|t-1}}$, *the error variance* σ^2_W *can be made equal to*

$$\sigma^2_W = \mu^{\overline{W^2}}_{t-1|t-1}. \tag{5.7}$$

Proof. Using Lemmas 1 & 2, and considering the one-to-one relationship between the moments of W and W^2, the prior predictive PDF of $W_{t|t-1}$ can be formulated as

$$f(w_t) = \mathcal{N}(w_t; 0, \mu^{\overline{W^2}}_{t-1|t-1}),$$

where by Lemma 2, the variance of $W_{t|t-1}$ is $\sigma^2_W = \mathbb{E}[W^2_{t|t-1}] = \mu^{\overline{W^2}}_{t-1|t-1}$. □

Using Equation 5.7, the prior PDF for the process error $W_{t-1|t-1}$ is reparameterized as $W_{t-1|t-1} \sim \mathcal{N}(0, \mu^{\overline{W^2}}_{t-1|t-1})$.

The next objective is to perform the prediction step in the filtering procedure (see Section 2.2.2) using the model matrices $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$, and $\mathbf{R}$ defined in Section 5.2.1 and the prior knowledge of σ^2_W. The transition model for $\overline{w^2}$ is $\overline{w^2}_t = \overline{w^2}_{t-1}$, where the hidden state $\overline{w^2}$ is assumed to be constant from $t-1$ to t. Using the prior knowledge for W, the augmented state vector $\boldsymbol{h}_{t-1}$ at any time $t-1$ is given by

$$\boldsymbol{h}_{t-1} = [\boldsymbol{x}\ w]^\intercal_{t-1}. \tag{5.8}$$

The prior predictive PDF of the hidden states $\boldsymbol{H}_{t|t-1}$ is given by

$$\boldsymbol{H}_{t|t-1} \sim \mathcal{N}(\boldsymbol{h}_t; \boldsymbol{\mu}^{\boldsymbol{H}}_{t|t-1}, \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t-1}),$$

where, using Equations 5.1 & 5.7, the mean vector and the covariance matrix are given by

$$\begin{aligned}\boldsymbol{\mu}^{\boldsymbol{H}}_{t|t-1} &= \begin{bmatrix}\mathbf{A}\boldsymbol{\mu}_{t-1|t-1}\\ 0\end{bmatrix}_{t|t-1},\\ \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t-1} &= \begin{bmatrix}\mathbf{A}\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{A}^{\intercal}+\mathbf{Q} & \boldsymbol{\Sigma}^{\boldsymbol{X}W}\\ (\boldsymbol{\Sigma}^{\boldsymbol{X}W})^{\intercal} & \mu^{\overline{W^2}}\end{bmatrix}_{t|t-1}.\end{aligned} \tag{5.9}$$

The covariance term $\boldsymbol{\Sigma}^{\boldsymbol{X}W}_{t|t-1}$ between $\boldsymbol{X}_{t|t-1}$ and $W_{t|t-1}$ in Equation 5.9 is formulated as

$$\begin{aligned}\boldsymbol{\Sigma}^{\boldsymbol{X}W}_{t|t-1} &= \operatorname{cov}(\mathbf{A}\boldsymbol{X}_{t-1|t-1}+\boldsymbol{W}_{t|t-1}, W_{t|t-1}),\\ &= \operatorname{cov}(\boldsymbol{W}, W)_{t|t-1},\end{aligned} \tag{5.10}$$

where $\boldsymbol{W}_{t|t-1}$ is a vector of random variables representing the process error terms in the state vector $\boldsymbol{h}$. Moreover the hidden states $\boldsymbol{X}_{t-1|t-1}$ and the process error $W_{t|t-1}$ are assumed to be independent of each other. The mean and variance of $Y_{t|t-1} \sim \mathcal{N}(y_t; \mu_Y, \sigma^2_Y)$ are given by

$$\begin{aligned}\mu_Y &= \mathbf{C}\boldsymbol{\mu}_{t|t-1}+\cancelto{0}{\mu_V},\\ \sigma^2_Y &= \mathbf{C}\boldsymbol{\Sigma}_{t|t-1}\mathbf{C}^{\intercal}+\sigma^2_V,\end{aligned}$$

given that $\boldsymbol{X}$ and V are assumed to be independent of each other. The covariance term $\boldsymbol{\Sigma}_{\boldsymbol{H}Y}$ between $\boldsymbol{H}_{t|t-1}$ and $Y_{t|t-1}$ is

$$\boldsymbol{\Sigma}_{\boldsymbol{H}Y} = \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t-1}\mathbf{F}^{\intercal}_t,$$

where the observation matrix is $\mathbf{F}_t = [\mathbf{C}\ \ 0]$.

The inference for the parameter σ^2_W requires two update steps; In the first step, the posterior PDF $f(\boldsymbol{h}_t|\boldsymbol{y}_{1:t})$ is estimated using the observation model defined in Equation 5.1 so that

$$f(\boldsymbol{h}_t|\boldsymbol{y}_{1:t}) = \frac{f(\boldsymbol{h}_t, y_t|\boldsymbol{y}_{1:t-1})}{f(y_t|\boldsymbol{y}_{1:t-1})} \approx \mathcal{N}(\boldsymbol{h}_t; \boldsymbol{\mu}^{\boldsymbol{H}}_{t|t}, \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t}), \tag{5.11}$$

which we approximate by a Gaussian distribution with a posterior mean vector $\boldsymbol{\mu}^{\boldsymbol{H}}_{t|t}$ and a covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t}$ that are obtained using the predicted moments provided in Equation 5.9 and the Gaussian conditional equations such that

$$\begin{aligned}\boldsymbol{\mu}^{\boldsymbol{H}}_{t|t} &= \boldsymbol{\mu}^{\boldsymbol{H}}_{t|t-1}+\frac{\boldsymbol{\Sigma}_{\boldsymbol{H}Y}}{\sigma^2_Y}(y_t-\mu_Y),\\ \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t} &= \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t-1}-\frac{\boldsymbol{\Sigma}_{\boldsymbol{H}Y}\cdot\boldsymbol{\Sigma}^{\intercal}_{\boldsymbol{H}Y}}{\sigma^2_Y}.\end{aligned}$$

Now that we have the posterior PDF $f(w_t|\boldsymbol{y}_{1:t})$ from Equation 5.11, we move to the second update step where we use this new information of W at time t to update our current knowledge of $\overline{W^2}$. Figure 5.1 shows the graphical model representing the relationship between the random variables $Y_{t|t-1}$, $\boldsymbol{X}_{t|t-1}$, $W_{t|t-1}$, $W^2_{t|t-1}$ and $\overline{W^2_{t|t-1}}$. Note that while considering $\mu_W = 0$, the first moment of W^2 is equal to the second moment of W (under Lemma 1). In this case, the knowledge of W is fully defined by the knowledge of W^2, which is denoted in Figure 5.1 by an undirected solid line between the nodes W^2 and W. Following the structure depicted in Figure 5.1, the subsequent lemmas are provided for obtaining the posterior knowledge of $\overline{W^2_{t|t}}$.

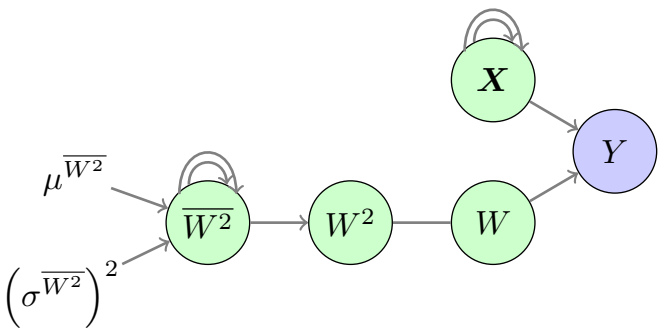


Figure 5.1 Illustration showing the graphical model for the online inference of the error variance parameter. The hidden and observed state variables are denoted by green and violet nodes. The double arrows on the nodes $\boldsymbol{X}$ and $\overline{W^2}$ represent that these variables are learnt recursively over time. For brevity, the subscript $t|t-1$ is dropped from each of the variables.

Lemma 3. *Considering the joint PDF of the random variables $Y_{t|t-1}$, $W^2_{t|t-1}$, and $\overline{W^2_{t|t-1}}$, and marginalizing out W^2 from the joint PDF, the posterior PDF of W^2 can be obtained by the following integral*

$$f(\overline{w^2}_t|\boldsymbol{y}_{1:t}) = \int f(w_t^2|\boldsymbol{y}_{1:t}) \cdot f(\overline{w^2}_t|w_t^2, \boldsymbol{y}_{1:t-1}) dw_t^2.$$

Proof. See Appendix C.3. □

Lemma 4. *The posterior mean and variance of $W^2_{t|t}$ are*

$$\begin{aligned} \mu^{W^2}_{t|t} &= (\mu^W_{t|t})^2 + (\sigma^W_{t|t})^2, \\ (\sigma^{W^2}_{t|t})^2 &= 2(\sigma^W_{t|t})^4 + 4(\sigma^W_{t|t})^2(\mu^W_{t|t})^2. \end{aligned}$$

Proof. See Appendix C.4. □

The Lemmas 3 & 4 are used for proving the following proposition.

Proposition 2. *The posterior mean and variance of* $\overline{W^2_{t|t}} \sim \mathcal{N}(\mu^{\overline{W^2}}_{t|t}, (\sigma^{\overline{W^2}}_{t|t})^2)$ *are given by*

$$\begin{aligned}
\mu^{\overline{W^2}}_{t|t} &= \mu^{\overline{W^2}}_{t|t-1} + K_t(\mu^{W^2}_{t|t} - \mu^{W^2}_{t|t-1}),\\
(\sigma^{\overline{W^2}}_{t|t})^2 &= (\sigma^{\overline{W^2}}_{t|t-1})^2 + K_t^2((\sigma^{W^2}_{t|t})^2 - (\sigma^{W^2}_{t|t-1})^2),\\
K_t &= \frac{(\sigma^{\overline{W^2}}_{t-1|t-1})^2}{(\sigma^{W^2}_{t|t-1})^2}.
\end{aligned}$$

Proof. See Appendix C.5. □

Both the update steps 1 and 2 are employed recursively as observations are collected in order to first estimate the posterior knowledge of W and then use this to update our knowledge of the expected value of W^2, i.e., $\overline{W^2}$, which is a variable that is equal to σ^2_W, the parameter we seek to infer. All the steps performed in a particular time step t are summarized in Algorithm 2.

5.3 Applied Examples

This section presents two case studies illustrating the application of AGVI for inferring the univariate process error variance. For the first case study, the online state estimation of the error variance is provided along with statistical consistency tests for showcasing the optimality of the filter, empirical validation of the uncertainty associated with the error variance estimates, as well as the impact of the $\frac{Q}{R}$ ratio on the posterior mean estimate $\mu_{\mathtt{T|T}}$ of the error variance. Moreover, the performance of the AGVI method is also compared to existing adaptive Kalman filtering (AKF) approaches namely the *indirect correlation method* (ICM) [48], the *adaptive limited memory filter* (ALMF) [63], and the *sliding window variational adaptive Kalman filter* (SWVAKF) [70]. For the second case study, the AGVI method is applied to the traffic-load data for which the predictive performance as well as the computational time is compared with the Newton-Raphson method.

5.3.1 Case Study 1

For this case study, the first-order autoregressive process (see Section 3.3.1) is considered for which the process error variance $\sigma^2_{\mathtt{AR}}$ is unknown and needs to be inferred. Data is simulated

Algorithm 2 One-time step of the proposed AGVI method

Input: $\boldsymbol{\mu}_{t-1|t-1}$, $\boldsymbol{\Sigma}_{t-1|t-1}$, $\mu_{t-1|t-1}^{\overline{W^2}}$, $(\sigma_{t-1|t-1}^{\overline{W^2}})^2$, y_t, $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$, and σ_V^2

Prior knowledge for the error variance parameter:

$\sigma_W^2 = \mu_{t-1|t-1}^{\overline{W^2}}$

Prediction Step:

$$\boldsymbol{\mu}_{t|t-1}^{\boldsymbol{H}} = \begin{bmatrix} \mathbf{A}\boldsymbol{\mu}_{t-1|t-1} \\ 0 \end{bmatrix}_{t|t-1}, \quad \boldsymbol{\Sigma}_{t|t-1}^{\boldsymbol{H}} = \begin{bmatrix} \mathbf{A}\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{A}^\intercal + \mathbf{Q} & \boldsymbol{\Sigma}^{\boldsymbol{X}W} \\ (\boldsymbol{\Sigma}^{\boldsymbol{X}W})^\intercal & \mu^{\overline{W^2}} \end{bmatrix}_{t|t-1},$$

$\mu_Y = \mathbf{C}\boldsymbol{\mu}_{t|t-1}, \quad \sigma_Y^2 = \mathbf{C}\boldsymbol{\Sigma}_{t|t-1}\mathbf{C}^\intercal + \sigma_V^2, \quad \boldsymbol{\Sigma}_{\boldsymbol{H}Y} = \boldsymbol{\Sigma}_{t|t-1}^{\boldsymbol{H}}\mathbf{F}_t^\intercal$

1^{st} Update Step:

$\boldsymbol{\mu}_{t|t}^{\boldsymbol{H}} = \boldsymbol{\mu}_{t|t-1}^{\boldsymbol{H}} + \frac{\boldsymbol{\Sigma}_{\boldsymbol{H}Y}}{\sigma_Y^2}(y_t - \mu_Y), \quad \boldsymbol{\Sigma}_{t|t}^{\boldsymbol{H}} = \boldsymbol{\Sigma}_{t|t-1}^{\boldsymbol{H}} - \frac{\boldsymbol{\Sigma}_{\boldsymbol{H}Y}\cdot\boldsymbol{\Sigma}_{\boldsymbol{H}Y}^\intercal}{\sigma_Y^2}$

Posterior Moments for W^2:

$\mu_{t|t}^{W^2} = (\mu_{t|t}^{W})^2 + (\sigma_{t|t}^{W})^2,$

$(\sigma_{t|t}^{W^2})^2 = 2(\sigma_{t|t}^{W})^4 + 4(\sigma_{t|t}^{W})^2(\mu_{t|t}^{W})^2$

2^{nd} Update Step:

$\mu_{t|t}^{\overline{W^2}} = \mu_{t|t-1}^{\overline{W^2}} + K_t(\mu_{t|t}^{W^2} - \mu_{t|t-1}^{W^2}), \quad (\sigma_{t|t}^{\overline{W^2}})^2 = (\sigma_{t|t-1}^{\overline{W^2}})^2 + K_t^2((\sigma_{t|t}^{W^2})^2 - (\sigma_{t|t-1}^{W^2})^2),$

$K_t = \frac{(\sigma_{t-1|t-1}^{\overline{W^2}})^2}{(\sigma_{t|t-1}^{W^2})^2}$

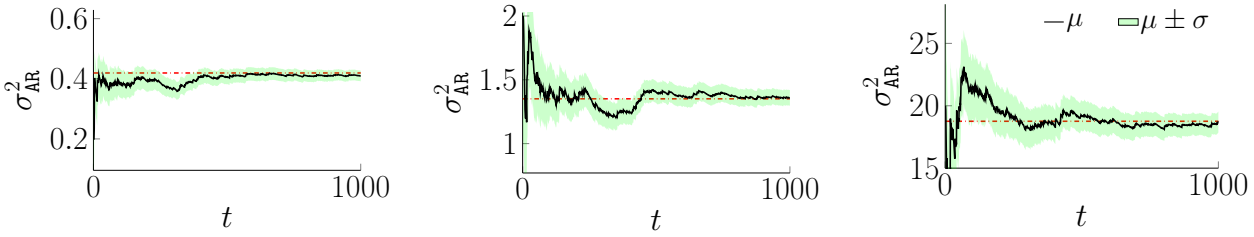
return $\boldsymbol{\mu}_{t|t}$, $\boldsymbol{\Sigma}_{t|t}$, $\mu_{t|t}^{\overline{W^2}}$, and $(\sigma_{t|t}^{\overline{W^2}})^2$

using the parameters $\phi^{\texttt{AR}} = 0.9$, $\sigma_V = 0.01$, and the true value of $\sigma_{\texttt{AR}}^2$ is randomly selected from the prior PDF of $\overline{W_{0|0}^2}$ such that

$$\overline{W_{0|0}^2} \sim \mathcal{N}(\overline{w_0^2}; \mu_{0|0}^{\overline{W^2}}, (\sigma_{0|0}^{\overline{W^2}})^2),$$

where three different true values are generated by considering different prior initialization for the pair of $\{\mu_{0|0}^{\overline{W^2}}, (\sigma_{0|0}^{\overline{W^2}})^2\}$ such that the three cases are (a) $\{\mu_{0|0}^{\overline{W^2}} = 0.2, (\sigma_{0|0}^{\overline{W^2}})^2 = 0.01\}$, (b) $\{\mu_{0|0}^{\overline{W^2}} = 2, (\sigma_{0|0}^{\overline{W^2}})^2 = 1\}$, and (c) $\{\mu_{0|0}^{\overline{W^2}} = 20, (\sigma_{0|0}^{\overline{W^2}})^2 = 100\}$. Figure 5.2 shows the online state estimation of the error variance term for each of the three cases. The true variance value is shown by the red dashed line and the estimated values and their $\pm 1\sigma$ uncertainty

bound are shown in black and green shaded area. These results confirm that the method is able to perform online inference for different magnitudes of the error variance starting from arbitrary initial estimates.

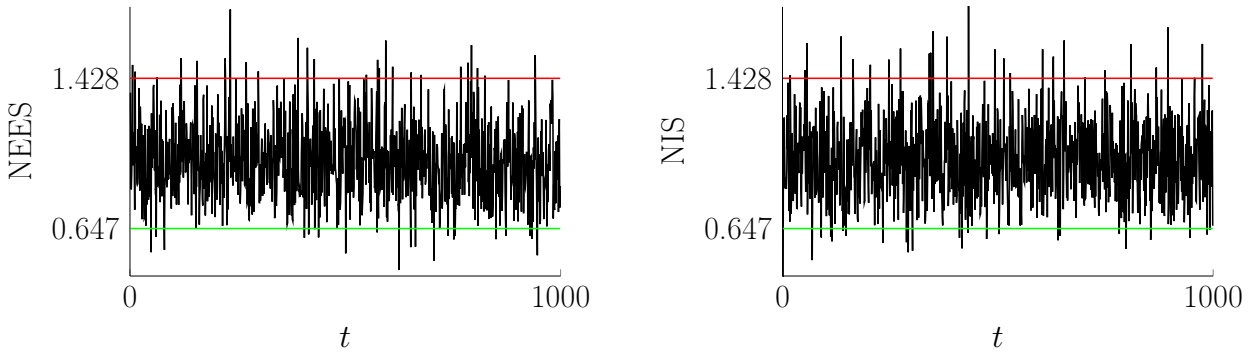


(a) $\mu_{0|0}^{\overline{W^2}} = 0.2, (\sigma_{0|0}^{\overline{W^2}})^2 = 0.01$ (b) $\mu_{0|0}^{\overline{W^2}} = 2, (\sigma_{0|0}^{\overline{W^2}})^2 = 1$ (c) $\mu_{0|0}^{\overline{W^2}} = 20, (\sigma_{0|0}^{\overline{W^2}})^2 = 100$

Figure 5.2 Online estimation of the error variance term for each of the three cases for which the different prior initializations are (a) $\mu_{0|0}^{\overline{W^2}} = 0.2, (\sigma_{0|0}^{\overline{W^2}})^2 = 0.01$, (b) $\mu_{0|0}^{\overline{W^2}} = 2, (\sigma_{0|0}^{\overline{W^2}})^2 = 1$, and (c) $\mu_{0|0}^{\overline{W^2}} = 20, (\sigma_{0|0}^{\overline{W^2}})^2 = 100$. The true $\sigma^2_{\mathtt{AR}}$ value in each case is shown in red dashed line, while the estimated values and their $\pm 1\sigma$ uncertainty bound are shown in black and green shaded area.

The optimality of the filter is evaluated using two chi-square (χ^2) tests that rely on the *normalised estimation error squared* (NEES) and the *normalised innovation error squared* (NIS) values [110]. These tests are conducted using 50 random simulations. Considering a 95% confidence interval (C.I.) and the degrees of freedom $v = \mathtt{X} = \mathtt{Y} = 1$, i.e., the size of the state and observation vector, the two-sided probability region is given by $[\chi^2_{50}(0.025)\ \chi^2_{50}(0.975)] = [32.3\ 71.4]$. By dividing the range by 50, we obtain the probability region for the average NEES and NIS values [0.647 1.428]. Figure 5.3 illustrates an example of the 95% region marked by the green and red lines for both the average NEES and NIS values in case (a). From the definition of the test, there should be approximately 5% of the total number of points outside the 95% region. The length of the time series for the case study is 1000 and hence, approximately 50 points should be outside the region. Table 5.1 presents the average number of points outside the probability region for both the NEES and NIS tests in all three cases where each of the 50 runs are carried out five times in order to compute the average value. The results verify that the filter is optimal and provide consistent estimates for the error variance term, given that the number of points outside the 95% probability region are in accordance to the theoretical results.

In order to check the statistical consistency for the variance of the error variance term, we created 1000 simulated time series where the true values of the error variance in each time series is generated from the prior knowledge of $\overline{W^2_{0|0}}$. Figure 5.4 presents, for each time step,



(a) NEES (b) NIS

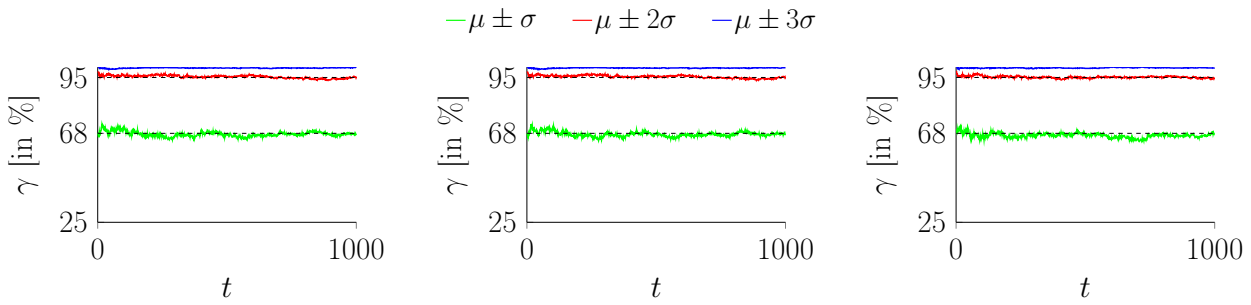
Figure 5.3 Illustration showing the average normalized state estimation error squared (NEES) and the average normalized innovation squared (NIS) for the case study (a) with its 95% probability region given by $[0.647, 1.428]$ is marked by the green and red lines.

Table 5.1 Average number of points outside the 95% probability region for the NEES and NIS values in all the three cases, i.e., (a) $\{\mu_{0|0}^{\overline{W^2}} = 0.2, (\sigma_{0|0}^{\overline{W^2}})^2 = 0.01\}$, (b) $\{\mu_{0|0}^{\overline{W^2}} = 2, (\sigma_{0|0}^{\overline{W^2}})^2 = 1\}$, and (c) $\{\mu_{0|0}^{\overline{W^2}} = 20, (\sigma_{0|0}^{\overline{W^2}})^2 = 100\}$.

Average number of points	case (a)	case (b)	case (c)
NEES	52.2	52.0	50.6
NIS	53.8	53.8	50.6

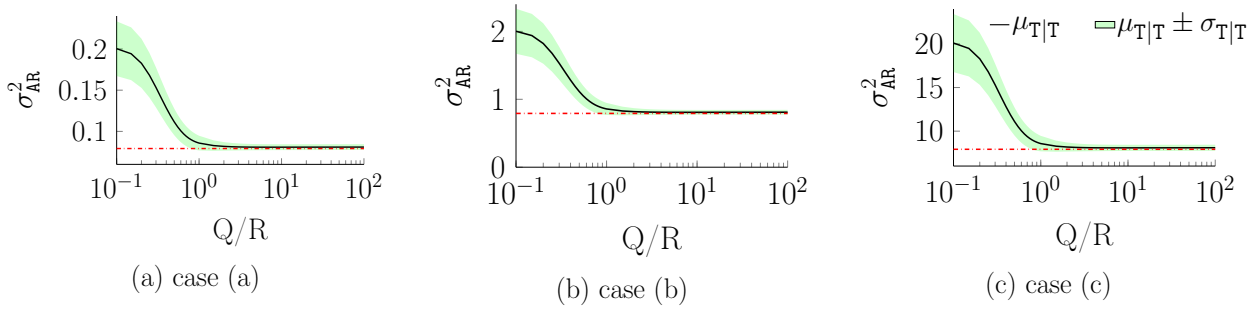
the percentage of realizations (γ) where the true value lies within the confidence interval (C.I.) for 1, 2, and 3 standard deviations from the mean estimate in each of the three case studies. The results in Figure 5.4 show that the γ values match the theoretical C.I. quantities, i.e., $\{68, 95, 99\}\%$, for the Gaussian distribution supporting the hypothesis that the Gaussian PDF for the error variance is adequate at each time step.

Also, we noticed the effect of $\frac{Q}{R} = \frac{\sigma_{\text{AR}}^2}{\sigma_V^2}$ ratio on the estimation accuracy. Figure 5.5 shows the posterior mean estimate $\mu_{\mathtt{T}|\mathtt{T}}$ and the confidence interval $\mu_{\mathtt{T}|\mathtt{T}} \pm \sigma_{\mathtt{T}|\mathtt{T}}$ for the error variance after $\mathtt{T}$ time steps with respect to different $\frac{Q}{R}$ values for the three cases, where $\mathtt{T} = 1000$ is the total length of the time series. The results validate that the AGVI method is accurate for $\frac{Q}{R} \geq 10$. For $1 < \frac{Q}{R} < 10$, the estimates are accurate with small biases in comparison to the true values, whereas for $\frac{Q}{R} < 1$, the estimates are inaccurate with large biases. This phenomenon is explained by the fact that the Kalman gain has a higher value with an increase in the $\frac{Q}{R}$



(a) case (a) (b) case (b) (c) case (c)

Figure 5.4 Empirical consistency check for the variance of the error variance estimate, where γ is the percentage of realizations where the true value lies within the three C.I. for the cases (a) $\{\mu_{0|0}^{\overline{W^2}} = 0.2, (\sigma_{0|0}^{\overline{W^2}})^2 = 0.01\}$, (b) $\{\mu_{0|0}^{\overline{W^2}} = 2, (\sigma_{0|0}^{\overline{W^2}})^2 = 1\}$, and (c) $\{\mu_{0|0}^{\overline{W^2}} = 20, (\sigma_{0|0}^{\overline{W^2}})^2 = 100\}$.



(a) case (a) (b) case (b) (c) case (c)

Figure 5.5 The posterior mean estimate and C.I of the error variance for different values of $\frac{Q}{R}$ for the cases (a) $\{\mu_{0|0}^{\overline{W^2}} = 0.2, (\sigma_{0|0}^{\overline{W^2}})^2 = 0.01\}$, (b) $\{\mu_{0|0}^{\overline{W^2}} = 2, (\sigma_{0|0}^{\overline{W^2}})^2 = 1\}$, and (c) $\{\mu_{0|0}^{\overline{W^2}} = 20, (\sigma_{0|0}^{\overline{W^2}})^2 = 100\}$. Note that the x-axis is in log-scale.

ratio, given that the system is observable [23, 111]; as a result, the Kalman filter put more weight on the measurements. Hence, we obtain a better mean and variance estimate of W by learning from each measurement which in turn provide better estimates for $\overline{W^2}$.

Table 5.2 compares the average RMSE values and computational time obtained using the AGVI method and the existing adaptive Kalman filtering (AKF) methods for the three cases where the true values are (a) $\sigma^2_{\mathtt{AR}} = 0.42$, (b) $\sigma^2_{\mathtt{AR}} = 1.35$, and (c) $\sigma^2_{\mathtt{AR}} = 18.75$. The results are averaged over five independent runs. Each of these AKF methods falls under separate categories as described in Section 2.3.3, where ICM is a correlation method, ALMF is a covariance-matching method (CMM), and the SWVAKF is a Bayesian variational method. The hyper parameters for ICM include the stable Kalman gain ($\mathbf{K}$) and the auto-covariance lag parameter which are fixed to 0.99 and 1, whereas for ALMF, the initial $\mathbf{Q}$ matrix is chosen

Table 5.2 Comparison of the average RMSE values and the computational time (in seconds) obtained from each method in all three cases where the true values are (a) $\sigma^2_{\texttt{AR}} = 0.42$, (b) $\sigma^2_{\texttt{AR}} = 1.35$, and (c) $\sigma^2_{\texttt{AR}} = 18.75$. The results are averaged over five independent runs. Each of the methods are picked from different AKF categories where AGVI and SWVAKF are Bayesian methods whereas ALMF is a covariance-matching method (CMM) and ICM is a correlation method.

Type	Category	Methods	RMSE			Time (s)
			$\sigma^2_{\texttt{AR}} = 0.42$	$\sigma^2_{\texttt{AR}} = 1.35$	$\sigma^2_{\texttt{AR}} = 18.75$	
Online	Bayesian	AGVI	0.014	**0.01**	**0.45**	0.044
Online	Bayesian	SWVAKF	0.060	0.09	2.07	4.640
Offline	CMM	ALMF	**0.010**	0.06	0.85	0.022
Offline	Correlation	ICM	0.018	0.05	0.50	**0.002**

as 1. For the SWVAKF, the same parameters are used as provided in the implementation code [70]. The results show that AGVI outperforms all methods in terms of predictive capacity in case (b) and case (c), while ALMF has better performance in case (a). While being an offline method, the ICM emerges as the fastest method among all others. In comparison to SWVAKF which is both a Bayesian and an online estimation method, AGVI is more than two order of magnitude faster. The offline methods, i.e., the ALMF and the ICM, are faster compared to the Bayesian methods but can only provide point estimates. However, it is necessary to quantify the epistemic uncertainties associated with the error variances in order to understand if the amount of available data is sufficient to estimate them accurately. Furthermore, it is crucial to have accurate and statistically consistent estimates of the posterior mean and variance associated with the error variance terms at every time step when learning sequentially from data. It enables faster convergence to the true values by extracting as much information as possible from each new data point.

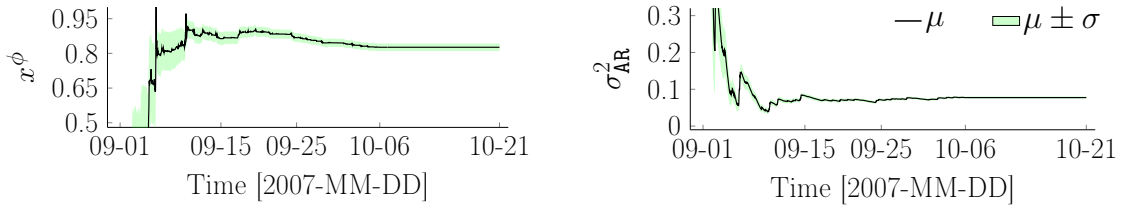
5.3.2 Case Study 2

For this case study, the AGVI method is applied to the traffic-load data as presented in Section 3.3.3 for which the process error variance needs to be inferred. The raw data have 2409 data points which is divided into a *training set* (1649 points) and a *test set* (760). The generic components used for modeling the data presented in Figure 3.5 are the local level (`LL`), two kernel regression components each having 50 and 30 control points to model the periodic patterns with periodicity of 7 days and 1 day respectively, the online autoregressive

component (OAR) to model the residuals, and the double kernel regression (DKR) to model the product of the two periodic patterns.

Figure 5.6 shows an example of the online estimation of both the AR parameter and the error variance $\sigma^2_{\texttt{AR}}$ for which the prior initialization for $\overline{W^2_{0|0}}$ is $\{\mu^{\overline{W^2}}_{0|0} = 4, (\sigma^{\overline{W^2}}_{0|0})^2 = 1\}$. The estimated values are shown by the black solid line and their $\pm 1\sigma$ uncertainty bound is shown by the green shaded region. The results confirm that both parameters from the BDLM framework can now be estimated online simultaneously using the Kalman filter. Table 5.3 compares the AGVI and the Newton-Raphson (NR) method to evaluate the error variance using the average test log-likelihood (LL), the test mean square error (MSE), the optimization time (in s), the training time (in s), and the final estimate of $\sigma_{\texttt{AR}}$. In the case of AGVI, the average values were obtained using different prior initialization for $\overline{W^2_{0|0}}$ such that $\mu^{\overline{W^2}}_{0|0} = \alpha \cdot 1$ and $(\sigma^{\overline{W^2}}_{0|0})^2 = 1$, where α ranges from 1 to 6. Similarly, in order to compute the average estimate using the NR method, the optimization was carried out starting from different initial values such that $\sigma_{\texttt{AR}} = \alpha \cdot 1$. The results show that AGVI has a similar predictive capacity in terms of MSE and a marginally better LL value compared to the NR method. However, the advantage of using AGVI is seen from its computational time, which is approximately 10 times faster than the NR method.

In the case of real data, only the chi-square test relying on normalized innovation errors squared (NIS) values can be employed to check the optimality of the filter. The two-sided probability region for a 95% C.I. having one degree of freedom, i.e., the size of the observation vector Y, is [0 5.02]. Considering that the total length of the training set is 1649, the theoretical 5% value of the number of acceptable points outside the 95% C.I. is 82.45. Using different prior initialization, the average number of points outside the probability region is



(a) x^ϕ (b) $\sigma^2_{\texttt{AR}}$

Figure 5.6 Illustration showing the online estimation of both the AR parameter and the error variance $\sigma^2_{\texttt{AR}}$ using the prior initialization $\{\mu^{\overline{W^2}}_{0|0} = 4, (\sigma^{\overline{W^2}}_{0|0})^2 = 1\}$. The estimated values are shown by the black solid line and their $\pm 1\sigma$ uncertainty bound is shown by the green shaded region.

Table 5.3 Comparison of the average test mean square error (MSE), test log-likelihood (LL), optimization time (in s), training time (in seconds), and the final estimate of $\sigma_{\texttt{AR}}$ using the AGVI and Newton-Raphson (NR) for the traffic-load dataset.

method	MSE	LL	Optimization Time (s)	Training Time (s)	$\sigma_{\texttt{AR}}$
AGVI	**0.302**	**−610.47**	0	4.39	0.282
NR	0.307	−620.64	49.78	**4.31**	0.278

71 which is acceptable given the theoretical results. Hence, both the case studies show that AGVI is a computationally efficient approach that provides competitive predictive capacity and statistically consistent estimates for the error variance.

5.4 Discussion

In addition to the two case studies presented in this chapter, the AGVI method has already been applied in Blanche Laurent's master thesis [112] to a network-scale framework that is used for monitoring the condition of transportation infrastructures over time. In the framework, visual inspections are used to quantify the structural condition. These inspections have been carried out by different individuals such that the inspections are subjective in nature given that each inspector has a different capacity to perform the evaluation task. Hence, it is critical to evaluate each of the inspector's variance. A *gradient-based framework* [27] was considered to estimate the variance of these inspectors. This process was computationally demanding due to the large number of inspectors across the network of bridges. The AGVI method successfully replaced the gradient-based framework with an analytical framework to perform joint estimation for over 250 variance parameters associated with the inspectors. The results show that the AGVI-based framework has reduced the computational time from 33 hours to 20 minutes while maintaining a comparable predictive capacity. This independent case study is another example showcasing the capability of the AGVI method for scaling up existing models for practical engineering applications.

The AGVI method has promising results and can provide an efficient way of reducing the computational time for estimating parameters associated with aleatory uncertainty in probabilistic models. However, one key limitation of AGVI that is not addressed in this thesis is the uncertainty associated with the error variance is not considered as σ_W^2 is shown to be equal to $\mu^{\overline{W^2}}$, i.e., the expected value of $\overline{W^2}$, as provided by Proposition 1. For example, if

we consider that W is a Gaussian random variable such that $W \sim \mathcal{N}(w; 0, \overline{W^2})$ for which the variance parameter is a random variable shown by $\overline{W^2} \sim \mathcal{N}(\overline{w^2}; 10, 4)$, it can be verified by sampling from the PDF of W that the sample variance is 10 thereby ignoring the uncertainty associated with the variance term. This limitation is attributed to the Gaussian assumption for the variance term, for which inverse gamma PDF is the theoretical distribution. On the other hand, closed-form inference from uncertain observations is not possible while starting from inverse gamma priors for the error variance. A future work that could further improve the AGVI method would be to consider inverse gamma priors for the error variance while maintaining the analytical tractability during inference.

5.5 Conclusion

The approximate Gaussian variance inference (AGVI) method proposed in this chapter is an analytically tractable online Bayesian inference method for state-space models that provides: 1) analytical inference of the univariate process error variance as a hidden state, 2) accurate as well as statistically consistent estimates of the mean and the variance of the error variance term at each time step, and 3) higher computational speed compared to the offline gradient-based optimization approaches. The method was validated and verified with both simulated and SHM-based real data.

The case study 1 shows the application of the AGVI method for obtaining the univariate process error variance starting from different prior initialization. The statistical consistency tests verify that the filter is optimal and that the AGVI method provide consistent estimates for the mean as well as the variance of the error variance term. In comparison to the existing adaptive Kalman filtering methods, the AGVI method provides better predictive capacity in two out of the three cases. In comparison to SWVAKF which is both a Bayesian and an online estimation method, it is more than two order of magnitude faster. The offline methods, i.e., the ALMF and the ICM, are faster compared to the Bayesian methods but can only provide point estimates which limits their applicability in decision-making tasks.

The case study 2 shows that with AGVI the Bayesian dynamic linear models can bypass offline optimization techniques such as the Newton-Raphson (NR) method for obtaining the error variance estimates at an order of magnitude faster. Moreover, AGVI provides both the mean as well as variance that gives us not only the estimate but also the associated uncertainty at any instant of time. Both case studies demonstrate that AGVI is a computationally efficient approach that provides a competitive predictive capacity and statistically consistent estimates for the error variance.

Nevertheless, note that the observation error variance needs to be known and the $\frac{\mathbf{Q}}{\mathbf{R}}$ ratio plays an influential role in the estimation accuracy of AGVI. The independent case study reviewed in Section 5.4 shows that the AGVI method is applicable to existing models for quantifying uncertainty associated with observations, which in this case, was visual inspections. The results show that by leveraging the AGVI method, the computational time of the framework could be reduced from 33 hours to 20 minutes while maintaining a comparable predictive capacity.

One key limitation not addressed in this thesis is that the uncertainty associated with the error variance is not considered as σ_W^2 is shown to be equal to $\mu^{\overline{W^2}}$. A future work to improve the AGVI method would be to consider inverse gamma priors for the error variance so that the uncertainty of the error variance is taken into account for estimating the variance parameter of the prior predictive PDF of the process error W while maintaining analytical tractability.

CHAPTER 6 Approximate Gaussian Variance Inference for Multivariate Process Errors

6.1 Introduction

For a network of sensors, the process errors associated with modeling different physical quantities can be correlated. Hence, it is often incorrect to assume a diagonal process error covariance matrix ($\mathbf{Q}$), as the model does require a full $\mathbf{Q}$ matrix with not only the error variance terms but also the covariance between the error terms. However the factor limiting the inference of higher dimensional $\mathbf{Q}$ matrix is related to numerical instabilities that occur as the matrix becomes non-positive semi-definite (non-PSD). This chapter extends the mathematical formulation for the approximate Gaussian variance inference (AGVI) described in Chapter 5 for the online inference of the multivariate process error variance parameters. Using AGVI for the multivariate observation model, one error variance term σ_W^2 is inferred for each observation equation along with the covariance for each pair of error terms. The chapter also provides a closed-form square-root filtering technique using the Cholesky decomposition such that the $\mathbf{Q}$ matrix is always PSD. Finally, the chapter provides the methodology and two applied examples where the first case study compares the performance of AGVI with the existing adaptive Kalman filtering (AKF) approaches and the second case study shows its application on real datasets from a concrete dam. The main contributions of this chapter are to

- Provide the mathematical formulation and the methodology for applying AGVI in the case of multivariate process errors.
- Provide a closed-form square-root filtering technique using the Cholesky decomposition.
- Compare the performance of AGVI with the existing AKF approaches and shows its application on real datasets from a concrete dam.

6.2 Multivariate Process Errors

This section presents the mathematical formulation of the AGVI for inferring multiple process error variance parameters.

6.2.1 Problem Formulation

Let us consider $\mathtt{D}$ observed time series for which the global state vector is $\boldsymbol{x}_t = [\boldsymbol{x}_t^1\ \boldsymbol{x}_t^2\ \cdots\ \boldsymbol{x}_t^{\mathtt{D}}]^\intercal$, where $\boldsymbol{x}_t^j,\ \forall j \in \{1, 2, \cdots, \mathtt{D}\}$ refers to the concatenation of all $\mathtt{S}_j$ generic components for the j^{th} time series. Similarly, the vector of correlated process errors is assembled following $\boldsymbol{w}_t = [\boldsymbol{w}_t^1\ \boldsymbol{w}_t^2\ \cdots \boldsymbol{w}_t^{\mathtt{D}}]^\intercal$. The global transition, observation, process error covariance, and observation error covariance matrices are assembled block diagonally as

$$\begin{aligned}
\mathbf{A} &= \text{blkdiag}[\mathbf{A}^1,\ \mathbf{A}^2,\ \cdots, \mathbf{A}^{\mathtt{D}}], \\
\mathbf{C} &= \text{blkdiag}[\mathbf{C}^1,\ \mathbf{C}^2,\ \cdots, \mathbf{C}^{\mathtt{D}}], \\
\mathbf{Q} &= \text{blkdiag}[\mathbf{Q}^1,\ \mathbf{Q}^2,\ \cdots, \mathbf{Q}^{\mathtt{D}}], \\
\mathbf{R} &= \text{blkdiag}[\mathbf{R}^1,\ \mathbf{R}^2,\ \cdots, \mathbf{R}^{\mathtt{D}}].
\end{aligned}$$

Covariance matrices $\text{cov}(\boldsymbol{W}^k, \boldsymbol{W}^n)$ exists between the process errors $\boldsymbol{W}^k$ and $\boldsymbol{W}^n$ of the k^{th} and n^{th} time series respectively, where $k, n \in \{1, 2, \cdots, \mathtt{D}\}$. The process error covariance matrix $\mathbf{Q}$ can be reformulated as follows

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}^1 & \mathbf{Q}^{1,2} & \cdots & \mathbf{Q}^{1,\mathtt{D}} \\ \vdots & \mathbf{Q}^2 & \cdots & \mathbf{Q}^{2,\mathtt{D}} \\ \vdots & \cdots & \ddots & \vdots \\ \text{sym.} & \cdots & \cdots & \mathbf{Q}^{\mathtt{D}} \end{bmatrix}, \tag{6.1}$$

where the covariance term $\text{cov}(\boldsymbol{W}^k, \boldsymbol{W}^n)$ is represented by $\mathbf{Q}^{k,n}$. The sub-matrices within the matrix $\mathbf{Q}^{k,n}$ in Equation 6.1 are themselves represented by $\text{cov}(\boldsymbol{W}^{jk}, \boldsymbol{W}^{mn}) = \mathbf{Q}^{jk,mn}$, where $j \in \{1, 2, \cdots, \mathtt{S}_j\}$ and $m \in \{1, 2, \cdots, \mathtt{S}_m\}$ are the j^{th} and m^{th} component of the k^{th} and n^{th} time series, respectively. As described in Section 5.2.1, each of the sub-matrices $\mathbf{Q}^{jk,mn}(\sigma^2_{jk}, \sigma^2_{mn}, \Delta t)$ can be represented as a function of the error variance parameters σ^2_{jk}, σ^2_{mn}, and Δt. Moreover, each of the elements within the sub-matrix $\mathbf{Q}^{jk,mn}$ is given by $\text{cov}(W^{ijk}, W^{lmn})$, which provides the covariance between the i^{th} process error term of the j^{th} component in the k^{th} time series, W^{ijk}, and the l^{th} process error term of the m^{th} component in the n^{th} time series, W_{lmn}. For example, let us consider two time series each modeled using a local trend ($\mathtt{LT}$) component as described in Section 5.2.1. The global $\mathbf{Q}$ matrix is assembled block diagonally such that

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}^1 & \mathbf{Q}^{1,2} \\ \mathbf{Q}^{2,1} & \mathbf{Q}^2 \end{bmatrix},$$

where $\mathbf{Q}^1$ and $\mathbf{Q}^2$ are the process error covariance matrices associated with their individual local trend components and $\mathbf{Q}^{1,2}$ is the cross-covariance matrix between the process errors of the two time series. Each of these covariance matrices are defined as follows: $\mathbf{Q}^1 = \sigma^2_{\mathtt{LT}_1} \cdot \mathbf{J}$, $\mathbf{Q}^2 = \sigma^2_{\mathtt{LT}_2} \cdot \mathbf{J}$, and $\mathbf{Q}^{1,2} = \sigma_{\mathtt{LT}_{12}} \cdot \mathbf{J}$, where $\sigma^2_{\mathtt{LT}_1}$, and $\sigma^2_{\mathtt{LT}_2}$ are the two error variance terms for each of the $\mathtt{LT}$ component, $\sigma_{\mathtt{LT}_{12}}$ is the covariance term between the two process errors $W^{\mathtt{LT}_1}$ and $W^{\mathtt{LT}_2}$. For a constant acceleration kinematic model [23,110], the matrix $\mathbf{J}$ is defined such that

$$\mathbf{J} = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix}.$$

Hence, for this case the terms to be inferred are: $\sigma^2_{\mathtt{LT}_1}$, $\sigma^2_{\mathtt{LT}_2}$, and $\sigma_{\mathtt{LT}_{12}}$. Similarly, for multiple time series, the goal is to infer one error variance term per time series along with the covariance terms for each pair of process error terms.

6.2.2 Methodology

Let us consider the multivariate process error term $\boldsymbol{w} = [w^1\ w^2\ \cdots\ w^i\ \cdots\ w^{\mathtt{D}}]^\intercal$, where $w^i,\ \forall i \in \{1,\ 2,\ \cdots, \mathtt{D}\}$ represents the one process error term for the i^{th} time series for which the variance term has to be inferred. Given that the expected value of W is zero, the covariance term between the i^{th} and j^{th} process error is

$$\mathrm{cov}(W^i, W^j) = \mathbb{E}[W^i W^j] - \cancelto{0}{\mathbb{E}[W^i]}\cancelto{0}{\mathbb{E}[W^j]} = \mathbb{E}[W^i W^j]. \tag{6.2}$$

Using Equation 6.2, the covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{W}}$ is given by

$$\boldsymbol{\Sigma}^{\boldsymbol{W}} = \begin{bmatrix} \mathbb{E}[(W^1)^2] & \mathbb{E}[W^1 W^2] & \cdots & \mathbb{E}[W^1 W^{\mathtt{D}}] \\ \vdots & \mathbb{E}[(W^2)^2] & \cdots & \mathbb{E}[W^2 W^{\mathtt{D}}] \\ \vdots & \cdots & \ddots & \vdots \\ \text{sym.} & \cdots & \cdots & \mathbb{E}[(W^{\mathtt{D}})^2] \end{bmatrix}, \tag{6.3}$$

where $\mathrm{var}(W^i) = \mathbb{E}[(W^i)^2]$ is the error variance for the i^{th} time series, and $\mathrm{cov}(W^i, W^j) = \mathbb{E}[W^i W^j]$ is the covariance term between the two process errors for the i^{th} and j^{th} time series. Similarly to the univariate process error, let us consider the approximation that each of the product terms $W^i W^j$ is a Gaussian random variable such that

$$W^i W^j \ \sim\ \mathcal{N}(w^i w^j; \mu^{W^i W^j}, (\sigma^{W^i W^j})^2), \tag{6.4}$$

where $\mathbb{E}[W^i W^j] = \mu^{W^i W^j}$ is the mean parameter and $\text{var}(W^i W^j) = (\sigma^{W^i W^j})^2$ is the variance. For $\mathtt{D}$ time series, there are a total of $\mathtt{D} \cdot (\mathtt{D}+1)$ product terms which are represented by the random vector $\boldsymbol{w^p} = [(w^1)^2 \ (w^2)^2 \ \cdots \ w^i w^j \ \cdots \ w^{\mathtt{D}} w^{\mathtt{D\text{-}1}}]^\intercal$ such that

$$\boldsymbol{W^p} \sim \mathcal{N}(\boldsymbol{w^p}; \boldsymbol{\mu^{W^p}}, \boldsymbol{\Sigma^{W^p}}), \tag{6.5}$$

where using Equation 6.4, the mean vector of $\boldsymbol{W^p}$ is given by

$$\boldsymbol{\mu^{W^p}} = \left[\mu^{(W^1)^2} \ \mu^{(W^2)^2} \ \cdots \mu^{(W^{\mathtt{D}})^2} \ \mu^{W^1 W^2} \ \cdots \ \mu^{W^{\mathtt{D\text{-}1}} W^{\mathtt{D}}}\right]^\intercal_{k \times 1}. \tag{6.6}$$

Similarly to Lemma 1, the covariance matrix $\boldsymbol{\Sigma^{W^p}}$ can be obtained in terms of the mean parameters in $\boldsymbol{\mu^{W^p}}$ such that

$$\boldsymbol{\Sigma^{W^p}} = \begin{bmatrix} 2(\mu^{(W^1)^2})^2 & 2(\mu^{W^1 W^2})^2 & \cdots & 2\mu^{W^1 W^{\mathtt{D}-1}} \mu^{W^1 W^{\mathtt{D}}} \\ \vdots & 2(\mu^{(W^2)^2})^2 & \cdots & 2\mu^{W^2 W^{\mathtt{D}-1}} \mu^{W^1 W^{\mathtt{D}}} \\ \vdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ \text{sym.} & \cdots & \cdots & (\mu^{(W^{\mathtt{D}-1})^2})^2 (\mu^{(W^{\mathtt{D}})^2})^2 + (\mu^{W^{\mathtt{D\text{-}1}} W^{\mathtt{D}}})^2 \end{bmatrix}_{k \times k},$$

where the variance $\text{var}(W^i W^j)$ and the covariance $\text{cov}(W^i W^j, W^k W^m)$ terms for the product of the errors $\ \forall i, j, k, m \in \{1, \ 2, \ \cdots, \mathtt{D}\}$ are obtained using the GMA equations defined in Section 3.2. The mean vector $\boldsymbol{\mu^{W^p}}$ defined in Equation 6.6 is considered to be random with a Gaussian PDF given by

$$\overline{\boldsymbol{W^p}} \sim \mathcal{N}(\overline{\boldsymbol{w^p}}; \boldsymbol{\mu}^{\overline{\boldsymbol{W^p}}}, \boldsymbol{\Sigma}^{\overline{\boldsymbol{W^p}}}). \tag{6.7}$$

where the vector $\overline{\boldsymbol{w^p}}$ is

$$\overline{\boldsymbol{w^p}} = \left[\overline{(w^1)^2} \ \overline{(w^2)^2} \ \cdots \ \overline{(w^{\mathtt{D}})^2} \ \overline{w^1 w^2} \ \cdots \ \overline{w^{\mathtt{D\text{-}1}} w^{\mathtt{D}}}\right]^\intercal.$$

The mean vector and the covariance matrix of $\overline{\boldsymbol{W}^{\boldsymbol{p}}}$ are

$$\begin{aligned}
\boldsymbol{\mu}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}} &= \left[\mu^{\overline{(W^1)^2}}\ \mu^{\overline{(W^2)^2}}\ \cdots\ \mu^{\overline{W^{\mathtt{D}\text{-}1}W^{\mathtt{D}}}}\right]^{\mathsf{T}}_{k\times 1}, \\
\boldsymbol{\Sigma}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}} &= \begin{bmatrix}
(\sigma^{\overline{(W^1)^2}})^2 & 0 & \cdots & 0 \\
\vdots & (\sigma^{\overline{(W^2)^2}})^2 & \cdots & 0 \\
\vdots & \cdots & \cdots & 0 \\
\vdots & \cdots & \ddots & \vdots \\
\text{sym.} & \cdots & \cdots & (\sigma^{\overline{W^{\mathtt{D}\text{-}1}W^{\mathtt{D}}}})^2
\end{bmatrix}_{k\times k},
\end{aligned}$$

where the random variables in $\overline{\boldsymbol{W}^{\boldsymbol{p}}}$ are assumed to be independent from each other as shown by the covariance matrix $\boldsymbol{\Sigma}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}}$ where the off-diagonal terms are zero.

Using the hyper-prior $\overline{\boldsymbol{W}^{\boldsymbol{p}}}$ defined in Equation 6.7, the first objective is to obtain the covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{W}}$ defined in Equation 6.3 by obtaining the prior predictive PDF of $\boldsymbol{W}^{\boldsymbol{p}}_{t|t-1}$ as provided by the following lemma and proposition.

Lemma 5. *Using the transition model $\boldsymbol{w}^{\boldsymbol{p}}_t = \boldsymbol{w}^{\boldsymbol{p}}_{t-1}$, the prior predictive PDF of $\boldsymbol{W}^{\boldsymbol{p}}_{t|t-1}$ is given by*

$$\boldsymbol{W}^{\boldsymbol{p}}_{t|t-1} \sim \mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{W}^{\boldsymbol{p}}}_{t|t-1}, \boldsymbol{\Sigma}^{\boldsymbol{W}^{\boldsymbol{p}}}_{t|t-1}),$$

where the mean terms in $\boldsymbol{\mu}^{\boldsymbol{W}^{\boldsymbol{p}}}_{t|t-1}$, and the variance and covariance terms in $\boldsymbol{\Sigma}^{\boldsymbol{W}^{\boldsymbol{p}}}_{t|t-1}$ are given by

$$\begin{aligned}
\mathbb{E}[W^iW^j] &= \mu^{\overline{W^iW^j}}, \\
\operatorname{var}((W^i)^2) &= 3(\sigma^{\overline{(W^i)^2}})^2 + 2(\mu^{\overline{(W^i)^2}})^2, \\
\operatorname{var}(W^iW^j) &= (\sigma^{\overline{W^iW^j}})^2 \\
&\quad + \frac{(\mu^{\overline{W^iW^j}})^2}{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\mu^{\overline{W^iW^j}})^2} \cdot (\sigma^{\overline{W^iW^j}})^2 \\
&\quad + \mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\mu^{\overline{W^iW^j}})^2, \\
\operatorname{cov}(W^iW^j, W^lW^m) &= \mu^{\overline{W^iW^l}}\mu^{\overline{W^jW^m}} + \mu^{\overline{W^iW^m}}\mu^{\overline{W^jW^l}}.
\end{aligned}$$

Proof. See Appendix D.1. □

Proposition 3. *The prior predictive PDF of $\boldsymbol{W}$ has a zero mean vector and covariance*

matrix $\boldsymbol{\Sigma}^{\boldsymbol{W}}_{t|t-1}$ *defined by*

$$\boldsymbol{\Sigma}^{\boldsymbol{W}}_{t|t-1} = \begin{bmatrix} \mu^{\overline{(W^1)^2}} & \mu^{\overline{W^1W^2}} & \cdots & \mu^{\overline{W^1W^{\mathtt{D}}}} \\ \vdots & \mu^{\overline{(W^2)^2}} & \cdots & \mu^{\overline{W^2W^{\mathtt{D}}}} \\ \vdots & \cdots & \ddots & \vdots \\ sym. & \cdots & \cdots & \mu^{\overline{(W^{\mathtt{D}})^2}} \end{bmatrix}_{t|t-1}. \tag{6.8}$$

Proof. Using Lemma 5, the covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{W}}$ for the prior predictive PDF of $\boldsymbol{W}$ is obtained by substituting the terms $\mathbb{E}[W^iW^j]$ in Equation 6.3 by the mean parameters of $\overline{\boldsymbol{W}^{\boldsymbol{p}}}$, i.e., $\mu^{\overline{W^iW^j}}$. □

In order to maintain positive semi-definiteness of $\boldsymbol{\Sigma}^{\boldsymbol{W}}_{t|t-1}$ shown by Equation 6.8, the prior information is built from a random vector $\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}$ that is defined in a Cholesky space as shown by the following lemma.

Lemma 6. *Any* ij^{th} *element of* $\boldsymbol{\Sigma}^{\boldsymbol{W}}$ *is obtained such that*

$$\mu^{\overline{W^iW^j}} = \mathbb{E}\left[\sum_{k=1}^{\mathtt{D}} L_{jk}L_{ki}\right],$$

where all elements of $\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}$ *are assumed to be Gaussian,* $L_{ij} \sim \mathcal{N}(\mu_{L_{ij}}, \sigma^2_{L_{ij}})$*, and the expectation of the product terms are obtained using the GMA equations. Moreover, any covariance term between the random vectors* $\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}$ *and* $\overline{\boldsymbol{W}^{\boldsymbol{p}}}$ *given by* $\boldsymbol{\Sigma}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}\overline{\boldsymbol{W}^{\boldsymbol{p}}}}_{t|t-1}$*, can be shown as*

$$\mathrm{cov}(L_{ij}, \overline{W^iW^j}) = \mathrm{cov}(L_{ij}, \sum_{k=1}^{\mathtt{D}} L_{jk}L_{ki}).$$

Proof. See Appendix D.2. □

Using the prior predictive PDF of $\boldsymbol{W}$, the next objective is to perform the prediction step. Let us consider the augmented vector of hidden states $\boldsymbol{h}_{t-1} = [\boldsymbol{x}^{\intercal}_{t-1}\ \boldsymbol{w}^{\intercal}_{t-1}]^{\intercal}$ such that the PDF of $\boldsymbol{H}_{t|t-1} \sim \mathcal{N}(\boldsymbol{h}_t, \boldsymbol{\mu}^{\boldsymbol{H}}_{t|t-1}, \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t-1})$ has a mean vector $\boldsymbol{\mu}^{\boldsymbol{H}}_{t|t-1}$ and a covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t-1}$ defined by

$$\boldsymbol{\mu}^{\boldsymbol{H}}_{t|t-1} = \left[\boldsymbol{\mu}^{\intercal}_{t|t-1}\ \mathbf{0}\right]^{\intercal}, \tag{6.9}$$

$$\boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t-1} = \begin{bmatrix} \mathbf{A}\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{A}^{\intercal} + \mathbf{Q} & \boldsymbol{\Sigma}^{\boldsymbol{XW}} \\ (\boldsymbol{\Sigma}^{\boldsymbol{XW}})^{\intercal} & \boldsymbol{\Sigma}^{\boldsymbol{W}} \end{bmatrix}_{t|t-1}, \tag{6.10}$$

where the covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{W}}$ defined in Equation 6.8 is obtained using the prior knowledge of $\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}$ defined in the Cholesky space as stated in Lemma 6. Similarly to Equation 5.10, the covariance matrix between $\boldsymbol{X}$ and $\boldsymbol{W}$ is given by

$$\begin{aligned}
\boldsymbol{\Sigma}^{\boldsymbol{XW}}_{t|t-1} &= \text{cov}(\boldsymbol{X}, \boldsymbol{W})_{t|t-1} \\
&= \text{cov}(\boldsymbol{X}, [W^1\ W^2\ \cdots\ W^{\mathtt{D}}]^\intercal)_{t|t-1}, \\
&= \text{cov}(\mathbf{A}\boldsymbol{X}_{t-1|t-1} + \boldsymbol{W}_{t|t-1}, [W^1\ W^2\ \cdots\ W^{\mathtt{D}}]^\intercal_{t|t-1}), \\
&= \text{cov}(\boldsymbol{W}, [W^1\ W^2\ \cdots\ W^{\mathtt{D}}]^\intercal)_{t|t-1},
\end{aligned}$$

where $\boldsymbol{W}_{t|t-1} = [W^1\ W^2\ \cdots\ W^{\mathtt{D}}]^\intercal_{t|t-1}$, is a vector of random variables that includes one process error term W from each of the $\mathtt{D}$ time series.

The inference for the covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{W}}$ requires two update steps. Using the Gaussian conditional equations defined in Section 5.2.2, the first update step is performed to obtain the posterior PDF of $\boldsymbol{H}$ so that

$$\boldsymbol{H}_{t|t} \sim \mathcal{N}(\boldsymbol{h}_t, \boldsymbol{\mu}^{\boldsymbol{H}}_{t|t}, \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t}). \tag{6.11}$$

We now move to the second update step where we use the posterior PDF $f(\boldsymbol{w}_t|\boldsymbol{y}_{1:t})$ obtained from Equation 6.11, and the GMA equations to obtain the posterior PDF $f(\boldsymbol{w}^{\boldsymbol{p}}_t|\boldsymbol{y}_{1:t})$ such that

$$f(\boldsymbol{w}^{\boldsymbol{p}}_t|\boldsymbol{y}_{1:t}) = \mathcal{N}(\boldsymbol{w}^{\boldsymbol{p}}_t; \boldsymbol{\mu}^{\boldsymbol{W^p}}_{t|t}, \boldsymbol{\Sigma}^{\boldsymbol{W^p}}_{t|t}).$$

The posterior PDF of $\overline{\boldsymbol{W^p}}$ is defined using the following lemma.

Lemma 7. *The posterior mean, variance and covariance terms of* $\overline{\boldsymbol{W^p}}$ *are*

$$\begin{aligned}
\boldsymbol{\mu}^{\overline{\boldsymbol{W^p}}}_{t|t} &= \boldsymbol{\mu}^{\overline{\boldsymbol{W^p}}}_{t|t-1} + \boldsymbol{K}_t(\boldsymbol{\mu}^{\boldsymbol{W^p}}_{t|t} - \boldsymbol{\mu}^{\boldsymbol{W^p}}_{t|t-1}), \\
\boldsymbol{\Sigma}^{\overline{\boldsymbol{W^p}}}_{t|t} &= \boldsymbol{\Sigma}^{\overline{\boldsymbol{W^p}}}_{t|t-1} + \boldsymbol{K}_t(\boldsymbol{\Sigma}^{\boldsymbol{W^p}}_{t|t} - \boldsymbol{\Sigma}^{\boldsymbol{W^p}}_{t|t-1})\boldsymbol{K}^\intercal_t, \\
\boldsymbol{K}_t &= \boldsymbol{\Sigma}^{\boldsymbol{W^p}\overline{\boldsymbol{W^p}}}_{t|t-1}(\boldsymbol{\Sigma}^{\boldsymbol{W^p}}_{t|t-1})^{-1}, \\
\boldsymbol{\Sigma}^{\boldsymbol{W^p}\overline{\boldsymbol{W^p}}}_{t|t-1} &= \boldsymbol{\Sigma}^{\overline{\boldsymbol{W^p}}}_{t|t-1}.
\end{aligned}$$

Proof. See Appendix D.3. □

Using the updated knowledge of $\overline{\boldsymbol{W^p}}$ in Lemma 7, the posterior moments for $\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}$ in the Cholesky space is defined using the following proposition.

Proposition 4. *The posterior moments of* $\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}$ *are*

$$\begin{aligned}
\boldsymbol{\mu}_{t|t}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}} &= \boldsymbol{\mu}_{t|t-1}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}} + \boldsymbol{K}_t^{\boldsymbol{L}}(\boldsymbol{\mu}_{t|t}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}} - \boldsymbol{\mu}_{t|t-1}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}}), \\
\boldsymbol{\Sigma}_{t|t}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}} &= \boldsymbol{\Sigma}_{t|t-1}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}} + \boldsymbol{K}_t^{\boldsymbol{L}}(\boldsymbol{\Sigma}_{t|t}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}} - \boldsymbol{\Sigma}_{t|t-1}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}})(\boldsymbol{K}_t^{\boldsymbol{L}})^{\intercal}, \\
\boldsymbol{K}_t^{\boldsymbol{L}} &= \boldsymbol{\Sigma}_{t|t-1}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}\overline{\boldsymbol{W}^{\boldsymbol{p}}}}(\boldsymbol{\Sigma}_{t|t-1}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}})^{-1}.
\end{aligned}$$

Proof. The proposition 4 is derived using the Lemmas 5, 6, and 7. □

Both steps are employed recursively in order to estimate the elements of the covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{W}}$ and then use this to update our knowledge of the mean vector of $\boldsymbol{W}^{\boldsymbol{p}}$, i.e., $\overline{\boldsymbol{W}^{\boldsymbol{p}}}$. All the steps performed in a particular time step t are summarized in Algorithm 4 as provided in Appendix D.

6.3 Applied Examples

This section presents two case studies illustrating the application of AGVI for multiple time series. The case study 1 presents a simulated multivariate random walk model with a full process error covariance matrix $\mathbf{Q}$ for which the performance of the AGVI method is compared to the adaptive Kalman filter (AKF) methods, namely the *indirect correlation method* (ICM) [48], the *adaptive limited memory filter* (ALMF) [63], and the *sliding window variational adaptive Kalman filter* (SWVAKF) [70]. Each of these AKF methods falls under separate categories as described in Section 2.3.3, where ICM is a correlation method, ALMF is a covariance-matching method (CMM), and the SWVAKF is a Bayesian variational method. The case study 2 shows the application of AGVI on real displacement datasets obtained from a concrete dam in Québec, Canada.

6.3.1 Case Study 1 – Multivariate Random Walk Model

This case study is conducted using five simulated datasets of 1000 time steps with a transition process error having a full covariance matrix $\mathbf{Q}$. The vector of hidden states $\boldsymbol{x}_t$ associated with five time series is given by

$$\boldsymbol{x}_t = [x_t^{\mathtt{LL}_1}\ x_t^{\mathtt{LL}_2}\ x_t^{\mathtt{LL}_3}\ x_t^{\mathtt{LL}_4}\ x_t^{\mathtt{LL}_5}]^{\intercal}.$$

The state transition matrix $\mathbf{A}$ and the observation matrix $\mathbf{C}$ are defined as $\mathbf{A} = \boldsymbol{I}_5$, and $\mathbf{C} = \boldsymbol{I}_5$, The $\mathbf{Q}$ and the $\mathbf{R}$ matrices are defined as

$$\begin{aligned}\mathbf{Q} &= \begin{bmatrix} 1 & -0.3 & -0.2 & -0.1 & 0.25 \\ -0.3 & 3 & 0.35 & 0.4 & 0.45 \\ -0.2 & 0.35 & 4 & 0.5 & 0.55 \\ -0.1 & 0.4 & 0.5 & 0.8 & 0.6 \\ 0.25 & 0.45 & 0.55 & 0.6 & 2 \end{bmatrix}, \\ \mathbf{R} &= 10^{-4} \cdot \boldsymbol{I}_5,\end{aligned}$$

where the off-diagonal covariance terms in the $\mathbf{Q}$ matrix are selected arbitrarily such that it is symmetric and positive-definite, i.e., the eigen values are positive. For AGVI, the prior knowledge for the augmented hidden states $\tilde{\boldsymbol{\mu}}_{0|0} = [\boldsymbol{\mu}_{0|0}; \boldsymbol{\mu}_{0|0}^{\overrightarrow{\boldsymbol{LW}}}]$ and $\tilde{\boldsymbol{\Sigma}}_{0|0} = \text{blkdiag}(\boldsymbol{\Sigma}_{0|0}, \boldsymbol{\Sigma}_{0|0}^{\overrightarrow{\boldsymbol{LW}}})$ are initialized by

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_{0|0} &= [\mathbf{0}_5^\intercal \;\; 1.5 \cdot \mathbf{1}_5^\intercal \;\; 0.1 \cdot \mathbf{1}_{10}^\intercal]^\intercal, \\ \tilde{\boldsymbol{\Sigma}}_{0|0} &= \text{diag}([\mathbf{1}_5^\intercal \;\; 0.1 \cdot \mathbf{1}_5^\intercal \;\; 0.5 \cdot \mathbf{1}_{10}^\intercal]), \end{aligned} \tag{6.12}$$

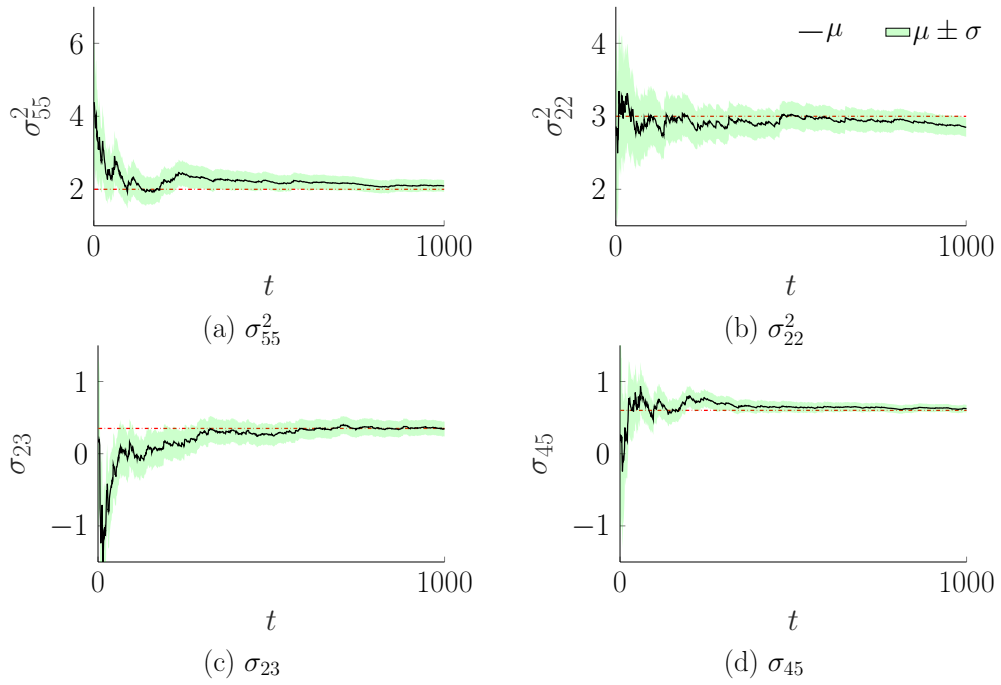
where $\mathbf{0}$ and $\mathbf{1}$ represent vector of zeros and ones, respectively. The mean vector and the covariance matrix for

$$\overrightarrow{\boldsymbol{LW}}_{0|0} = [L_{11} \; L_{22} \; \cdots L_{55} \; L_{12} \; \cdots L_{45}]_{0|0}^\intercal,$$

are given by

$$\begin{aligned}\boldsymbol{\mu}_{0|0}^{\overrightarrow{\boldsymbol{LW}}} &= [1.5 \cdot \mathbf{1}_5^\intercal \;\; 0.1 \cdot \mathbf{1}_{10}^\intercal]^\intercal, \\ \boldsymbol{\Sigma}_{0|0}^{\overrightarrow{\boldsymbol{LW}}} &= \text{diag}([\, 0.1 \cdot \mathbf{1}_5^\intercal \;\; 0.5 \cdot \mathbf{1}_{10}^\intercal]). \end{aligned} \tag{6.13}$$

For the AKF methods, the hidden states are initialized similarly to Equation 6.12 where the mean vector is $\boldsymbol{\mu}_{0|0} = \mathbf{0}_5$ and the covariance matrix is $\boldsymbol{\Sigma}_{0|0} = \boldsymbol{I}_5$. The hyperparameters for ICM include the stable Kalman gain ($\mathbf{K}$) and the auto-covariance lag parameter which are fixed to 0.99 and 1, whereas for ALMF, the initial $\mathbf{Q}$ matrix is chosen as $\boldsymbol{I}_5$. For the SWVAKF, the same parameters are used as provided in the implementation code [70]. Figure 6.1 compares the true values with the online hidden state estimates obtained using AGVI for the four elements of the $\mathbf{Q}$ matrix, namely σ_{55}^2, σ_{22}^2, σ_{23}, and σ_{45}. The true values for each element is shown by the dashed red line and the estimated values are shown by the black solid line and their $\pm 1\sigma$ uncertainty bound is shown using the green shaded region. Table 6.1 shows the average RMSE values over five independent runs for estimating some



(a) σ^2_{55} (b) σ^2_{22} (c) σ_{23} (d) σ_{45}

Figure 6.1 Online estimation of the error variance term (a) σ^2_{55} and (b) σ^2_{22} and the covariance terms (c) σ_{23} and (d) σ_{45} from the full $\mathbf{Q}$ matrix compared to their true values marked by the dashed red line. The estimated values are shown by the black solid line and their $\pm 1\sigma$ uncertainty bound is shown using the green shaded region.

of the elements chosen arbitrarily from the $\mathbf{Q}$ matrix as well as the average computational time for each method. The results show that AGVI outperforms all methods in terms of predictive capacity for most of the variance and covariance terms. In comparison to SWVAKF

Table 6.1 Comparison of the average RMSE values and the computational time (in seconds) for each method. The results are averaged over five independent runs. Each of the methods are picked from different AKF categories where AGVI and SWVAKF are Bayesian methods whereas ALMF is a covariance-matching method (CMM) and ICM is a correlation method. The variance terms and the covariance terms are represented by σ^2_{ii} and σ^2_{ij}, $\forall i, j \in 1, \cdots, \mathtt{D}$.

Type	Category	Methods	RMSE						Time (s)
			σ^2_{22}	σ^2_{33}	σ_{55}	σ_{12}	σ_{24}	σ_{35}	
Online	Bayesian	AGVI	**0.1633**	**0.1437**	0.1567	**0.0624**	**0.0641**	**0.0851**	1.45
Online	Bayesian	SWVAKF	0.4670	0.3133	0.1618	0.0846	0.1177	0.1552	20
Offline	CMM	ALMF	0.1824	0.3295	0.1773	0.0765	0.0662	0.1045	0.12
Offline	Correlation	ICM	0.2421	0.1476	**0.1524**	0.0863	0.0764	0.1280	**0.04**

which is also both a Bayesian and an online estimation method, it is more than an order of magnitude faster. The offline methods, i.e., the ALMF and the ICM, are faster compared to the Bayesian methods but can only provide point estimates which limit their potential in learning sequentially from data as described in Section 5.3.1. The RMSE values and the plots for the remaining variance and covariance terms are provided in Appendix D.

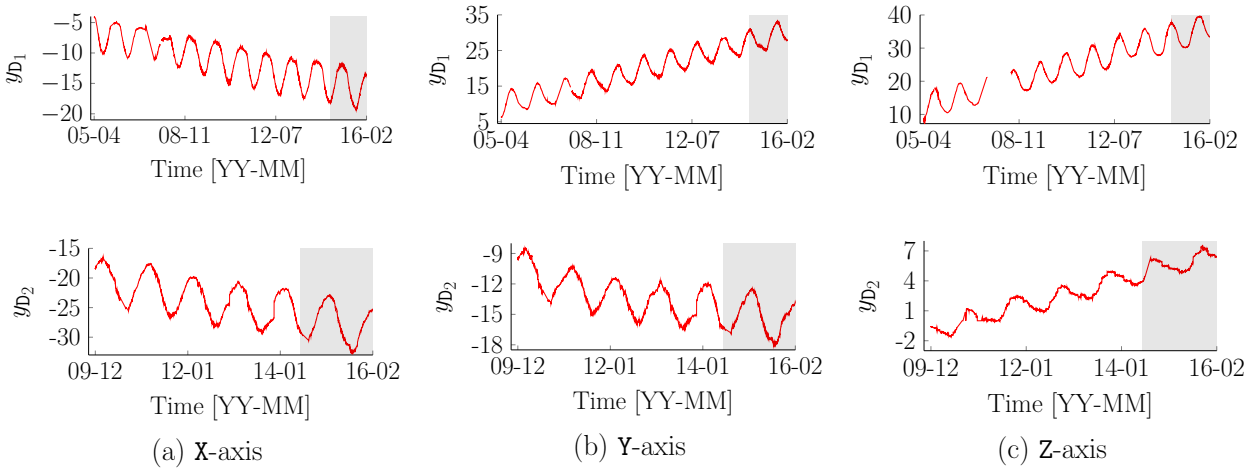
For the consistency test using the normalized innovation square (NIS) values, the two-sided probability region for a 95% C.I. having five degrees of freedom, i.e., the size of the observation vector $\mathtt{Y}$, is [0.831 12.833]. Considering that the total length of the training set is 1000, the theoretical 5% value for the number of acceptable points outside the 95% C.I. is 50. The different prior initialization are chosen such that $\boldsymbol{\mu}_{0|0}^{\overrightarrow{\boldsymbol{LW}}} = [\alpha \cdot \mathbf{1}_5^\intercal \ \beta \cdot \mathbf{1}_{10}^\intercal]^\intercal$, where $\alpha = \{1.5 : 0.1 : 2\}$ and $\beta = \{0.5 : 0.1 : 1\}$ while considering the same covariance matrix as defined in Equation 6.13. Table 6.2 presents the average number of points outside the probability region for the different prior initialization of $\overrightarrow{\boldsymbol{LW}}_{0|0}$ where the average value is computed using five simulated datasets for each combination of $\{\alpha, \beta\}$. The results show that there are on average ≈ 56 points that lie outside the 95% probability region which is comparable to the theoretical value of 50, which verifies that the filter is optimal and provide consistent estimates for the error variance and covariance terms of the full $\mathbf{Q}$ matrix.

Table 6.2 Average number of points outside the 95% probability region for the different prior initialization of $\overrightarrow{\boldsymbol{LW}}_{0|0}$. Each column presents the average value computed using the five simulated datasets for one combination of $\{\alpha, \beta\}$.

	$\{1.5, 0.5\}$	$\{1.6, 0.6\}$	$\{1.7, 0.7\}$	$\{1.8, 0.8\}$	$\{1.9, 0.9\}$	$\{2, 1\}$	Mean
NIS	57.6	57	55.6	54.6	56.6	55.4	56.13

6.3.2 Case Study 2 – Dam Displacement

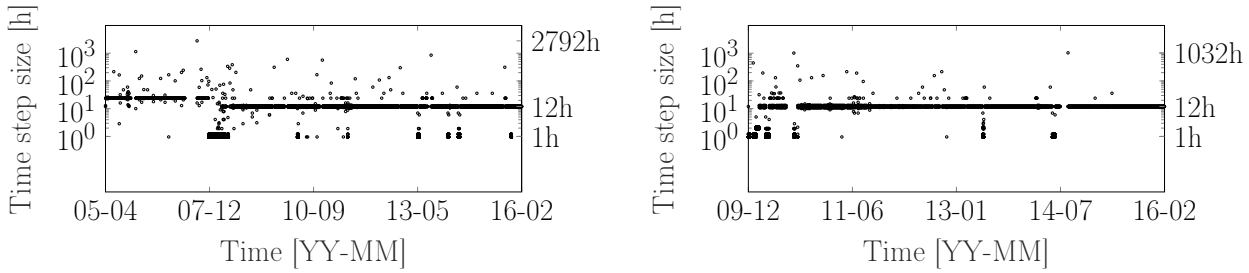
This case study is conducted on data collected by two sensors measuring the displacement from a concrete dam in Québec, Canada. Figure 6.2 shows the displacement datasets along all three orthogonal directions (a) $\mathtt{X}$-axis, (b) $\mathtt{Y}$-axis, and (c) $\mathtt{Z}$-axis; where top plots present the displacement datasets $\boldsymbol{y}_{\mathtt{D}_1}$ that are available from April 2005 to February 2016 with a total of 9995 points and bottom plots present the displacement datasets $\boldsymbol{y}_{\mathtt{D}_2}$ that are available from December 2009 to February 2016 with a total of 5667 points. A test set consisting of 1095 points is considered for all the datasets shown by the gray regions in Figure 6.2. Both



(a) X-axis (b) Y-axis (c) Z-axis

Figure 6.2 Plots showing the displacement datasets in all three directions collected by two sensors from a concrete dam in Canada.

datasets $\boldsymbol{y}_{\mathtt{D}_1}$ and $\boldsymbol{y}_{\mathtt{D}_2}$ are recorded with a non-uniform time-step size Δt as illustrated in Figure 6.3. The vertical axis showing the time-step size is plotted with a log-scale where we see that the most frequent time-step size is 12 hours for both datasets. Figure 6.3(a) shows the time-step size for $\boldsymbol{y}_{\mathtt{D}_1}$ that varies in the range of 1 to 2792 hours and Figure 6.3(b) shows the time-step size for $\boldsymbol{y}_{\mathtt{D}_2}$ that varies in the range of 1 to 1032 hours.



(a) $y_{\mathtt{D}_1}$ (b) $y_{\mathtt{D}_2}$

Figure 6.3 Plots showing the time-step size for the displacement datasets (a) $\boldsymbol{y}_{\mathtt{D}_1}$ and (b) $\boldsymbol{y}_{\mathtt{D}_2}$. The y-axis showing the time-step size is plotted in log-scale.

The Bayesian dynamic linear model (BDLM) components used to model the patterns in the data are the *local trend* to model the baseline and the *kernel regression* component to model the periodic pattern. The process errors are modeled by a zero-mean vector and a full process error covariance matrix $\mathbf{Q}$ to be inferred using the multivariate AGVI method. For handling

non-uniform time-step size [113], a reference time-step $\Delta t^{\texttt{ref}}$ is chosen such that it is the most frequent time-step size in the datasets. The variance and the covariance parameters in the $\mathbf{Q}$ matrix are estimated for the reference time-step $\Delta t^{\texttt{ref}}$, and for any time-step size Δt different than $\Delta t^{\texttt{ref}}$, these parameter values are linearly scaled by the ratio between the current time-step and the reference time-step shown by

$$\mathbf{Q}^{\Delta t} = \mathbf{Q}^{\Delta t^{\texttt{ref}}} \cdot \frac{\Delta t}{\Delta t^{\texttt{ref}}},$$

where $\mathbf{Q}^{\Delta t}$ is the updated process error covariance matrix at the current time-step and $\mathbf{Q}^{\Delta t^{\texttt{ref}}}$ is the covariance matrix for the reference time-step.

The prior knowledge for the hidden states are defined using the default values provided by the OpenBDLM library [113] and the kernel length parameters for the three datasets from each sensor are obtained by offline optimization using the Newton-Raphson method such that $\{\ell_{\mathtt{X}_{\mathtt{D}_1}} = 0.350,\ \ell_{\mathtt{Y}_{\mathtt{D}_1}} = 0.362, \ell_{\mathtt{Z}_{\mathtt{D}_1}} = 0.347\}$ and $\{\ell_{\mathtt{X}_{\mathtt{D}_2}} = 0.288,\ \ell_{\mathtt{Y}_{\mathtt{D}_2}} = 0.289, \ell_{\mathtt{Z}_{\mathtt{D}_2}} = 0.602\}$. The observation error covariance matrix $\mathbf{R}$ is set to $10^{-6} \cdot \boldsymbol{I}_3$ such that the process errors model the residuals considering that the measurements from the sensors are exact. Since there are three time series from each sensor, the $\mathbf{Q}$ matrix has a size of 3×3 with three variance terms and three covariance terms to be evaluated. The prior mean vector and the covariance matrix for $\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}_{0|0}} = [L_{11}\ L_{22}\ L_{33}\ L_{12}\ L_{13}\ L_{23}]^{\intercal}_{0|0}$, are given by

$$\begin{aligned}
\boldsymbol{\mu}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}_{\mathtt{D}_1}}}_{0|0} &= [1 \cdot \mathbf{1}_3\ \ \text{1e-03} \cdot \mathbf{1}_3]^{\intercal}, & \boldsymbol{\mu}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}_{\mathtt{D}_2}}}_{0|0} &= [1\ 1\ 0.5\ \ \text{1e-03} \cdot \mathbf{1}_3]^{\intercal}, \\
\boldsymbol{\Sigma}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}_{\mathtt{D}_1}}}_{0|0} &= \text{diag}([\ \text{1e-04} \cdot \mathbf{1}_3\ \ \text{1e-02} \cdot \mathbf{1}_3]), & \boldsymbol{\Sigma}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}_{\mathtt{D}_2}}}_{0|0} &= \text{diag}([\ \text{1e-04} \cdot \mathbf{1}_3\ \ \text{1e-02} \cdot \mathbf{1}_3]).
\end{aligned}$$

Figure 6.4 shows the online estimation for one variance and two covariance terms in the full $\mathbf{Q}$ matrix for both datasets $\boldsymbol{y}_{\mathtt{D}_1}$ and $\boldsymbol{y}_{\mathtt{D}_2}$. The estimated values are shown by the black solid line and their $\pm 1\sigma$ uncertainty bound is shown using the green shaded region. The online estimation for the remaining terms are provided in Appendix D.

The predictive performance of using AGVI is compared with the one obtained using the Newton-Raphson method where the variance parameters are learned offline through optimization. Table 6.3 shows the test-set root mean square error (RMSE) and log-likelihood values obtained using the AGVI and the Newton-Raphson method for the two displacement datasets along all three axis. Table 6.4 compares the two methods in terms of computational time required, i.e., optimization time and training time expressed in seconds. The results show that AGVI has an accuracy comparable to the Newton-Raphson in terms of RMSE and outperforms it in terms of log-likelihood. Moreover, AGVI is orders of magnitude more

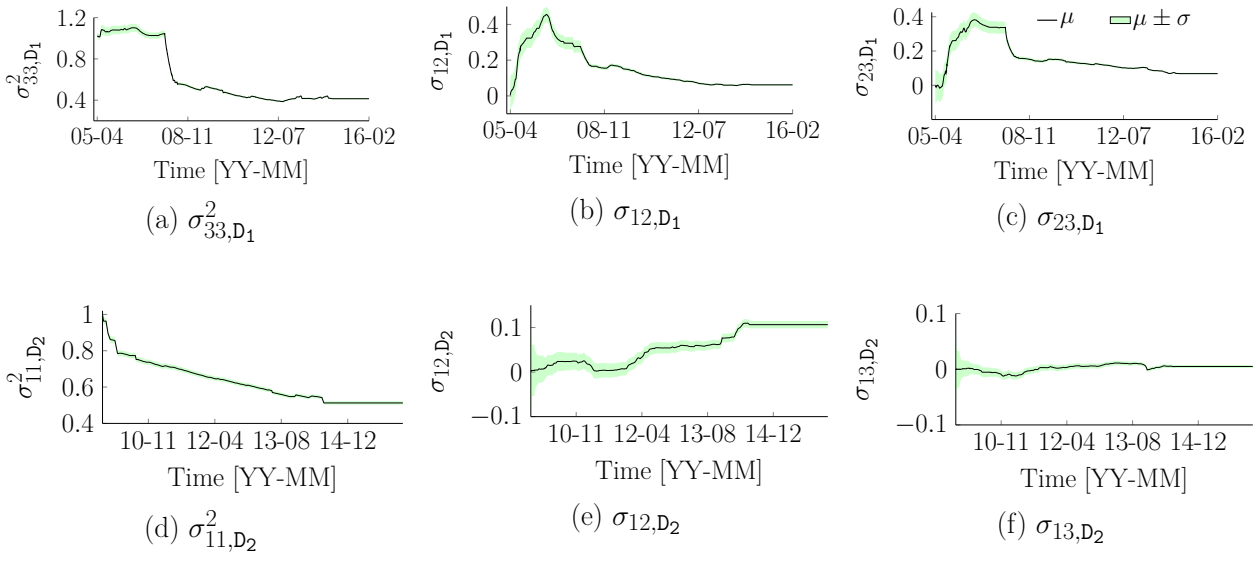


Figure 6.4 Online estimation of the error variance and covariance terms in the full $\mathbf{Q}$ matrix for both datasets $\boldsymbol{y}_{\mathtt{D}_1}$ and $\boldsymbol{y}_{\mathtt{D}_2}$; where (a) $\sigma^2_{33,\mathtt{D}_1}$, (b) $\sigma_{12,\mathtt{D}_1}$, and (c) $\sigma_{23,\mathtt{D}_1}$, (d) $\sigma^2_{11,\mathtt{D}_2}$, (e) $\sigma_{12,\mathtt{D}_2}$, and (f) $\sigma_{13,\mathtt{D}_2}$. The estimated values are shown by the black solid line and their $\pm 1\sigma$ uncertainty bound is shown using the green shaded region.

computationally efficient than Newton-Raphson as it facilitates online learning of the process error variance and covariance terms, thereby avoiding the parameter optimization step. This example shows that AGVI is applicable to real case studies for evaluating the full $\mathbf{Q}$ matrix involving multiple time series.

Table 6.3 Root mean square error (RMSE) and log-likelihood values obtained with the AGVI and the Newton-Raphson methods for the displacements datasets $\boldsymbol{y}_{\mathtt{D}_1}$ and $\boldsymbol{y}_{\mathtt{D}_2}$ along all three axis.

	RMSE		Log-likelihood	
method	AGVI	Newton-Raphson	AGVI	Newton-Raphson
$\mathtt{X}_{\mathtt{D}_1}$	**0.598**	0.622	**−1576.8**	−1585.1
$\mathtt{Y}_{\mathtt{D}_1}$	**0.35**	**0.35**	**−556.76**	−630.38
$\mathtt{Z}_{\mathtt{D}_1}$	**0.64**	**0.64**	**−1031**	−1034.1
$\mathtt{X}_{\mathtt{D}_2}$	**0.57**	**0.57**	**−992.81**	−1069.2
$\mathtt{Y}_{\mathtt{D}_2}$	**0.84**	0.85	**−1414.80**	−2303.9
$\mathtt{Z}_{\mathtt{D}_2}$	**0.17**	**0.17**	**372.37**	361.43

Table 6.4 Comparison of optimization time (in seconds) and training time (in seconds) using the AGVI and the Newton-Raphson method.

method	Optimization Time (s)		Training Time (s)	
	AGVI	Newton-Raphson	AGVI	Newton-Raphson
$\boldsymbol{y}_{\mathtt{D}_1}$	0	2894	61	51
$\boldsymbol{y}_{\mathtt{D}_2}$	0	961	34	26

6.4 Conclusion

In this chapter, the AGVI method is extended to the multivariate case so that the full process error covariance matrix $\mathbf{Q}$ associated with multiple time series can be obtained. The chapter also provided a closed-form square-root filtering technique using the Cholesky decomposition that combines with the AGVI method such that the estimated $\mathbf{Q}$ matrix remain positive semi-definite. The case study 1 shows the application of AGVI for a multivariate random walk model with a full $\mathbf{Q}$ matrix and compares its performance with existing adaptive Kalman filtering (AKF) methods. The results show that AGVI outperforms all methods in terms of predictive capacity for most of the variance and covariance terms, and yields statistically consistent estimates. In comparison to SWVAKF which is both a Bayesian and an online estimation method, it is more than an order of magnitude faster. The offline methods, i.e., the ALMF and the ICM, are faster compared to the Bayesian methods but can only provide point estimates which limits their applicability in decision-making tasks.

The case study 2 shows the application of AGVI on displacement datasets obtained by two sensors along all three directions from a concrete dam in Canada. The predictive performance obtained using AGVI is compared with the one obtained using the Newton-Raphson method where the variance terms are learned offline. The results show that AGVI has a comparable accuracy with Newton-Raphson in terms of RMSE and outperforms it in terms of log-likelihood. Moreover, AGVI is orders of magnitude more computationally efficient than Newton-Raphson as it facilitates online learning of the process error variance and covariance terms, and bypasses the optimization step. Hence, the proposed method is capable of online estimation of aleatory uncertainty in regard to state-space models involving multiple time series as verified with synthetic and validated by real datasets.

CHAPTER 7 Heteroscedastic Aleatory Uncertainty Quantification in Bayesian Neural Network

7.1 Introduction

The *tractable approximate Gaussian inference* (TAGI) method allows for analytical parameter inference in Bayesian neural networks. In its current form (see Section 2.4.2), TAGI can only model homoscedastic aleatory uncertainty that is quantified by a constant error variance across the input covariate-domain. This chapter extends the application of approximate Gaussian variance inference (AGVI) method to model analytically the heteroscedastic aleatory uncertainty for regression tasks with TAGI. This chapter provides the methodology and presents its application on toy problems as well as on benchmark regression datasets where its performance is compared with existing approximate inference methods. The main contributions of this chapter are to

- Provide the methodology for applying the AGVI method to handle heteroscedastic aleatory uncertainty within the TAGI framework.
- Validate the application of AGVI for heteroscedastic regression tasks.
- Provide a comparative analysis for benchmark regression datasets with existing approximate inference methods.

7.2 Methodology

In this section, we apply the AGVI method described in Chapter 5 for analytically inferring the error variance σ_V^2 within the TAGI framework. Similarly to the procedure described in Section 5.2.2 for the process error variance in the context of state space models, the methodology for inferring σ_V^2 can also be summarized in two steps; first, we establish the relationship between the random variables describing the error V, the square of that error V^2, and the expected value of V^2, to obtain the prior knowledge for the error variance σ_V^2; second, we leverage these relationships, and use the posterior PDF of the error V to obtain the posterior knowledge for σ_V^2. Here, we describe AGVI through the univariate case having a single σ_V^2 associated with the observation unit Y whereas without modification, it can be extended for the multivariate case with diagonal covariance matrix $\mathbf{\Sigma_V} = \text{diag}(\boldsymbol{\sigma}_V^2)$. For the first step, the GMA is employed, as presented in Section 3.2 to model V^2 using a Gaussian

random variable such that

$$f(v^2) = \mathcal{N}(v^2; \mu_{V^2}, \sigma^2_{V^2}). \tag{7.1}$$

Using Equation 7.1, and given that V is zero-mean, the variance for V is by definition equal to the expected value for V^2, so that the PDF of V is described by

$$f(v) = \mathcal{N}(v; 0, \mu_{V^2}). \tag{7.2}$$

Following Lemma 1 from Chapter 5, we show that the PDF of V^2 can be described using only the expected value μ_{V^2} following

$$f(v^2|\mu_{V^2}) = \mathcal{N}(v^2; \mu_{V^2}, 2\mu^2_{V^2}), \tag{7.3}$$

where using the GMA, the variance for V^2 is $\sigma^2_{V^2} = 2\mu^2_{V^2}$. In order to maintain the analytical tractability, we assume the hyperparameter μ_{V^2} in Equation 7.3 to be a Gaussian random variable described by $\overline{V^2} \sim \mathcal{N}(\overline{v^2}; \mu_{\overline{V^2}}, \sigma^2_{\overline{V^2}})$, using which, Equation 7.3 can be re-written as

$$f(v^2|\overline{v^2}) \;\; = \;\; \mathcal{N}(v^2; \overline{v^2}, 2(\overline{v^2})^2), \tag{7.4}$$

where the PDF of V^2 is defined using the knowledge of $\overline{v^2}$. Figure 7.1 shows the graphical model representing the relationships between the random variables V, V^2, and $\overline{V^2}$, denoted by the green nodes. The causal relationship between the nodes $\overline{V^2}$ and V^2 is shown by the directed arrow as defined in Equation 7.4. The undirected solid line between the nodes V^2 and V represents the one-to-one relationship between their moments as defined by Equations 7.1 & 7.2. Hence, we obtain the prior predictive PDF of V using the prior predictive PDF of V^2. Using Lemma 2 from Chapter 5, the moments for the prior predictive PDF of V^2 are

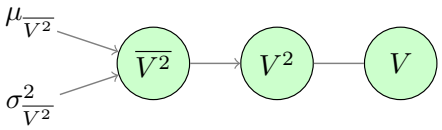


Figure 7.1 Graphical model representing the relationship between the random variables V, V^2, and $\overline{V^2}$, denoted by the green nodes. The causal relationship between the nodes $\overline{V^2}$ and V^2 is shown by the directed arrow as demonstrated by Equation 7.4. The undirected solid line between the nodes V^2 and V represents the one-to-one relationship between their moments as defined by Equations 7.1 & 7.2.

given by

$$\mu_{V^2} = \mu_{\overline{V^2}}, \tag{7.5}$$

$$\sigma^2_{V^2} = 3\sigma^2_{\overline{V^2}} + 2\mu^2_{\overline{V^2}}. \tag{7.6}$$

Using Equations 7.2 & 7.5, the prior predictive PDF of V is described by

$$f(v) = \mathcal{N}(v; 0, \mu_{\overline{V^2}}),$$

where the variance of V is $\sigma^2_V = \mu_{\overline{V^2}}$. Therefore, we obtain the prior knowledge for σ^2_V by using the prior PDF of $\overline{V^2}$ described by its moments $\mu_{\overline{V^2}}$ and $\sigma^2_{\overline{V^2}}$.

In order to obtain the moments for $\overline{V^2}$ as a function of the input covariates $\boldsymbol{x}$, we use a neural network having a two-headed output layer where the first output unit $Z^{(0)}$ models the expected value of the system response and the second output unit is $\overline{V^2}$. This network setup allows handling *heteroscedastic aleatory uncertainty* in regression tasks. Figure 7.2 shows the graphical model for a feedforward network where the two output units are the random variables $Z^{(0)}$ and $\overline{V^2}$. The output unit for $\overline{V^2}$ has its own set of parameters $\boldsymbol{\theta}^{(L)}_{\overline{V^2}}$ connected to the last hidden layer L as shown in red. This graphical model also shows the causal relationship between the random variables Y, $Z^{(0)}$, and V, as per the observation model, along with the graphical model shown in Figure 7.1. This structure presents the flow of information from $\overline{V^2}$ to V, and then to the observation unit Y. Note that in order to restrict the possible values for $\overline{v^2}$ to the positive domain, the original values are transformed using an exponential activation function $\exp(\cdot)$. This lead to a log-normal PDF for which closed-form expressions for the moments are available in [21, §4.2.1]. The moments for the

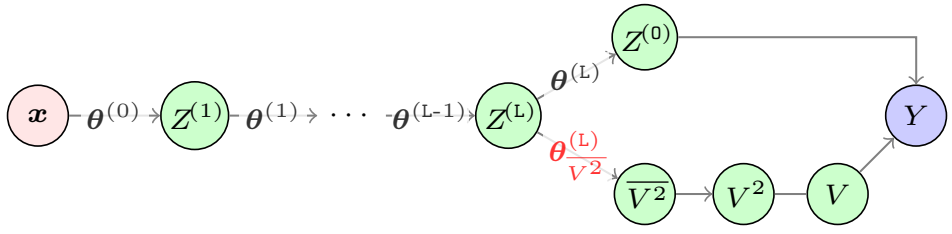


Figure 7.2 Network architecture for TAGI having a two-headed output layer for obtaining the random variables $\boldsymbol{Z}^0$ and $\overline{V^2}$ as a function of the input covariates $\boldsymbol{x}$. The output unit for $\overline{V^2}$ has an additional set of parameters $\boldsymbol{\theta}^{(L)}_{\overline{V^2}}$ connected to the last hidden layer L as shown in red. Also, it shows the extended graphical model representing the causal relationship between the random variables Y, Z^0, and V, as per the observation model, along with the graphical model shown in Figure 7.1.

transformed random variable $\overline{\widetilde{V^2}}$ are shown by

$$\begin{aligned}
\mu_{\overline{\widetilde{V^2}}} &= \exp\Big(\mu_{\overline{V^2}} + 0.5\sigma^2_{\overline{V^2}}\Big), && (7.7)\\
\sigma^2_{\overline{\widetilde{V^2}}} &= \exp\Big(2\mu_{\overline{V^2}} + \sigma^2_{\overline{V^2}}\Big) \cdot (\exp\Big(\sigma^2_{\overline{V^2}}\Big) - 1), && (7.8)\\
\text{cov}(\overline{V^2}, \overline{\widetilde{V^2}}) &= \sigma^2_{\overline{V^2}} \cdot \exp\Big(\mu_{\overline{V^2}} + 0.5\sigma^2_{\overline{V^2}}\Big), && (7.9)
\end{aligned}$$

where $\text{cov}(\overline{V^2}, \overline{\widetilde{V^2}})$ is the covariance between the transformed random variable $\overline{\widetilde{V^2}}$ and the original $\overline{V^2}$.

In order to infer σ_V^2 following the structure provided in Figure 7.2, the process is divided into two steps. Considering that the vector of output units is $\boldsymbol{h} = [z^{(0)}\ v]^\intercal$, the posterior PDF $f(\boldsymbol{h}|y)$ is defined by

$$f(\boldsymbol{h}|y) = \frac{f(\boldsymbol{h}, y)}{f(y)} \approx \mathcal{N}(\boldsymbol{h}; \boldsymbol{\mu}_{\boldsymbol{H}|y}, \boldsymbol{\Sigma}_{\boldsymbol{H}|y}). \tag{7.10}$$

Using the Gaussian conditional equations presented in Section 5.2.2, the posterior mean vector $\boldsymbol{\mu}_{\boldsymbol{H}|y}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{H}|y}$ are obtained following

$$\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{H}|y} &= \boldsymbol{\mu}_{\boldsymbol{H}} + \frac{\boldsymbol{\Sigma}_{\boldsymbol{H}Y}}{\sigma_Y^2}(y - \mu_Y),\\
\boldsymbol{\Sigma}_{\boldsymbol{H}|y} &= \boldsymbol{\Sigma}_{\boldsymbol{H}} - \frac{\boldsymbol{\Sigma}_{\boldsymbol{H}Y} \cdot \boldsymbol{\Sigma}_{\boldsymbol{H}Y}^\intercal}{\sigma_Y^2}. && (7.11)
\end{aligned}$$

Second, the current knowledge of $\overline{V^2}$ is updated using the posterior PDF $f(v|y)$ derived from Equation 7.11. Following Lemma 4 and Proposition 2, the posterior moments for V^2 and $\overline{V^2}$ are given by

$$\begin{aligned}
\mu_{V^2|y} &= \mu^2_{V|y} + \sigma^2_{V|y},\\
\sigma^2_{V^2|y} &= 2(\sigma_{V|y})^4 + 4\sigma^2_{V|y}\mu^2_{V|y},\\
\mu_{\overline{V^2}|y} &= \mu_{\overline{V^2}} + k(\mu_{V^2|y} - \mu_{V^2}),\\
\sigma^2_{\overline{V^2}|y} &= \sigma^2_{\overline{V^2}} + k^2(\sigma^2_{V^2|y} - \sigma^2_{V^2}),\\
k &= \frac{\sigma^2_{\overline{V^2}}}{\sigma^2_{V^2}}.
\end{aligned}$$

The updated knowledge for $Z^{(0)}$ and $\overline{V^2}$ are used to obtain the posterior for the parameters and the hidden units using the layer-wise recursive inference detailed in Section 2.4.2. By

combining the frameworks of TAGI and AGVI, we can perform the analytically tractable inference of the neural network's parameters as well as the error variance, and enable heteroscedastic aleatory uncertainty quantification for regression tasks. The proposed method is referred to as TAGI-V.

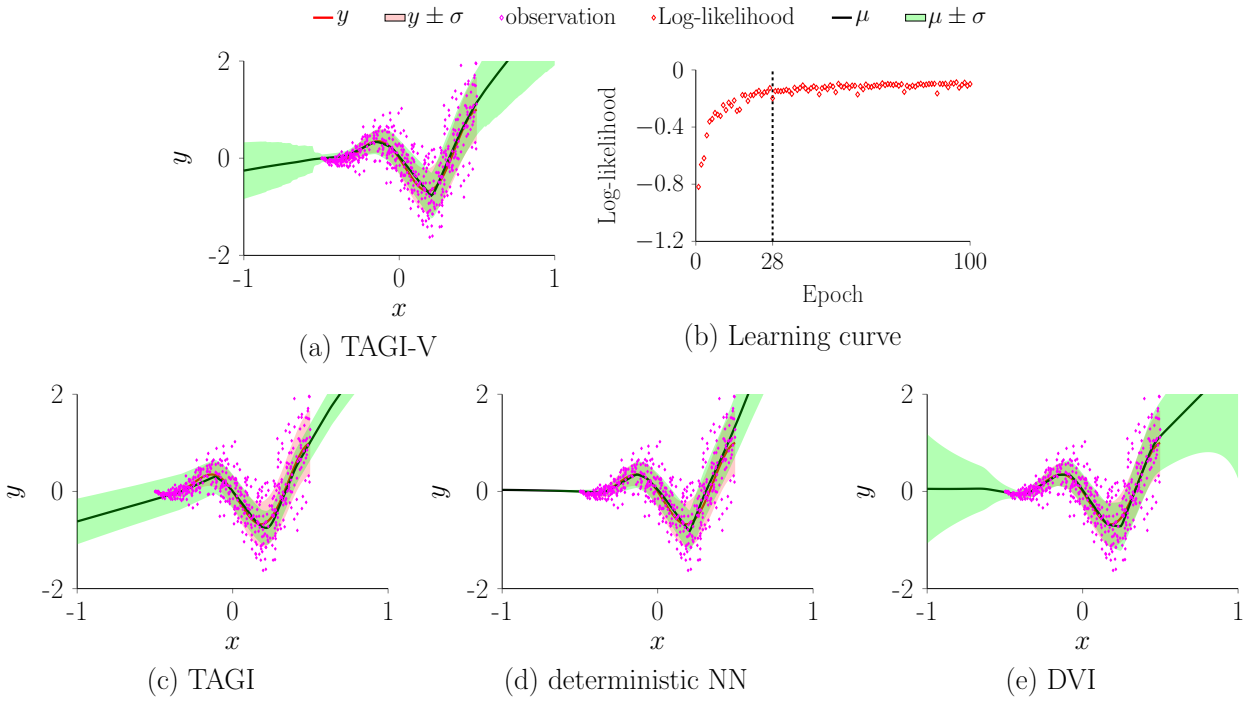
7.3 Applied Examples

In this section, experiments are performed using the TAGI-V method for 1D toy problems and for the UCI regression benchmark datasets [10]. A comparative analysis is provided with other approximate inference methods used for the same regression tasks.

7.3.1 Toy Problem

The TAGI-V method is applied to a 1D heteroscedastic regression problem for $y = -(x+0.5)\cdot \sin(3\pi x) + v$, such that $v \sim \mathcal{N}(0, \sigma_V^2)$, where the heteroscedastic error variance is modeled by $\sigma_V^2 = 0.45\cdot(x+0.5)^2$. A total of 500 observations are generated that are sampled uniformly in the range $[-0.5, 0.5]$ and a two-layer fully-connected network of 128 hidden units is used with ReLU activation function. The prior weights and bias are initialized using He's approach [16] and the inference is carried out using one observation at a time. The TAGI-V is compared with the homoscedastic original version of TAGI [7], a deterministic heteroscedastic neural network trained with backpropagation [15], and the deterministic variational inference (DVI) [13]. Figure 7.3 compares the true function used to generate the data with the predictions described by the expected values and their $\pm 1\sigma$ confidence regions for each of these methods.

The results in Figure 7.3(a) show that TAGI-V is capable of handling the heteroscedastic error variance in the region where training data is available and is able to extrapolate the confidence region outside the training data in order to represent a lack of knowledge. Figure 7.3(b) shows the learning curve representing the evolution of the test log-likelihood as a function of the number of epochs. The optimal epoch for the toy problem is identified to be $\texttt{E} = 28$ using an early-stopping procedure and patience of 5 epochs. In Figure 7.3(c), we can see that the original TAGI method is not capable of handling heteroscedastic error variance and can only model a constant one, both within as well as beyond the training region. Figure 7.3(d) shows the predictions using a deterministic NN trained with backpropagation that can, to a certain extent, model heteroscedastic uncertainty where training data is available but fails to extrapolate the uncertainty beyond the training region. As shown by Figure 7.3(e), DVI is capable of handling heteroscedastic error variance and is better than



(a) TAGI-V (b) Learning curve

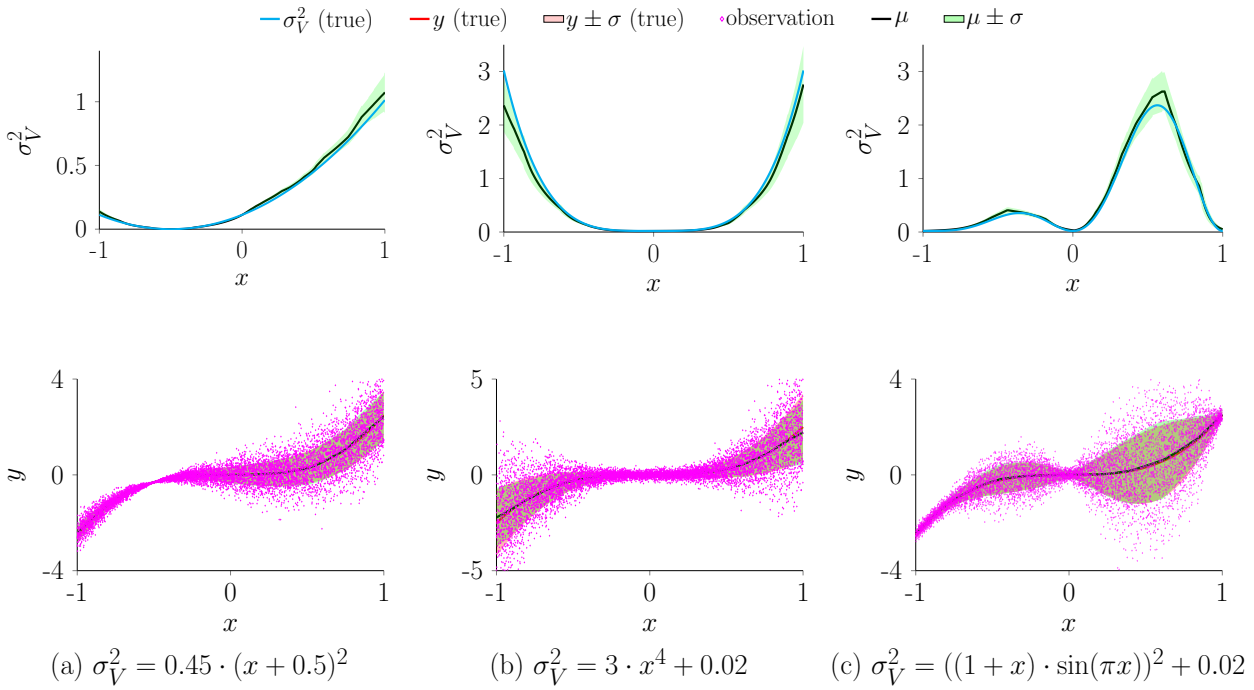
(c) TAGI (d) deterministic NN (e) DVI

Figure 7.3 Application of TAGI-V to a toy problem having a heteroscedastic error variance modeled using $\sigma_V^2 = 0.45 \cdot (x + 0.5)^2$. The training data points are plotted in magenta, the true function $y = -(x + 0.5) \cdot \sin(3\pi x) + v$, and their $\pm 1\sigma$ confidence regions are shown by the red solid line and red shaded region, and the model predictions and their $\pm 1\sigma$ confidence regions are shown by the black solid line and green shaded area. Figure (a) shows the predictions using TAGI-V and (b) shows the learning curve providing the evolution of the test log-likelihood as a function of the number of epochs. Figures (c)-(e) show the predictions using the original version of TAGI [7], a deterministic neural network [15], and DVI [13].

TAGI-V at extrapolating the confidence interval to represent the lack of knowledge outside the training region. However, DVI requires order of magnitudes more epochs (20000) for achieving convergence compared to TAGI-V (28) as shown in Figure 7.3(b).

Figure 7.4 shows the heteroscedastic error variance estimation in three different cases where the true variance in each case is modeled using (a) $\sigma_V^2 = 0.45 \cdot (x+0.5)^2$, (b) $\sigma_V^2 = 3 \cdot x^4 + 0.02$, and (c) $\sigma_V^2 = ((1 + x) \cdot \sin(\pi x))^2 + 0.02$. For each case, the top figure illustrates the true error variance using the cyan solid line, the mean estimate of the variance using the black solid line and their $\pm 1\sigma$ confidence regions in green shaded area. Similarly to Figure 7.3, the bottom figure presents the model predictions and the true observation function $y = 2.5 \cdot x^3 + v$ along with their confidence regions. A total of 10^4 training points are generated in the range $[-1, 1]$ and the same network setup is used as described for the toy problem in Figure 7.3. Early-stopping is used with a patience of 5 to stop the training procedure. The results in Figure

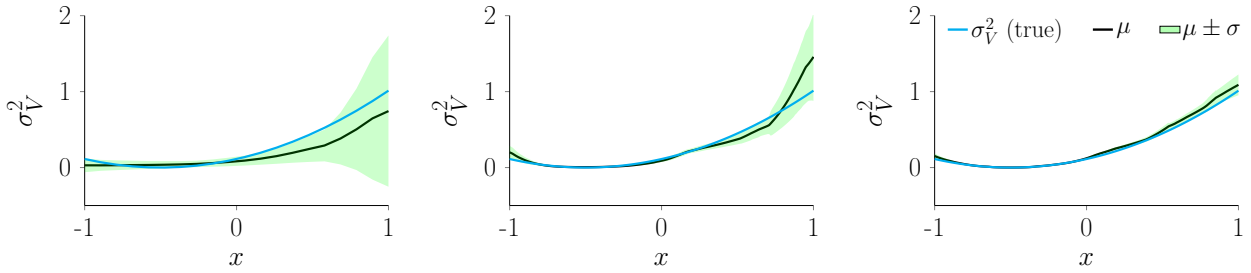
7.3 show that the method is capable of identifying the true error variance as shown for three different functions with respect to the input x.



(a) $\sigma_V^2 = 0.45 \cdot (x + 0.5)^2$ (b) $\sigma_V^2 = 3 \cdot x^4 + 0.02$ (c) $\sigma_V^2 = ((1 + x) \cdot \sin(\pi x))^2 + 0.02$

Figure 7.4 Application of TAGI-V on three toy problems where the true heteroscedastic error variance for each case is modeled using (a) $\sigma_V^2 = 0.45 \cdot (x+0.5)^2$, (b) $\sigma_V^2 = 3 \cdot x^4 + 0.02$, and (c) $\sigma_V^2 = ((1 + x) \cdot \sin(\pi x))^2 + 0.02$. For each case, the top figure illustrates the true error variance using the cyan solid line, whereas the mean estimate of the variance is shown by the black solid line along with their $\pm 1\sigma$ confidence regions in green shaded area. The bottom figures presents the training data points in magenta, the true observation function $y = 2.5 \cdot x^3 + v$ and the $\pm 1\sigma$ confidence regions using the red solid line and red shaded area, and the model predictions and their $\pm 1\sigma$ confidence regions by the black solid line and green shaded area. A total of 10^4 training points are generated in the range $[-1, 1]$ and the same network setup is used as described for the toy problem in Figure 7.3.

Figure 7.5 shows the impact of the amount of available training data on the error variance estimation. When the number of training data points are small, i.e., $\mathtt{D} = 10^2$ as shown by Figure 7.5(a), the mean estimate as shown by the black solid line is inaccurate and there is a large epistemic uncertainty associated with the estimated values. But as the number of data points are increased as shown by Figures 7.5(b) and 7.5(c), not only the epistemic uncertainty shrinks but the mean estimate is also close to the true variance. Hence, the epistemic uncertainty is crucial when few data points are available for learning the heteroscedastic error variance.



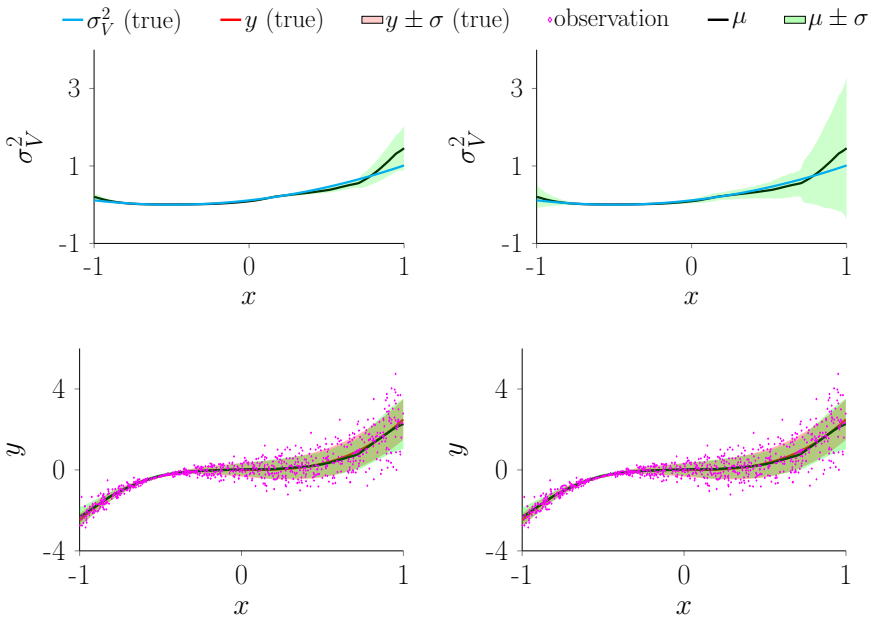
(a) $\mathtt{D} = 10^2$ (b) $\mathtt{D} = 10^3$ (c) $\mathtt{D} = 10^4$

Figure 7.5 Illustration showing the estimated error variance in three different cases where the number of training points are (a) $\mathtt{D} = 10^2$, (b) $\mathtt{D} = 10^3$, and (c) $\mathtt{D} = 10^4$. The true error variance is shown using the cyan solid line, the mean estimate of the variance using the black solid line and their $\pm 1\sigma$ confidence regions in green shaded area.

As mentioned in Section 5.5, Figure 7.6 shows an illustrative example for highlighting the limitation for the current TAGI-V formulation which comes from the mathematical formulation of the AGVI method. It is seen that the epistemic uncertainty of the error variance does not come into play for computing the predictive uncertainty of the model outputs. The top plot in Figure 7.6(a) presents the original estimations of the error variance, while the bottom plot shows the model predictions, whereas the top plot in Figure 7.6(b) displays the same mean estimate for the error variance but with artificially increased epistemic uncertainty. The predictive uncertainty shown in the bottom plot of Figure 7.6(b) do not differ from that of (a) as the mean estimates are the same even when the epistemic uncertainties are different.

7.3.2 Regression Benchmarks

In this section, TAGI-V is compared with *probabilistic backpropagation* (PBP) [10], MC-dropout [8,114], a deterministic neural network [15], ensemble of neural networks [9], DVI [13], *probabilistic backpropagation with the matrix-variate Gaussian* (MVG) distribution (PBP-MV) [11], *Variational matrix Gaussian* (VMG) [12], and the original version of TAGI [7] for the small UCI regression datasets using the experimental setup provided by Hernández-Lobato and Adams [10]. This setup has been extensively used in the literature to evaluate the predictive capacity of the approximate inference methods which are reviewed in Section 2.4.1. The implementation details for each method is provided in Appendix E.1.



(a) original epistemic uncertainty (b) modified epistemic uncertainty

Figure 7.6 Illustration showing the limitation of TAGI-V where it does not account for the epistemic uncertainty of the error variance while computing the predictive uncertainty of the model output. The top plot in (a) presents the original estimations of the error variance and the bottom plot shows the model predictions. The top plot in (b) shows the same mean estimate for the error variance while its epistemic uncertainty is artificially increased. As in (a), the bottom plot in (b) shows the predictive uncertainty associated with the modified epistemic uncertainty which remains unchanged.

Small UCI Datasets

Each dataset is randomly split into a training and test set having 90% and 10% of the data, and the same indices are maintained in both sets for each method. A total of 20 data splits are considered to compute the average test performance. For comparative purposes, a network having a single hidden layer of 50 units is considered for each dataset except for Protein which has 100 units. For TAGI-V, the data is normalized, a ReLU activation function is used, and the batch size considered is $\mathtt{B} = 32$. The prior covariances for weights and bias are initialized using He's approach [16]. Note that the scaling factor associated with the prior variance of the weights for the mean as well as the error variance are tuned for each dataset for proper initialization. The details regarding the grid-search procedure are provided in Appendix E.2. Moreover, an early-stopping procedure is used to identify the optimal number of epochs for each dataset by dividing the training set into an $80-20\%$ train-validation set, see details in Appendix E.3.

Two sets of comparison are provided. First, the *epoch setting* is presented, where each method is trained for 100 epochs and the learning curves are plotted showing the average test log-likelihood and test RMSE during training. Second, the *time setting* is considered, where the learning curves are reported as a function of the average training time per epoch for each method. The epoch setting allows assessing the performance for each method solely on the basis of predictive accuracy. On the other hand, the time setting provides a better assessment of the methods both in terms of accuracy and computational time.

For the epoch setting, the performance of each method is reviewed for all the datasets based on the test log-likelihood and test RMSE over 100 epochs. Figure 7.7 shows the learning curves for the datasets Boston, Kin8nm and Power under the epoch setting while the results for the other datasets are presented in Appendix E.4. In general, there is not one method that outperforms the others in all the datasets. While evaluating the absolute performance in terms of the test log-likelihood, TAGI-V provides the best results in 3 out of 9 datasets,

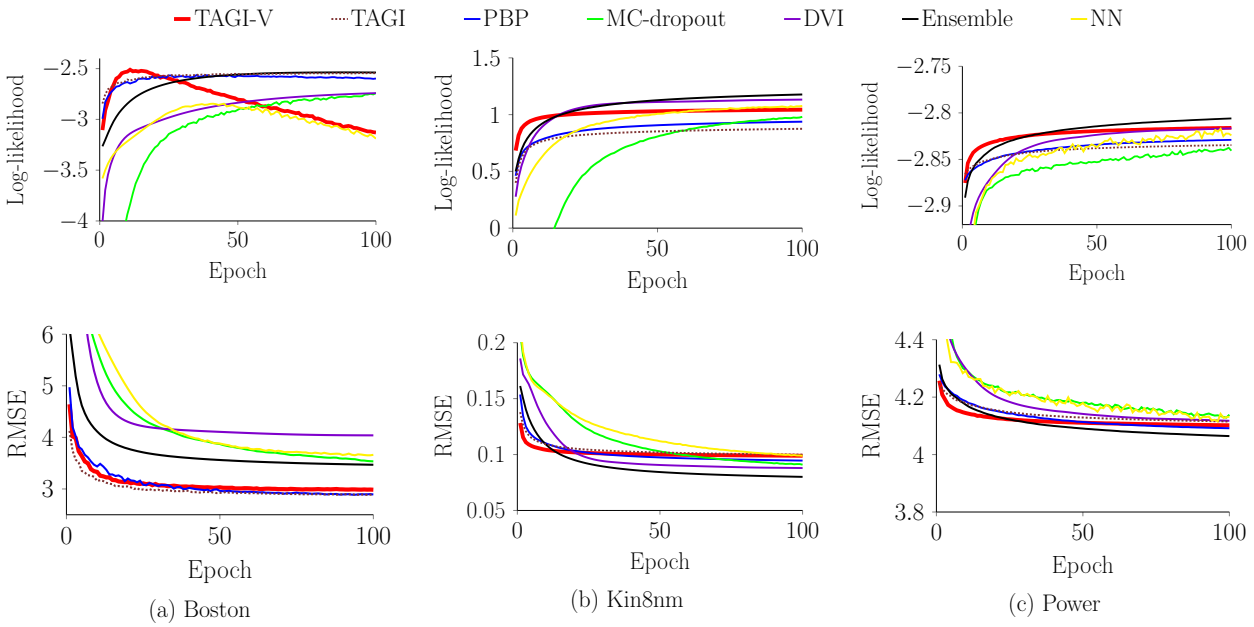


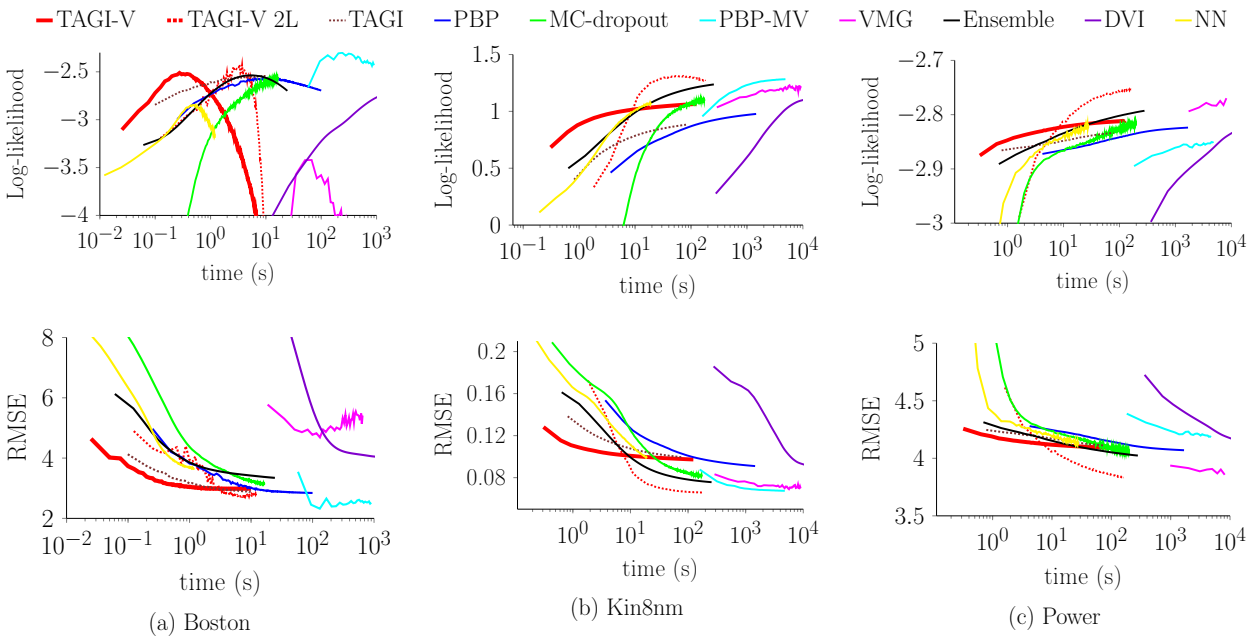
Figure 7.7 Comparison for the test log-likelihood and test RMSE for the datasets a) Boston, b) Kin8nm, and c) Power under the epoch setting. For each subset of figures, the top and bottom graphs shows the learning curves for test log-likelihood and test RMSE respectively for a total of 100 epochs. The horizontal axis shows the number of epochs and the vertical axis shows the test log-likelihood (top figure) or the test RMSE (bottom figure). The colored line plots are: TAGI-V (red solid line), PBP (blue solid line) [10], MC-dropout (green solid line) [8], DVI (purple solid line) [13], deterministic NN (yellow solid line) [15], Ensemble (black solid line) [9], and TAGI (brown dotted line) [7].

namely Boston, Energy and Yacht, and second only to DVI in Concrete, and Naval. It is seen that using an ensemble of neural networks outperforms a single neural network in all datasets except in Naval. Also, Ensemble outperforms the other methods in 4 out of 9 datasets (Protein, Wine, Power, and Kin8nm). In terms of test RMSE, the PBP and Ensemble provide the best results in 3 out of the 9 datasets. PBP provides the best test RMSE in Yacht, Boston and Concrete, and Ensemble in Power, Kin8nm, and Wine. In comparison with the original version of TAGI, TAGI-V provides similar test RMSE while outperforming it in terms of test log-likelihood. It is to be noted that MC-dropout can achieve a higher predictive performance than what is reported here for 100 epochs, if trained until convergence ($\approx$ 4000 epochs) [114]. Although the epoch setting presents the current standard used by other authors for comparing predictive performances, it only shows the predictive accuracy achieved over a fixed number of epochs, and not the computational time required to achieve those results. To better assess the trade-off between predictive accuracy and computational time, the methods need to be evaluated under the time setting.

Figure 7.8 shows the learning curves for the datasets Boston, Kin8nm and Power under the time setting where the horizontal axis represents the training time (s) for each method. The horizontal axis is presented in log-scale (base 10) for accommodating the large disparities with respect to training time between the methods. The figures comparing the results for other datasets are presented in Appendix E.4. The learning curves for PBP-MV and VMG in Figures 7.8, E.6 and E.7 are provided by Sun et al. [11].

While comparing the average training time per epoch between the methods, TAGI-V is found to be $\approx$ 100 times faster than PBP-MV and VMG, $\approx$ 10 times faster than PBP, and $\approx$ 3 times faster than Ensemble. MC-dropout has an average training time per epoch equivalent to TAGI-V but requires hyperparameter tuning (dropout rate and τ parameter) and orders of magnitude more epochs ($\approx$ 4000) to achieve the stated predictive performance [114]. The deterministic NN is the fastest method by a factor of $\approx$ 1.5 compared with TAGI-V, but it provides a poor predictive accuracy as shown in Figure 7.8. The average training time per epoch for all the methods is provided in the Table E.2 from Appendix E.5. In terms of absolute predictive performance, PBP-MV reports the best test log-likelihood and test RMSE among all methods in 8 out of the 9 datasets, while VMG outperforms PBP-MV in Power. Even though PBP-MV and VMG produce state-of-the-art results in terms of predictive accuracy, these methods take orders of magnitudes more computational time than all other methods except DVI which has a similar computational demand. The details regarding the predictive performance for each of the method is provided in Appendix E.6.

In order to further demonstrate the superiority of TAGI-V in comparison with approaches



(a) Boston (b) Kin8nm (c) Power

Figure 7.8 Comparison for the test log-likelihood and test RMSE for the datasets a) Boston, b) Kin8nm, and c) Power under the time setting. The horizontal axis represents training time (in sec) in log scale (base 10) and the vertical axis represents the test log-likelihood (top figure) or the test RMSE (bottom figure) in linear scale. The colored line plots are: TAGI-V (red solid line), PBP (blue solid line) [10], MC-dropout (green solid line) [8], DVI (purple solid line) [13], deterministic NN (yellow solid line) [15], Ensembles (black solid line) [9], TAGI (brown dotted line) [7], TAGI-V 2L (red dotted line) that represents a TAGI-V network of two layers and 100 hidden nodes, PBP-MV (cyan solid line) [11], and VMG (magenta solid line) [12]. The learning curves for PBP-MV and VMG are reproduced directly from the original article [11].

such as PBP-MV that can reach a high accuracy at the expense of computational efficiency, the performance of TAGI-V is tested by using 2 layers and 100 hidden nodes (TAGI-V 2L) as represented by the red dotted lines in Figures 7.8, E.6 and E.7. The two-layer network not only outperform PBP-MV and VMG for test log-likelihood in all datasets except in Concrete and Wine, but while doing so remains two orders of magnitude faster than these methods. For the test RMSE, the two-layer network exceeds the performance of PBP-MV and VMG in 5 out of 9 datasets (Concrete, Energy, Kin8nm, Yacht and Power). These results demonstrate that TAGI-V can achieve the state-of-the-art predictive accuracy by using a larger neural network architecture while still being computationally faster than any of the other approaches.

Large UCI Datasets

The TAGI-V is tested for the large UCI datasets: *elevators*, *keggdirected*, *keggundirected*, *pol*, and *skillcraft*. The experimental framework provided by Wilson et al. [4] is used. Each dataset is randomly split into a training and test set having 90% and 10% of the data. The experiment is carried out for 10 random splits for computing the average test RMSE and normalized test log-likelihood as per the original setup by Wilson et al. [1, 4].

On all datasets, except skillcraft, a network of five hidden layers is used where the number of hidden units in each layer are: $[1000, 1000, 500, 50, 2]$. For skillcraft, a smaller network is used such that the structure is: $[1000, 500, 50, 2]$. A ReLU activation unit is used, the batch size considered is $\mathtt{B} = 10$, and an exponential function is used for the error variance output. The prior variances for weights and bias are initialized using He's approach [16]. Similarly to the procedure for the small UCI datasets, the gain parameters associated with the variances for all the hidden layers and the output layer connected to the mean and error variance are tuned using a grid-search procedure. Also, an early-stopping procedure is used to stop the training process with a fixed patience of 3 epochs. The hyperparameters used for each dataset are provided in Table E.5 from Appendix E.7.

Tables 7.1 & 7.2 provides the test RMSE and normalized test log-likelihood for the large UCI datasets. The direct comparison is made with the best performing *sub-space inference method* [1] i.e., principal component analysis combined with variational inference (PCA+VI), along with the *stochastic weight averaging-Gaussian* (SWAG) [2], the *orthogonally decoupled variational Gaussian Processes* (Orth VGP) [3], the *deep kernel learning with a spectral mixture kernel* (DKL) [4], the *Bayesian final layers* (NL) [5], the *stochastic gradient descent* (SGD) obtained from Izmailov et al. (2020) [1], and the *fastfood kernel Gaussian process* (FF) [6]. The test log-likelihood values shows that TAGI-V performs better than all the methods in four of the five datasets except in Pol for which it is second best after SI. The method is also competitive for RMSE values for which DKL is the best performing method.

Table 7.1 RMSE comparison between the inference methods on large UCI regression datasets. The direct comparison is made with the best performing *sub-space inference method* [1] i.e., principal component analysis combined with variational inference (PCA+VI), along with the *stochastic weight averaging-Gaussian* (SWAG) [2], the *orthogonally decoupled variational Gaussian Processes* (Orth VGP) [3], the *deep kernel learning with a spectral mixture kernel* (DKL) [4], the *Bayesian final layers* (NL) [5], the *stochastic gradient descent* (SGD) obtained from Izmailov et al. (2020) [1], and the *fastfood kernel Gaussian process* (FF) [6] (Rank legend: **first**). The $\pm\sigma$ represents one standard deviation computed over 10 splits. The results for TAGI-V are averaged over 3 random seeds.

Datasets	TAGI-V	PCA + VI (SI)	SWAG	DKL	Orth VGP	NL	SGD	FF
Elevators	0.085 ± 0.002	0.088 ± 0.001	0.088 ± 0.001	**0.084 ± 0.02**	0.0952	0.101 ± 0.002	0.103 ± 0.035	0.089 ± 0.002
KeggD	0.129 ± 0.005	0.128 ± 0.029	0.129 ± 0.029	**0.10 ± 0.01**	0.119	0.134 ± 0.036	0.132 ± 0.017	0.12 ± 0.00
KeggU	0.122 ± 0.002	0.160 ± 0.043	0.160 ± 0.043	**0.11 ± 0.00**	0.117	0.120 ± 0.003	0.186 ± 0.034	0.12 ± 0.00
Pol	2.737 ± 0.135	**2.50 ± 0.068**	3.11 ± 0.070	6.617 ± 0.00	4.30 ± 0.20	4.380 ± 0.853	3.900 ± 6.003	–
Skillcraft	0.45 ± 0.135	0.293 ± 0.015	0.293 ± 0.015	**0.25 ± 0.00**	–	0.253 ± 0.011	0.288 ± 0.014	**0.25 ± 0.02**

Table 7.2 Normalized log-likelihood comparison between the inference methods on large UCI regression datasets. The direct comparison is made with the best performing *sub-space inference method* [1] i.e., principal component analysis combined with variational inference (PCA+VI), along with the *stochastic weight averaging-Gaussian* (SWAG) [2], the *orthogonally decoupled variational Gaussian Processes* (Orth VGP) [3], the *deep kernel learning with a spectral mixture kernel* (DKL) [4], the *Bayesian final layers* (NL) [5], the *stochastic gradient descent* (SGD) obtained from Izmailov et al. (2020) [1], and the *fastfood kernel Gaussian process* (FF) [6] (Rank legend: **first**, **second**). The $\pm\sigma$ represents one standard deviation computed over 10 splits. The results for TAGI-V are averaged over 3 random seeds.

Datasets	TAGI-V	PCA + VI (SI)	SWAG	DKL	Orth VGP	NL	SGD	FF
Elevators	**-0.298 ± 0.027**	-0.325 ± 0.019	-0.374 ± 0.021	–	-0.4479	-0.698 ± 0.039	-0.538 ± 0.108	–
KeggD	**1.274 ± 0.122**	1.085 ± 0.031	1.080 ± 0.035	–	1.0224	0.935 ± 0.265	1.012 ± 0.154	–
KeggU	**0.793 ± 1.034**	0.757 ± 0.028	0.749 ± 0.029	–	0.7007	0.670 ± 0.038	0.602 ± 0.224	–
Pol	0.718 ± 0.108	**1.764 ± 0.271**	1.533 ± 1.084	–	0.1586	-2.84 ± 0.226	1.073 ± 0.858	–
Skillcraft	**-0.981 ± 0.031**	-1.179 ± 0.033	-1.180 ± 0.033	–	–	-1.002 ± 0.050	-1.162 ± 0.032	–

7.4 Conclusion

The TAGI-V method proposed in this chapter provides an analytical method for handling heteroscedastic aleatory uncertainty and overcomes the limitation of the original version of TAGI which can only handle homoscedastic error variance. The method proposed combines the TAGI framework that allows for the analytical inference of the parameters' posterior PDF in Bayesian neural networks, and AGVI that enables analytical inference of the error variance. TAGI-V outperforms the original version of TAGI in terms of test log-likelihood while providing similar test RMSE for all small UCI datasets. In comparison with other approximate inference methods, TAGI-V is an order of magnitude faster and exhibits superior predictive performance. The TAGI-V framework was also tested for large UCI datasets for which it provided better log-likelihood for four out of five datasets compared to the benchmark methods while providing competitive performance in terms of RMSE. However, the existing framework cannot yet quantify aleatory uncertainties in a joint prediction model for multi-output regression tasks. Moreover, the framework requires identifying the hyperparameters associated with the initial variances for the weights and biases, i.e., the gain parameters α and β using a grid-search procedure. A future work to further improve the framework would be infer these gain parameters analytically.

CHAPTER 8 Conclusion

8.1 Thesis Conclusion

This thesis proposes new analytical Bayesian methods for parameter inference in probabilistic models. The methods developed improve the models' predictive performance and scalability in regard to practical engineering applications. The following section present the conclusions derived from this thesis.

State-space models and Bayesian neural networks involve unknown parameters for not only modeling a physical phenomenon, but also for quantifying the model's epistemic and aleatory uncertainties. In the context of state-space models, parameters that exist in the transition and observation equations can be inferred as hidden states using a multiplicative structure. However, a key limitation in the existing framework is that an analytically tractable formulation does not exist for multiplicative state-space models that would allow closed-form inference of the model parameters as hidden states. To overcome this limitation, the thesis proposed an analytical method to handle multiplicative state-space models by leveraging the Gaussian multiplicative approximation (GMA). The GMA provides closed-form moment equations for the product of two Gaussian hidden states. The framework enables the analytical Bayesian inference for the hidden state vector in a multiplicative structure involving the product of two hidden states in the transition and/or observation models. The framework ensures that the Kalman filter is still applicable for performing closed-form posterior inference for a multiplicative structure which can now be represented by a Bayesian dynamic linear model (BDLM). The proposed method is validated with synthetic as well as SHM-based real datasets and have shown to exceed the performance of the cubature Kalman filter both in terms of predictive capacity and computational complexity.

Additionally, the current BDLM framework is limited to modeling linear relationships between the independent and the interdependent time series described by a constant regression coefficient. For modeling a nonlinear dependency between two time series, the thesis proposes the state-based regression (`SR`) method that allows closed-form inference of the state-dependent regression coefficient as well as the interdependent state variable. The `SR` method provides an interpretable representation of how each nonlinear dependency explains specific patterns in the interdependent time series. The two case studies involving a dam's displacement datasets show that the predictive performance for the `SR` method is superior both in terms of root mean square error (RMSE) and log-likelihood compared to the linear dependency model in the existing BDLM framework.

Another key feature of state-space models is that it is computationally cheap to estimate the expected values and the covariance matrix that are quantifying the mean and the epistemic uncertainties for the hidden state variables because we can rely on an analytical formulation for performing Bayesian inference. In contrast, obtaining optimal estimates for the variance parameters in the process ($\mathbf{Q}$) and observation ($\mathbf{R}$) error covariance matrices that quantifies the model's aleatory uncertainties is typically the most computationally demanding task in the state estimation procedure. Even though in many situations the matrix $\mathbf{R}$ can be considered to be known from the measuring instrument specifications, it remains a challenge to develop a computationally efficient method which is able to perform closed-form online estimation of the matrix $\mathbf{Q}$. To overcome this limitation, this thesis proposes the approximate Gaussian variance inference (AGVI) method that provides closed-form analytical inference for the univariate as well as the multivariate process error's variance and covariance terms. The AGVI method is verified and validated using both synthetic and real datasets and have shown to provide accurate as well as statistically consistent estimates for the mean and variance of the process error's variance and covariance terms in the $\mathbf{Q}$ matrix at each time step. In comparison to the offline gradient-based optimization approaches and the existing adaptive Kalman filtering (AKF) methods, the AGVI method has a better performance in terms of its predictive capacity as well as a higher computational speed.

Analytical parameter estimation is feasible in Bayesian neural networks using the tractable approximate Gaussian inference (TAGI) method. However, a key limitation of the TAGI framework is that it is restricted to modeling homoscedastic aleatory uncertainty. The thesis provides the TAGI-V framework to model the heteroscedastic aleatory uncertainty in Bayesian neural networks by combining the AGVI method with the TAGI framework. In comparison to existing approximate inference methods applied to the small UCI regression benchmark, the framework is an order of magnitude faster and exhibits superior predictive performance. The TAGI-V framework was also tested for large UCI datasets for which it provided a better log-likelihood for four out of five datasets compared to the benchmark methods while providing competitive performance in terms of RMSE.

It is important to highlight that the GMA formulation is fundamental to all the analytical methods developed in this thesis. Without the closed-form moment equations for the product of two Gaussian hidden states, it would not have been possible to perform the parameter inference step analytically. The simple algebraic expressions for the expected values, the variance, and the covariance terms have allowed us to obtain the moments associated with the multiplicative terms. For instance, we could model parameters such as the autoregressive parameter in the first-order autoregressive process as a hidden state, generate new components in the BDLM framework such as the trend multiplicative and the double kernel regression

that models the product of a local trend and a periodic component and the product of two periodic components. The state-based regression method relies on GMA to analytically infer the product of the regression coefficient and the independent hidden state. The AGVI method uses the GMA equations for formulating the relationship between the process errors, the square of the process errors, and their expected values that allow closed-form posterior inference of the full $\mathbf{Q}$ matrix in the context of state-space models. Moreover, the AGVI method found its application in the context of Bayesian neural networks for analytically quantifying the heteroscedastic aleatory uncertainty necessary for practical regression tasks. Overall, the GMA is the thread line connecting each chapter of this thesis that has enabled developing analytical inference methods for estimating parameters in probabilistic models.

In conclusion, the methods proposed in this thesis have addressed key limitations in estimating parameters for state-space models and Bayesian neural networks. The case studies presented in the context of state-space models focuses on structural health monitoring (SHM) applications. However, the methods themselves are not restricted to SHM and are applicable in general for time series forecasting and regression tasks in many other engineering fields including navigation, aerospace, telecommunications, etc., where parameter estimation is a necessary step. Finally, this thesis lay the groundwork for analytical tractability in parameter estimation which is at the core of advancing the existing methods for large-scale implementation.

8.2 Limitations

This section examines the limitations that exist in the methods proposed in this thesis. Resolving these limitations, can further improve the applicability of these methods for a wide array of engineering applications.

8.2.1 The Gaussian Multiplicative Approximation

The Gaussian multiplicative approximation provides closed-form moments for the product of two hidden states that could be leveraged to analytically model parameters as hidden states in the context of state-space models. However, the closed-form moments are only available for the product of two Gaussian random variables. The current formulation does not provide closed-form moments for higher-order monomials or polynomials. This could allow handling dynamic systems having general polynomial functions and not be restricted to the product of two random variables.

8.2.2 State-Based Regression

The state-based regression method is capable of identifying short-term and long-term stationary patterns in time series. However, as seen in the case studies, the method cannot handle non-stationary patterns by itself and would require combining the `SR` method with a regime-switching approach. This would enable identifying whether the model is in a normal regime or an abnormal regime allowing the predictive model to identify anomalies in real-time. Moreover, it might be worth investigating additional explanatory variables that can identify specific dam behavior such as creep or creep-relief effects. Furthermore, a key limitation in the state-based regression method is the need for feature engineering to preselect the explanatory variables such as the average long-term trend, mean-centered water level, and the moving averages of the air temperature's residuals as shown in the case studies that require domain specific knowledge. This aspect is currently the factor limiting the scalability of the approach for analyzing large SHM databases.

8.2.3 Approximate Gaussian Variance Inference

The AGVI method has shown promising results and can provide an efficient way of reducing the computational time for estimating the parameters quantifying aleatory uncertainties in probabilistic models. However, one key limitation of AGVI is that the uncertainty associated with the process error's variance is not considered as σ_W^2 is shown to be equal to $\mu^{\overline{W^2}}$, i.e., the expected value of $\overline{W^2}$. This limitation is attributed to the Gaussian assumption for the variance term, for which inverse-gamma PDF is the theoretical distribution. Moreover, in complex practical applications, the $\mathbf{Q}$ matrix might be time-varying, i.e., the true values for the variance and covariance terms in the $\mathbf{Q}$ matrix might vary as a function of time. The existing framework cannot handle time-varying $\mathbf{Q}$ matrix and requires it to be stationary over time.

8.2.4 TAGI-V

TAGI-V was successfully applied to quantify heteroscedastic aleatory uncertainty for single-output regression tasks. However, the existing framework cannot yet quantify aleatory uncertainties in a joint prediction model for multi-output regression tasks. Moreover, the framework requires identifying the hyperparameters associated with the initial variances for the weights and biases, i.e., the gain parameters α and β using a grid-search procedure. A future work to further improve the framework would be infer these gain parameters analytically.

8.3 Future Research

This section provides two future research directions that builds upon the mathematical formulations developed in this thesis.

8.3.1 Analytically Tractable Skewness Inference

One future case study is to analytically infer the *skewness* parameter along with the expected value and the variance parameter for any skewed probability distribution. This would allow describing *generalized extreme value distributions* (GEV) used to model the maxima or minima of a sequence of random variables [115]. The GEV distribution is widely used to model financial risks, as well as extreme events such as maximum precipitation, temperature, fire hazards, etc.

Generalized Extreme Value Distribution

Given that $Y \sim \text{GEV}(\alpha, \beta, \zeta)$, where $\alpha \in \mathbb{R}$ is the location parameter, $\beta \in \mathbb{R}^+$ is the scale parameter, $\zeta \in \mathbb{R}$ is the shape parameter. The moments for Y are shown by

$$\begin{aligned} \mathbb{E}[Y] &= \alpha + (g_1 - 1)\frac{\beta}{\zeta}, \text{ for } \zeta < 1, \\ \text{var}(Y) &= (g_2 - g_1^2)\frac{\beta^2}{\zeta^2}, \\ \text{skewness}(Y) &= \frac{g_3 - 3g_2g_1 + 2g_1^3}{(g_2 - g_1^2)^{3/2}}, \text{ for } \zeta > 0, \end{aligned}$$

where the mean, the variance, and the skewness are functions of α, β, and ζ. The term $g_k = \Gamma(1 - k\zeta),\ k \in \{1, 2, 3\}$ is a gamma function. With the knowledge of these three moments, we can identify the three parameters using which the GEV distribution can be constructed. The mean and the variance parameters can be analytically inferred using the AGVI method. But we do not have the mathematical formulation to infer the skewness parameter.

Skewness

The skewness of a random variable Y is the 3^{rd} standardized moment represented by $\tilde{\mu}_3$ and defined as

$$\begin{aligned}\tilde{\mu}_3 &= \mathbb{E}\left[\left(\frac{Y-\mu_Y}{\sigma_Y}\right)^3\right], \\ \tilde{\mu}_3 &= \frac{\mathbb{E}\left[Y^3 - 3\mu_Y\sigma_Y^2 - \mu_Y^3\right]}{\sigma_Y^3}. \end{aligned} \tag{8.1}$$

Using the observation model $y = z^{(0)} + v,\ v : V \sim \mathcal{N}(0, \sigma_V^2)$, we can simplify the numerator term in Equation 8.1 shown by

$$\begin{aligned}\tilde{\mu}_3 &= \frac{\mathbb{E}\left[(Z^{(0)}+V)^3\right] - 3\mathbb{E}[Z^{(0)}]\text{var}(Z^{(0)}+V) - \mathbb{E}[Z^{(0)}]^3}{\sigma_Y^3}, \\ &= \frac{\mathbb{E}\left[(Z^{(0)}+V)^3\right] - 3\mathbb{E}[Z^{(0)}]\text{var}(Z^{(0)}) - 3\mathbb{E}[Z^{(0)}]\text{var}(V) - \mathbb{E}[Z^{(0)}]^3}{\sigma_Y^3}. \end{aligned} \tag{8.2}$$

Using GMA, the term $\mathbb{E}\left[(Z^{(0)}+V)^3\right]$ can be simplified further using

$$\begin{aligned}\mathbb{E}\left[(Z^{(0)}+V)^3\right] &= \mathbb{E}\left[\left(Z^0\right)^3 + V^3 + 3Z^0V^2 + 3V\left(Z^0\right)^2\right], \\ &= \mathbb{E}\left[\left(Z^0\right)^3\right] + \mathbb{E}[V^3] + 3\mathbb{E}[Z^0V^2] + \cancelto{0}{3\mathbb{E}[VZ^0]}, \\ &= \mathbb{E}[Z^0]^3 + 3\mathbb{E}[Z^0]\text{var}(Z^0) + \mathbb{E}[V^3] + 3\mathbb{E}[Z^0]\text{var}(V). \end{aligned} \tag{8.3}$$

Using Equations 8.2 & 8.3, we re-write the expression for skewness as

$$\begin{aligned}\tilde{\mu}_3 &= \frac{\mathbb{E}\left[(Z^{(0)}+V)^3 - 3\mathbb{E}[Z^{(0)}]\text{var}(Z^{(0)}) - 3\mathbb{E}[Z^{(0)}]\text{var}(V) - \mathbb{E}[Z^{(0)}]^3\right]}{\sigma_Y^3}, \\ &= \frac{\cancel{\mathbb{E}[Z^0]^3} + \cancel{3\mathbb{E}[Z^0]\text{var}(Z^0)} + \mathbb{E}[V^3] + \cancel{3\mathbb{E}[Z^0]\text{var}(V)} - \cancel{3\mathbb{E}[Z^{(0)}]\text{var}(Z^{(0)})} - \cancel{3\mathbb{E}[Z^{(0)}]\text{var}(V)} - \cancel{\mathbb{E}[Z^{(0)}]^3}}{\sigma_Y^3}, \\ \tilde{\mu}_3 &= \frac{\mathbb{E}[V^3]}{\sigma_Y^3}. \end{aligned} \tag{8.4}$$

Hence, the skewness of Y is a function of $\mathbb{E}[V^3]$ and σ_Y. The first-step for inferring the skewness, would be to obtain the closed-form moments for the cube of the Gaussian random variable V, i.e., $\mathbb{E}[V^3]$ and $\text{var}(V^3)$. Considering that term $\mathbb{E}[V^3]$ is represented by the random

variable $\overline{V^3}$, the next step would be to formulate the prediction and update equations similarly to the AGVI method described in Section 5.2.2 of Chapter 5, but for the posterior inference of $\overline{V^3}$.

8.3.2 Time-Varying Process Error's Variance Inference

In the current AGVI formulation, the process error's variance term is considered to be stationary over time. The transition model for the random variable representing the error's variance, i.e., $\overline{W^2}$ is constant from one time step to the next as given by

$$\overline{w_t^2} = \overline{w_{t-1}^2}\,. \tag{8.5}$$

For modeling time-varying process error's variance, the transition model shown in Equation 8.5 should include the random error term s shown by

$$\overline{w_t^2} = \overline{w_{t-1}^2} + s_t,\ \ s : S \sim \mathcal{N}(0, \sigma_S^2),$$

where σ_S^2 is the variance parameter associated with the process error s. This variance term can also be inferred with the current AGVI formulation. Let us represent the error's variance σ_S^2 by the Gaussian random variable $\overline{S^2}$. Similarly to the procedure shown in Section 5.2.2, the Gaussian random variables S^2 and $\overline{S^2}$ can be expressed in terms of the standard Gaussian variable ϵ and ζ shown by

$$\begin{aligned} S^2 &= \overline{S^2} + \sqrt{2}\,\overline{S^2}\epsilon, \quad \epsilon \sim \mathcal{N}(0,1) && (8.6)\\ \overline{S^2} &= \mu^{\overline{S^2}} + \sigma^{\overline{S^2}}\zeta, \quad \zeta \sim \mathcal{N}(0,1). && (8.7) \end{aligned}$$

Starting from the prior knowledge for $\overline{S^2}$, the first objective is to obtain the marginalized moments for S^2, thereby providing the variance for S which would be equal to $\mu^{\overline{S^2}}$. Using the prior knowledge for $\overline{W^2}$ and the variance term $\mu^{\overline{S^2}}$, the mean and variance for the prior predictive PDF of $\overline{W^2_{t|t-1}}$ are

$$\begin{aligned} \mathbb{E}[\overline{W^2_{t|t-1}}] &= \mu^{\overline{W^2}}_{t-1|t-1},\\ \text{var}(\overline{W^2_{t|t-1}}) &= (\sigma^{\overline{W^2}}_{t-1|t-1})^2 + \mu^{\overline{S^2}}_{t|t-1}. \end{aligned}$$

Using these moments, the same procedure taken for the AGVI method can be followed to obtain the posterior PDF of $\overline{W^2}$ at any time t. Thereafter, the next objective would be to formulate the update step for obtaining the posterior moments for $\overline{S^2}$ using the updated

knowledge of $\overline{W^2}$. Hence, for the time-varying case, the AGVI method needs to be applied twice for each time step to obtain the posterior knowledge of the time-varying error's variance parameter.

8.4 Concluding Remark

The methods developed in this thesis are fundamental to the engineering community and can be used in many future applications. Using the mathematical concepts developed in this thesis, structured ideas can already be formulated as shown by the future research section. Nevertheless, the limitations section confirms that these methods are still in a nascent stage and that they can be improved significantly in the future.

REFERENCES

[1] P. Izmailov, W. J. Maddox, P. Kirichenko, T. Garipov, D. Vetrov, and A. G. Wilson, "Subspace inference for Bayesian deep learning," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 1169–1179.

[2] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for Bayesian uncertainty in deep learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[3] H. Salimbeni, C.-A. Cheng, B. Boots, and M. Deisenroth, "Orthogonally decoupled variational Gaussian processes," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[4] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Artificial Intelligence and Statistics*. PMLR, 2016, pp. 370–378.

[5] C. Riquelme, G. Tucker, and J. Snoek, "Deep Bayesian bandits showdown," in *International Conference on Learning Representations*, 2018.

[6] Z. Yang, A. Wilson, A. Smola, and L. Song, "A la carte–learning fast kernels," in *Artificial Intelligence and Statistics*. PMLR, 2015, pp. 1098–1106.

[7] J.-A. Goulet, L. H. Nguyen, and S. Amiri, "Tractable approximate Gaussian inference for Bayesian neural networks," *Journal of Machine Learning Research*, vol. 22, no. 251, pp. 1–23, 2021.

[8] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1050–1059.

[9] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 6405–6416.

[10] J. M. Hernández-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1861–1869.

[11] S. Sun, C. Chen, and L. Carin, "Learning structured weight uncertainty in Bayesian neural networks," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1283–1292.

[12] C. Louizos and M. Welling, "Structured and efficient variational deep learning with matrix Gaussian posteriors," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1708–1716.

[13] A. Wu, S. Nowozin, E. Meeds, R. E. Turner, J. M. Hernandez-Lobato, and A. L. Gaunt, "Deterministic variational inference for robust Bayesian neural networks," in *International Conference on Learning Representations*, 2019.

[14] "16^{th} International Benchmark Workshop on Numerical Analysis of Dams," https://icold-bw2022.fgg.uni-lj.si/submission, accessed: 2022-09-22.

[15] P. Adam, G. Sam, C. Soumith, and C. Gregory, "Pytorch," https://github.com/pytorch/pytorch, 2021, accessed: 2022-09-22.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[17] 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.

[18] J.-A. Goulet, "Bayesian dynamic linear models for structural health monitoring," *Structural Control and Health Monitoring*, vol. 24, no. 12, p. e2035, 2017.

[19] A. Gelman, B. Carlin, S. Stern, B. Dunson, A. Vehtari, and B. Rubin, *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[20] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

[21] J.-A. Goulet, *Probabilistic machine learning for civil engineers*. MIT Press, 2020.

[22] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021.

[23] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[24] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? does it matter?" *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009.

[25] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT Press, 2012.

[26] L. H. Nguyen, "Real-time anomaly detection in the behaviour of structures," Ph.D. dissertation, Polytechnique Montréal, 2019.

[27] Z. Hamida, "Stochastic modelling of infrastructures deterioration and interventions based on network-scale visual inspections," Ph.D. dissertation, Polytechnique Montréal, 2020.

[28] H. Nguyen and J.-A. Goulet, "Anomaly detection with the switching Kalman filter for structural health monitoring," *Structural Control and Health Monitoring*, vol. 25, no. 4, p. e2136, 2018.

[29] H. Nguyen, I. Gaudot, S. Khazaeli, and J.-A. Goulet, "A kernel-based method for modeling non-harmonic periodic phenomena in Bayesian dynamic linear models," *Frontiers in Built Environment*, vol. 5, p. 8, 2019.

[30] O. Aguilar and M. West, "Analysis of hospital quality monitors using hierarchical time series models," in *Case Studies in Bayesian Statistics*. Springer, 1999, pp. 287–302.

[31] H. Afshari, A. Gadsden, and S. Habibi, "Gaussian filters for parameter and state estimation: A general review of theory and recent trends," *Signal Processing*, vol. 135, pp. 218–238, 2017.

[32] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[33] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[34] L. Ljung, "Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems," *IEEE Transactions on Automatic Control*, vol. 24, no. 1, pp. 36–50, 1979.

[35] A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. IEEE, 2000, pp. 153–158.

[36] J. Gordon, J. Salmond, and M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE proceedings F (Radar and Signal processing)*. IET, 1993, pp. 107–113.

[37] J. Julier and K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Robotics Research Group, Department of Engineering Science, Tech. Rep., 1996.

[38] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov chain Monte Carlo*. CRC Press, 2011.

[39] Z. Ghahramani and E. Hinton, "Parameter estimation for linear dynamical systems," University of Toronto, Dept. of Computer Science, Tech. Rep., 1996.

[40] H. Haario, E. Saksman, and J. Tamminen, "An adaptive Metropolis algorithm," *Bernoulli*, pp. 223–242, 2001.

[41] M. Vihola, "Robust adaptive Metropolis algorithm with coerced acceptance rate," *Statistics and Computing*, vol. 22, no. 5, pp. 997–1008, 2012.

[42] M. Neal *et al.*, "MCMC using Hamiltonian dynamics," *Handbook of Markov chain Monte Carlo*, vol. 2, no. 11, p. 2, 2011.

[43] C. T. Kelley, *Iterative methods for optimization*. SIAM, 1999.

[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[45] Z. Ghahramani, G. E. Hinton *et al.*, "The EM algorithm for mixtures of factor analyzers," University of Toronto, Dept. of Computer Science, Tech. Rep., 1996.

[46] H. E. Rauch, C. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450 0001–1452, 1965.

[47] I. Arasaratnam and S. Haykin, "Cubature Kalman smoothers," *Automatica*, vol. 47, no. 10, pp. 2245–2250, 2011.

[48] R. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 15, no. 2, pp. 175–184, 1970.

[49] J. Duník, O. Straka, O. Kost, and J. Havlík, "Noise covariance matrices in state-space models: A survey and comparison of estimation methods—part I," *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 11, pp. 1505–1543, 2017.

[50] R. Mehra, "Approaches to adaptive filtering," *IEEE Transactions on Automatic Control*, vol. 17, no. 5, pp. 693–698, 1972.

[51] J. Wang, "Stochastic Modeling for Real-Time Kinematic GPS/GLONASS Positioning," *Navigation*, vol. 46, no. 4, pp. 297–305, 1999.

[52] R. Mehra, "On-line identification of linear dynamic systems with applications to Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 16, no. 1, pp. 12–21, 1971.

[53] K. J. Åström and P. Eykhoff, "System identification—a survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.

[54] H. Nguyen and J.-A. Goulet, "Real-time anomaly detection with Bayesian dynamic linear models," *Structural Control and Health Monitoring*, p. e2404, 2019.

[55] W. Mader, Y. Linke, M. Mader, L. Sommerlade, J. Timmer, and B. Schelter, "A numerically efficient implementation of the expectation maximization algorithm for state space models," *Applied Mathematics and Computation*, vol. 241, pp. 222–232, 2014.

[56] J. W. Graham, P. E. Cumsille, and A. E. Shevock, "Methods for handling missing data." *Handbook of Psychology: Research Methods in Psychology*, pp. 109–141, 2013.

[57] K. Murphy and S. Russell, "Rao-blackwellised particle filtering for dynamic Bayesian networks," in *Sequential Monte Carlo Methods in Practice*. Springer, 2001, pp. 499–515.

[58] G. Grisettiyz, C. Stachniss, and W. Burgard, "Improving grid-based slam with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 2432–2437.

[59] P. R. Belanger, "Estimation of noise covariance matrices for a linear time-varying stochastic process," *Automatica*, vol. 10, no. 3, pp. 267–275, 1974.

[60] J. Zhou and R. Luecke, "Estimation of the covariances of the process noise and measurement noise for a linear discrete dynamic system," *Computers & Chemical Engineering*, vol. 19, no. 2, pp. 187–195, 1995.

[61] B. J. Odelson, M. R. Rajamani, and J. B. Rawlings, "A new autocovariance least-squares method for estimating noise covariances," *Automatica*, vol. 42, no. 2, pp. 303–308, 2006.

[62] J. Duník, O. Straka, and O. Kost, "Measurement difference autocovariance method for noise covariance matrices estimation," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 365–370.

[63] K. Myers and B. Tapley, "Adaptive sequential estimation with unknown noise statistics," *IEEE Transactions on Automatic Control*, vol. 21, no. 4, pp. 520–523, 1976.

[64] P. Sage and W. Husa, "Algorithms for sequential adaptive estimation of prior statistics," in *1969 IEEE Symposium on Adaptive Processes (8th) Decision and Control*. IEEE, 1969, pp. 61–61.

[65] A. Assa and K. N. Plataniotis, "Adaptive Kalman filtering by covariance sampling," *IEEE Signal Processing Letters*, vol. 24, no. 9, pp. 1288–1292, 2017.

[66] R. H. Shumway, D. S. Stoffer, and D. S. Stoffer, *Time series analysis and its applications*. Springer, 2000.

[67] R. Kashyap, "Maximum likelihood identification of stochastic linear systems," *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 25–34, 1970.

[68] S. Sarkka and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Transactions on Automatic control*, vol. 54, no. 3, pp. 596–600, 2009.

[69] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin, "On particle methods for parameter estimation in state-space models," *Statistical Science*, vol. 30, no. 3, pp. 328–351, 2015.

[70] Y. Huang, F. Zhu, G. Jia, and Y. Zhang, "A slide window variational adaptive Kalman filter," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3552–3556, 2020.

[71] L. Chin, "Advances in adaptive filtering," *Advanced in Theory and Applications*, vol. 15, pp. 277–356, 1979.

[72] T. Berry and T. Sauer, "Adaptive ensemble Kalman filtering of non-linear systems," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 65, no. 1, p. 20331, 2013.

[73] V. A. Bavdekar, A. P. Deshpande, and S. C. Patwardhan, "Identification of process and measurement noise covariance for state and parameter estimation using extended Kalman filter," *Journal of Process Control*, vol. 21, no. 4, pp. 585–601, 2011.

[74] T. Kontoroupi and A. W. Smyth, "Online noise identification for joint state and parameter estimation of nonlinear systems," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 2, no. 3, p. B4015006, 2016.

[75] G. Storvik, "Particle filters for state-space models with the presence of unknown static parameters," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 281–289, 2002.

[76] X. R. Li and Y. Bar-Shalom, "A recursive multiple model approach to noise identification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 3, pp. 671–684, 1994.

[77] S. Särkkä and J. Hartikainen, "Non-linear noise adaptive Kalman filtering via variational Bayes," in *2013 IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 2013, pp. 1–6.

[78] Y. Huang, Y. Zhang, Z. Wu, N. Li, and J. Chambers, "A novel adaptive Kalman filter with inaccurate process and measurement noise covariance matrices," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 594–601, 2017.

[79] J. Ma, H. Lan, Z. Wang, X. Wang, Q. Pan, and B. Moran, "Improved adaptive Kalman filter with unknown process noise covariance," in *2018 21st International Conference on Information Fusion*. IEEE, 2018, pp. 1–5.

[80] T. Ardeshiri, E. Özkan, U. Orguner, and F. Gustafsson, "Approximate Bayesian smoothing with unknown process and measurement noise covariances," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2450–2454, 2015.

[81] X. Dong, L. Chisci, and Y. Cai, "An adaptive variational Bayesian filter for nonlinear multi-sensor systems with unknown noise statistics," *Signal Processing*, vol. 179, p. 107837, 2021.

[82] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012.

[83] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, University of Cambridge, 2016.

[84] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proceedings of the 28th International Conference on Machine Learning*. Citeseer, 2011, pp. 681–688.

[85] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, no. 1, pp. 5–43, 2003.

[86] D. J. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Computation*, vol. 4, no. 3, pp. 448–472, 1992.

[87] A. Graves, "Practical variational inference for neural networks," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, 2011, p. 2348–2356.

[88] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[89] A. K. Gupta and D. K. Nagar, *Matrix variate distributions*. Chapman and Hall/CRC, 2018.

[90] H. E. Robbins, "An empirical Bayes approach to statistics," in *Breakthroughs in Statistics*. Springer, 1992, pp. 388–394.

[91] B. Deka, L. Ha Nguyen, S. Amiri, and J.-A. Goulet, "The Gaussian multiplicative approximation for state-space models," *Structural Control and Health Monitoring*, p. e2904, 2021.

[92] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[93] X. Glorot and Y. Bengio, "Xavier initialization," *Journal of Machine Learning Research, 2010b. ISSN*, vol. 15324435, 2010.

[94] B. Efron, *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*. Cambridge University Press, 2012.

[95] L.-H. Nguyen and J.-A. Goulet, "Analytically tractable hidden-states inference in Bayesian neural networks," *Journal of Machine Learning Research*, vol. 23, no. 50, pp. 1–33, 2022.

[96] M. G. Bulmer, *Principles of statistics*. MIT Press, 1979.

[97] C. Vignat, "A generalized Isserlis theorem for location mixtures of Gaussian random vectors," *Statistics & Probability Letters*, vol. 82, no. 1, pp. 67–71, 2012.

[98] A. H. Jazwinski, *Stochastic processes and filtering theory*. Academic Press, 1970.

[99] I. Arasaratnam, S. Haykin, and R. Hurd, "Cubature Kalman filtering for continuous-discrete systems: theory and simulations," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 4977–4993, 2010.

[100] K. Roushangar, S. Garekhani, and F. Alizadeh, "Forecasting daily seepage discharge of an earth dam using wavelet–mutual information–Gaussian process regression approaches," *Geotechnical and Geological Engineering*, vol. 34, no. 5, pp. 1313–1326, 2016.

[101] S. Chen, C. Gu, C. Lin, Y. Wang, and M. A. Hariri-Ardebili, "Prediction, monitoring, and interpretation of dam leakage flow via adaptative kernel extreme learning machine," *Measurement*, vol. 166, p. 108161, 2020.

[102] T. Wang, D. Perissin, F. Rocca, and M.-S. Liao, "Three gorges dam stability monitoring with time-series insar image analysis," *Science China Earth Sciences*, vol. 54, no. 5, pp. 720–732, 2011.

[103] B. Dagum, "Time series modeling and decomposition," *Statistica*, vol. 70, no. 4, pp. 433–457, 2010.

[104] L. H. Nguyen, I. Gaudot, and J.-A. Goulet, "Uncertainty quantification for model parameters and hidden state variables in Bayesian dynamic linear models," *Structural Control and Health Monitoring*, vol. 26, no. 3, p. e2309, 2019.

[105] E. Cross, K. Koo, J. Brownjohn, and K. Worden, "Long-term monitoring and data analysis of the tamar bridge," *Mechanical Systems and Signal Processing*, vol. 35, no. 1-2, pp. 16–34, 2013.

[106] 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.

[107] K. Malm R, A. M Simon, S. F, and H. R, "Summary of Theme A: Behaviour prediction of a concrete arch dam," in 16^{th} *International Benchmark Workshop on Numerical Analysis of Dams*. International Commission on Large Dams, 2022.

[108] F. Salazar, M. Á. Toledo, E. Oñate, and B. Suárez, "Interpretation of dam deformation and leakage with boosted regression trees," *Engineering Structures*, vol. 119, pp. 230–251, 2016.

[109] Z. P. Bažant and S. Wu, "Rate-type creep law of aging concrete based on Maxwell chain," *Matériaux et Construction*, vol. 7, no. 1, pp. 45–60, 1974.

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

[111] A. E. Bryson, "Applied optimal control: Optimization," *Estimization and Control*, vol. 2, 1975.

[112] B. Laurent, "Analytical inference for visual inspection uncertainty in the context of transportation infrastructures," Master's thesis, Polytechnique Montréal, 2022.

[113] I. Gaudot, L. Nguyen, K. S., and J.-A. Goulet, "OpenBDLM, an Open-Source Software for Structural Health Monitoring using Bayesian Dynamic Linear Models," https://github.com/CivML-PolyMtl/OpenBDLM, 2019, accessed: 2022-09-22.

[114] J. Mukhoti, P. Stenetorp, and Y. Gal, "On the importance of strong baselines in Bayesian deep learning," in *NeurIPS, Workshop on Bayesian Deep Learning*, 2018.

[115] S. Coles, J. Bawa, L. Trenner, and P. Dorazio, *An introduction to statistical modeling of extreme values*. Springer, 2001.

[116] I. Song and S. Lee, "Explicit formulae for product moments of multivariate Gaussian random variables," *Statistics & Probability Letters*, vol. 100, pp. 27–34, 2015.

[117] P. H. M. Janssen and P. Stoica, "On the expectation of the product of four matrix-valued Gaussian random variables," *IEEE Transactions on Automatic Control*, vol. 33, no. 9, pp. 867–870, 1988.

[118] J. M. Hernández-Lobato and R. Adams, "Probabilistic-backpropagation," https://github.com/HIPS/Probabilistic-Backpropagation, 2016, accessed: 2022-09-22.

[119] J. Mukhoti, P. Stenetorp, and Y. Gal, "Dropoutuncertaintyexps," https://github.com/yaringal/DropoutUncertaintyExps, 2018, accessed: 2022-09-22.

[120] F. Chollet, "Keras: Deep learning for humans," https://github.com/fchollet/keras, 2015, accessed: 2022-09-22.

[121] A. Wu, S. Nowozin, E. Meeds, R. E. Turner, J. M. Hernández-Lobato, and A. L. Gaunt, "Deterministic variational inference," https://github.com/microsoft/deterministic-variational-inference, 2018, accessed: 2022-09-22.

[122] J.-A. Goulet, L. H. Nguyen, and S. Amiri, "TAGI," https://github.com/CivML-PolyMtl/TAGI, 2021, accessed: 2022-09-22.

APPENDIX A

A.1 The GMA Equations Using Gaussian Moment Generating Function

Let $\boldsymbol{X} = [X_1 \dots X_p]^\intercal$ be a vector of Gaussian random variables, $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is the mean vector, $\boldsymbol{\Sigma}$ is the covariance matrix, and $\mathbf{t} = [t_1 \dots t_p]^\intercal \in \mathbb{R}^p$, then the following Equation [116,117] is held which analytically computes the multivariate moments encountered in the nonlinear Kalman filter given by

$$\mathbb{E}[X_1^{k_1} \dots X_p^{k_p}] = \frac{\partial^k}{\partial t^{k_1} \dots \partial t^{k_p}} \exp\left(\sum_{i=1}^{p} t_i \mu_i + \frac{1}{2} \sum_{i,j=1}^{p} t_i t_j \mathrm{cov}(X_i, X_j)\right)\Big|_{t_1=\dots=t_p=0} \quad \text{(A.1)}$$

where k_i's are non-negative integers and $k = \sum_{i=1}^{p} k_i$. This Equation is derived from the moment generating function of multivariate Gaussian;

$$\begin{aligned} \frac{\partial^k}{\partial t_1^{k_1} \dots \partial t_p^{k_p}} M_{\boldsymbol{x}}(\mathbf{t}^\intercal) &= \frac{\partial^k}{\partial t_1^{k_1} \dots \partial t_p^{k_p}} \mathbb{E}\left[\exp\left(\sum_{i=1}^{p} t_i X_i\right)\right] \\ &= \mathbb{E}\left[\frac{\partial^k}{\partial t^{k_1} \dots \partial t^{k_p}} \exp\left(\sum_{i=1}^{p} t_i X_i\right)\right] \\ &= \mathbb{E}\left[X_1^{k_1} \dots X_p^{k_p} \exp\left(\sum_{i=1}^{p} t_i X_i\right)\right] \end{aligned} \quad \text{(A.2)}$$

Setting $\mathbf{t} = [0 \dots 0]^\intercal$, we obtain

$$\mathbb{E}[X_1^{k_1} \dots X_p^{k_p}] = \frac{\partial^k}{\partial t^{k_1} \dots \partial t^{k_p}} M_{\boldsymbol{x}}(\mathbf{t}^\intercal)$$

Given Gaussian random variables, $M_{\boldsymbol{x}}(\mathbf{t}^\intercal) = \mathbb{E}[e^{\mathbf{t}^\intercal \boldsymbol{x}}] = e^{\mathbf{t}^\intercal \boldsymbol{\mu} + \frac{1}{2}\mathbf{t}^\intercal \boldsymbol{\Sigma} \mathbf{t}}$, the following equations to evaluate product terms can be directly obtained from Equation A.1.

$$\begin{aligned} \mathbb{E}[X_1 X_2] &= \mu_1\mu_2 + \mathrm{cov}(X_1, X_2), \\ \mathbb{E}[X_1 X_2 X_3] &= \mathrm{cov}(X_1, X_2)\mu_3 + \mathrm{cov}(X_1, X_3)\mu_2 + \mathrm{cov}(X_2, X_3)\mu_1 + \mu_1\mu_2\mu_3, \\ \mathbb{E}[X_1 X_2 X_3 X_4] &= \mathrm{cov}(X_1X_2)\Big(\mathrm{cov}(X_3, X_4) + \mu_3\mu_4\Big) + \mathrm{cov}(X_1X_3)\Big(\mathrm{cov}(X_2, X_4) + \mu_2\mu_4\Big) + \\ &\quad \mathrm{cov}(X_2X_3)\Big(\mathrm{cov}(X_1, X_4) + \mu_1\mu_4\Big) + \mathrm{cov}(X_1, X_4)\mu_2\mu_3 + \mathrm{cov}(X_2, X_4)\mu_1\mu_3 \\ &\quad + \mathrm{cov}(X_3, X_4)\mu_1\mu_2 + \mu_1\mu_2\mu_3\mu_4. \end{aligned}$$

A.2 The GMA Equations Using 2^{nd} Order Taylor Series Expansion

Let us consider the function $h(\cdot)$ in two variables x_1 and x_2, where $h(x_1, x_2) = x_1 x_2$ that represents the product of two random variables. Using 2^{nd} order Taylor series expansion, we express $h(x_1, x_2)$ as

$$\begin{aligned} h(x_1,x_2) &\approx h(\mu_1,\mu_2) + \frac{\partial h}{\partial x_1}\Big|_{\mu_1,\mu_2}(x_1-\mu_1) + \frac{\partial h}{\partial x_2}\Big|_{\mu_1,\mu_2}(x_2-\mu_2) + 0.5\cdot\frac{\partial^2 h}{\partial x_1^2}\Big|_{\mu_1,\mu_2}(x_1-\mu_1)^2 \\ &\quad + 0.5\cdot\frac{\partial^2 h}{\partial x_2^2}\Big|_{\mu_1,\mu_2}(x_2-\mu_2)^2 + \cdot\frac{\partial^2 h}{\partial x_1 \partial x_2}\Big|_{\mu_1,\mu_2}(x_1-\mu_1)(x_2-\mu_2), \\ &\approx \mu_1\mu_2 + \mu_2(x_1-\mu_1) + \mu_1(x_2-\mu_2) + (x_1-\mu_1)(x_2-\mu_2), \end{aligned} \tag{A.3}$$

where $h(\mu_1, \mu_2) = \mu_1\mu_2$, $\frac{\partial h}{\partial x_1}\Big|_{\mu_1,\mu_2} = \mu_2$, $\frac{\partial h}{\partial x_2}\Big|_{\mu_1,\mu_2} = \mu_1$, and $\frac{\partial^2 h}{\partial x_1 \partial x_2}\Big|_{\mu_1,\mu_2} = 1$. Using Equation A.3, the expected value $\mathbb{E}[X_1X_2]$ is

$$\begin{aligned} \mathbb{E}[X_1X_2] &= \mu_1\mu_2 + \mathbb{E}[(X_1-\mu_1)(X_2-\mu_2)], \\ &= \mu_1\mu_2 + \text{cov}(X_1,X_2), \end{aligned} \tag{A.4}$$

where using the properties of random variables $\mathbb{E}[(X_1-\mu_1)(X_2-\mu_2)] = \text{cov}(X_1, X_2)$. The variance term $\text{var}(X_1X_2)$ is given by

$$\begin{aligned} \text{var}(X_1X_2) &= \text{var}\Big(\mu_1\mu_2 + \mu_2(X_1-\mu_1) + \mu_1(X_2-\mu_2) + (X_1-\mu_1)(X_2-\mu_2)\Big), \\ &= \text{var}\Big(\mu_2(X_1-\mu_1)\Big) + \text{var}\Big(\mu_1(X_2-\mu_2)\Big) + \text{var}\Big((X_1-\mu_1)(X_2-\mu_2)\Big) \\ &\quad + 2\text{cov}\Big(\mu_2(X_1-\mu_1), \mu_1(X_2-\mu_2)\Big) + 2\text{cov}\Big(\mu_2(X_1-\mu_1), (X_1-\mu_1)(X_2-\mu_2)\Big) \\ &\quad + 2\text{cov}\Big((x_1-\mu_1)(X_2-\mu_2), \mu_1(X_2-\mu_2)\Big), \\ &= \mu_2^2\sigma_1^2 + \mu_1^2\sigma_2^2 + \sigma_1^2\sigma_2^2 + \text{cov}(X_1,X_2)^2 + 2\mu_1\mu_2\text{cov}(X_1,X_2), \end{aligned} \tag{A.5}$$

where the terms in Equation A.5 are evaluated as follows

$$\begin{aligned} \text{var}\Big((X_1-\mu_1)(X_2-\mu_2)\Big) &= \mathbb{E}[(X_1-\mu_1)^2(X_2-\mu_2)^2] - \mathbb{E}[(X_1-\mu_1)(X_2-\mu_2)]^2, \\ &= \sigma_1^2\sigma_2^2 + 2\text{cov}(X_1,X_2)^2 - \text{cov}(X_1,X_2)^2, \\ &= \sigma_1^2\sigma_2^2 + \text{cov}(X_1,X_2)^2, \end{aligned} \tag{A.6}$$

where using Isserlis theorem [97], $\mathbb{E}[(X_1-\mu_1)^2(X_2-\mu_2)^2] = \sigma_1^2\sigma_2^2 + 2\text{cov}(X_1,X_2)^2$. Similarly, the covariance terms in Equation A.5 are given by

$$\begin{aligned}
2\text{cov}\Big(\mu_2(X_1-\mu_1), \mu_1(X_2-\mu_2)\Big) &= 2\mu_1\mu_2\text{cov}(X_1,X_2), && \text{(A.7)}\\
2\text{cov}\Big(\mu_2(X_1-\mu_1), (X_1-\mu_1)(X_2-\mu_2)\Big) &= 2\mu_2\Big(\mathbb{E}[(X_1-\mu_1)^2(X_2-\mu_2)] \\
&\quad - \cancelto{0}{\mathbb{E}[X_1-\mu_1]}\mathbb{E}[(X_1-\mu_1)(X_2-\mu_2)]\Big) \\
&= 0, && \text{(A.8)}\\
2\text{cov}\Big(\mu_1(X_2-\mu_2), (X_1-\mu_1)(X_2-\mu_2)\Big) &= 2\mu_1\Big(\mathbb{E}[(X_2-\mu_2)^2(X_1-\mu_1)] \\
&\quad - \cancelto{0}{\mathbb{E}[X_2-\mu_2]}\mathbb{E}[(X_1-\mu_1)(X_2-\mu_2)]\Big), \\
&= 0, && \text{(A.9)}
\end{aligned}$$

where using Isserlis theorem, $\mathbb{E}[(X_1-\mu_1)^2(X_2-\mu_2)] = \mathbb{E}[(X_2-\mu_2)^2(X_1-\mu_1)] = 0$ as the expected values of the product of odd powers of Gaussian random variables of zero-mean values are 0. The covariance between X_3 and X_1X_2 is

$$\begin{aligned}
\text{cov}(X_3, X_1X_2) &= \text{cov}\Big(X_3, \mu_1\mu_2 + \mu_2(X_1-\mu_1) + \mu_1(X_2-\mu_2) + (X_1-\mu_1)(X_2-\mu_2)\Big), \\
&= \text{cov}\Big(X_3, \mu_2(X_1-\mu_1)\Big) + \text{cov}\Big(X_3, \mu_1(X_2-\mu_2)\Big) \\
&\quad + \text{cov}\Big(X_3, (X_1-\mu_1)(X_2-\mu_2)\Big), \\
&= \mu_2\text{cov}(X_3,X_1) + \mu_1\text{cov}(X_3,X_2),
\end{aligned}$$

where similar to Equations A.8 and A.9, the term $\text{cov}\Big(X_3, (X_1-\mu_1)(X_2-\mu_2)\Big) = 0$ and evaluated as follows

$$\begin{aligned}
\text{cov}\Big(X_3, (X_1-\mu_1)(X_2-\mu_2)\Big) &= \mathbb{E}[X_3(X_1-\mu_1)(X_2-\mu_2)] - \mathbb{E}[X_3]\mathbb{E}[(X_1-\mu_1)(X_2-\mu_2)], \\
&= \mathbb{E}[(X_3-\mu_3+\mu_3)(X_1-\mu_1)(X_2-\mu_2)] - \mu_3\text{cov}(X_1,X_2), \\
&= \mathbb{E}[(X_3-\mu_3)(X_1-\mu_1)(X_2-\mu_2)] + \mathbb{E}[(\mu_3)(X_1-\mu_1)(X_2-\mu_2)] \\
&\quad - \mu_3\text{cov}(X_1,X_2), \\
&= 0 + \mu_3\text{cov}(X_1,X_2) - \mu_3\text{cov}(X_1,X_2), \\
&= 0.
\end{aligned}$$

The covariance between the product terms X_1X_2 and X_3X_4 is given by

$$\begin{aligned}
\text{cov}(X_1X_2, X_3X_4) &= \text{cov}\Big(\mu_1\mu_2 + \mu_2(X_1-\mu_1) + \mu_1(X_2-\mu_2) + (X_1-\mu_1)(X_2-\mu_2), \\
&\quad \mu_3\mu_4 + \mu_4(X_3-\mu_3) + \mu_3(X_4-\mu_4) + (X_3-\mu_3)(X_4-\mu_4)\Big), \\
&= \text{cov}\Big(\mu_2(X_1-\mu_1), \mu_4(X_3-\mu_3)\Big) + \text{cov}\Big(\mu_2(X_1-\mu_1), \mu_3(X_4-\mu_4\Big) \\
&\quad + \text{cov}\Big(\mu_2(X_1-\mu_1), (X_3-\mu_3)(X_4-\mu_4)\Big) \\
&\quad + \text{cov}\Big(\mu_1(X_2-\mu_2), \mu_4(X_3-\mu_3)\Big) + \text{cov}\Big(\mu_1(X_2-\mu_2), \mu_3(X_4-\mu_4)\Big) \\
&\quad + \text{cov}\Big(\mu_1(X_2-\mu_2), (X_3-\mu_3)(X_4-\mu_4)\Big) \\
&\quad + \text{cov}\Big((X_1-\mu_1)(X_2-\mu_2), \mu_4(X_3-\mu_3)\Big) \\
&\quad + \text{cov}\Big((X_1-\mu_1)(X_2-\mu_2), \mu_3(X_4-\mu_4)\Big) \\
&\quad + \text{cov}\Big((X_1-\mu_1)(X_2-\mu_2), (X_3-\mu_3)(X_4-\mu_4)\Big), \\
&= \text{cov}(X_1,X_3)\text{cov}(X_2,X_4) + \text{cov}(X_1,X_4)\text{cov}(X_2,X_3) + \mu_2\mu_4\text{cov}(X_1,X_3) \\
&\quad + \mu_2\mu_3\text{cov}(X_1,X_4) + \mu_1\mu_4\text{cov}(X_2,X_3) + \mu_1\mu_3\text{cov}(X_2,X_4), \qquad \text{(A.10)}
\end{aligned}$$

where the terms in Equation A.10 are evaluated as follows

$$\begin{aligned}
\text{cov}\Big(\mu_2(X_1-\mu_1), \mu_4(X_3-\mu_3)\Big) &= \mu_2\mu_4\text{cov}(X_1,X_3), \\
\text{cov}\Big(\mu_2(X_1-\mu_1), \mu_3(X_4-\mu_4\Big) &= \mu_2\mu_3\text{cov}(X_1,X_4), \\
\text{cov}\Big(\mu_2(X_1-\mu_1), (X_3-\mu_3)(X_4-\mu_4)\Big) &= \mu_2\text{cov}\Big((X_1-\mu_1), (X_3-\mu_3)(X_4-\mu_4)\Big), \\
&= \mu_2\mathbb{E}[(X_1-\mu_1)(X_3-\mu_3)(X_4-\mu_4)] \\
&\quad - \mathbb{E}[(X_1-\mu_1)]\mathbb{E}[(X_3-\mu_3)(X_4-\mu_4)], \\
&= 0,
\end{aligned}$$

$$
\begin{aligned}
\mathrm{cov}\Big(\mu_1(X_2-\mu_2),\mu_4(X_3-\mu_3)\Big) &= \mu_1\mu_4\mathrm{cov}(X_2,X_3),\\
\mathrm{cov}\Big(\mu_1(X_2-\mu_2),\mu_3(X_4-\mu_4)\Big) &= \mu_1\mu_3\mathrm{cov}(X_2,X_4),\\
\mathrm{cov}\Big(\mu_1(X_2-\mu_2),(X_3-\mu_3)(X_4-\mu_4)\Big) &= 0,\\
\mathrm{cov}\Big((X_1-\mu_1)(X_2-\mu_2),\mu_4(X_3-\mu_3)\Big) &= 0,\\
\mathrm{cov}\Big((X_1-\mu_1)(X_2-\mu_2),\mu_3(X_4-\mu_4)\Big) &= 0,\\
\mathrm{cov}\Big((X_1-\mu_1)(X_2-\mu_2),(X_3-\mu_3)(X_4-\mu_4)\Big) &= \mathbb{E}[(X_1-\mu_1)(X_2-\mu_2)(X_3-\mu_3)(X_4-\mu_4)]\\
&\quad -\mathbb{E}[(X_1-\mu_1)(X_2-\mu_2)]\mathbb{E}[(X_3-\mu_3)(X_4-\mu_4)],\\
&= \mathrm{cov}(X_1,X_2)\mathrm{cov}(X_3,X_4)\\
&\quad +\mathrm{cov}(X_1,X_3)\mathrm{cov}(X_2,X_4)\\
&\quad +\mathrm{cov}(X_1,X_4)\mathrm{cov}(X_2,X_3)\\
&\quad -\mathrm{cov}(X_1,X_2)\mathrm{cov}(X_3,X_4),\\
&= \mathrm{cov}(X_1,X_3)\mathrm{cov}(X_2,X_4)\\
&\quad +\mathrm{cov}(X_1,X_4)\mathrm{cov}(X_2,X_3),
\end{aligned}
$$

where using Isserlis theorem [97], the expected value of the product of centered-Gaussian random variables $X_i, X_j, X_k, \text{and} X_n$ is $\mathbb{E}[X_iX_jX_kX_n] = \sigma_{ij}\sigma_{kn} + \sigma_{ik}\sigma_{jn} + \sigma_{in}\sigma_{jk}$, considering $\sigma_{ij} = \mathrm{cov}(X_i, X_j)$.

A.3 Model Matrices for the Trend Multiplicative

$$
\begin{aligned}
\mathbf{A} &= \text{blockdiag}\left(\begin{bmatrix}1 & \Delta t\\ 0 & 1\end{bmatrix}, \begin{bmatrix}\cos\omega & \sin\omega\\ -\sin\omega & \cos\omega\end{bmatrix}, \begin{bmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}, \begin{bmatrix}1 & \Delta t & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\right),\\
\mathbf{C} &= [1\,0\,0\,0\,1\,0\,0\,0\,0\,1],\\
\mathbf{R} &= \sigma_V^2,\\
\mathbf{Q} &= \text{blockdiag}\left((\sigma_W^{\mathtt{LT}})^2\begin{bmatrix}\frac{\Delta t^3}{3} & \frac{\Delta t^2}{2}\\ \frac{\Delta t^2}{2} & \Delta t\end{bmatrix}, (\sigma_W^{\mathtt{S}})^2\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, (\sigma_W^{\mathtt{AR}})^2\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}, (\sigma_W^{\mathtt{TP}})^2\begin{bmatrix}\frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0\\ \frac{\Delta t^2}{2} & \Delta t & 0\\ 0 & 0 & 0\end{bmatrix}\right).
\end{aligned}
$$

where frequency $\omega = \dfrac{2\pi\Delta t}{p}$ with $p = 365.24$ days and $\Delta t = 1$ day. The prediction and update steps in the Kalman filter are given by

$$[\tilde{\boldsymbol{\mu}}_{t|t-1}, \tilde{\boldsymbol{\Sigma}}_{t|t-1}] = \text{Predict}\left(\boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1}, \mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}\right),$$

$$[\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}] = \text{Update}\left(\tilde{\boldsymbol{\mu}}_{t|t-1}, \tilde{\boldsymbol{\Sigma}}_{t|t-1}, \mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}\right).$$

A.4 Model Matrices for the Double Kernel Regression

$$\begin{aligned}
\mathbf{A} &= \text{blockdiag}\left(1, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & \tilde{k}^{\texttt{KR}_1}(t, \mathbf{t}^{\texttt{KR}}) \\ \mathbf{0}_{50\times 1} & \mathbf{I}_{50\times 50} \end{bmatrix}, \begin{bmatrix} 0 & \tilde{k}^{\texttt{KR}_2}(t, \mathbf{t}^{\texttt{KR}}) \\ \mathbf{0}_{30\times 1} & \mathbf{I}_{30\times 30} \end{bmatrix}, 0\right), \\
\mathbf{C} &= [1\ 1\ 0\ 0\ \mathbf{0}_{1\times 51}\ \mathbf{0}_{1\times 31}\ 1], \\
\mathbf{R} &= \sigma_V^2, \\
\mathbf{Q} &= \text{blockdiag}\left((\sigma_W^{\texttt{LL}})^2, (\sigma_W^{\texttt{AR}})^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{0}_{51\times 51}, \mathbf{0}_{31\times 31}, 0\right).
\end{aligned}$$

A.5 Computational Complexity

Algorithm 3 Kalman filter algorithm with the GMA

Input: $\boldsymbol{\mu}_{t-1|t-1}$,$\boldsymbol{\Sigma}_{t-1|t-1}$
Output: $\boldsymbol{\mu}_{t|t}$,$\boldsymbol{\Sigma}_{t|t}$

1: $\boldsymbol{\mu}_{t|t-1} = \mathbf{A}\tilde{\boldsymbol{\mu}}_{t-1|t-1}$.
2: $\boldsymbol{\Sigma}_{t|t-1} = \mathbf{A}\tilde{\boldsymbol{\Sigma}}_{t-1|t-1}\mathbf{A}^\intercal + \mathbf{Q}$.
3: $\mathbf{K} = \boldsymbol{\Sigma}_{t-1|t-1}\mathbf{C}^\intercal(\mathbf{C}\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{C}^\intercal + \mathbf{R})^{-1}$.
4: $r_t = y_t - \mathbf{C}\boldsymbol{\mu}_{t|t-1}$.
5: $\boldsymbol{\mu}_{t|t} = \boldsymbol{\mu}_{t|t-1} + \mathbf{K}r_t$.
6: $\boldsymbol{\Sigma}_{t|t} = (\mathbf{I} - \mathbf{KC})\boldsymbol{\Sigma}_{t|t-1}$.

Since the filtering method is recursive, it is enough to determine the computational complexity of a single time step going from $t-1$ to t to evaluate the total complexity of the algorithm. The computational complexity here refers to the time complexity of an algorithm which is denoted by the big $\mathcal{O}$ notation. The time complexity (or from here on complexity) of the matrix operations to be used in algorithm 1 are described as follows.

1. Matrix multiplication: The multiplication of two matrices of size $n \times n$ has a complexity of $\mathcal{O}(\mathbf{n^3})$. In general, matrix multiplication of two matrices of size $n \times m$ and $m \times p$ has a complexity of $\mathcal{O}(\mathbf{mnp})$.

2. Matrix addition: The addition of matrices of size $m \times n$ has a complexity of $\mathcal{O}(\mathbf{mn})$.

3. Algebraic operations: Algebraic operations are considered to have constant complexity or $\mathcal{O}(\mathbf{1})$ as these operations are unaffected by the size of state vector and will be performed in the same time.

Considering the big $\boldsymbol{O}$ notation for the various matrix operations and the size of the state vector to be n, the step-by-step complexity of algorithm 1 is presented.
Step 1: This step consists of multiplication of two matrices, $[\mathbf{A}]_{n\times n}$ and $[\tilde{\boldsymbol{\mu}}]_{n\times 1}$ with a complexity is $\mathcal{O}(\mathbf{n^2})$. The complexity of computing $\tilde{\boldsymbol{\mu}}$ is $\mathcal{O}(\mathbf{n})$. Since, the GMA equations are algebraic operations unaffected by the size of the state vector, these will have a complexity of $\mathcal{O}(\mathbf{1})$. Hence the total complexity in Step 1 is $\mathcal{O}(\mathbf{n^2 + n})$.
Step 2: This step consists of two matrix multiplication and one transposition operation for computing $[\mathbf{A}]_{n\times n}[\tilde{\boldsymbol{\Sigma}}]_{n\times n}[\mathbf{A}]^{\intercal}_{n\times n}$and one matrix addition. The complexity for this step is $\mathcal{O}(\mathbf{2n^3 + n^2})$.
Step 3: This step consists of computing the terms $[\boldsymbol{\Sigma}]_{n\times n}[\mathbf{C}^{\intercal}]_{n\times 1}$ having a complexity of $\mathcal{O}(\mathbf{n^2})$ and $[\mathbf{C}]_{1\times n}[\boldsymbol{\Sigma}]_{n\times n}[\mathbf{C}^{\intercal}]_{n\times 1}$ having a complexity of $\mathcal{O}(\mathbf{2n^2})$. Finally, the total complexity of calculating the Kalman gain is $\mathcal{O}(\mathbf{3n^2 + n})$.
Step 4: This step consists of a matrix multiplication of $[\mathbf{C}]_{1\times n}[\boldsymbol{\mu}]_{n\times 1}$ having a complexity of $\mathcal{O}(\mathbf{n})$.
Step 5: This step consists of a matrix addition of $[\boldsymbol{\mu}]_{n\times 1}$ and multiplication of a matrix by a scalar, $[\mathbf{K}]_{n\times 1}[r]_{1\times 1}$. The total complexity in this step is $\mathcal{O}(\mathbf{2n})$.
Step 6: The final step to compute $\boldsymbol{\Sigma}_{t|t}$ consists of two matrix multiplication and one matrix subtraction. The total complexity in this step is $\mathcal{O}(\mathbf{n^3 + 2n^2})$.
Hence, the total complexity of the Kalman filter algorithm using the GMA is of the order $\equiv$ $\mathcal{O}(\mathbf{3n^3})$.

APPENDIX B

B.1 BDLM Model Structure

The model matrices for the local level LL, local trend LT, kernel regression KR, and autoregressive AR components [21] are as follows:

$$\mathbf{A}^{\mathtt{LL}} = 1,\ \mathbf{A}^{\mathtt{LT}} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix},\ \mathbf{A}^{\mathtt{KR}} = \begin{bmatrix} 0 & \tilde{k}^{\mathtt{KR}}(t, \mathbf{t}^{\mathtt{KR}}) \\ \mathbf{0}_{\mathtt{N}\times 1} & \mathbf{I}_{\mathtt{N}} \end{bmatrix},\ \mathbf{A}^{\mathtt{AR}} = \phi^{\mathtt{AR}}, \tag{B.1}$$

where N represents the number of control-points for kernel regression and $\Delta t = 1$day. The observation matrix $\mathbf{C}$ for these components are given by

$$\mathbf{C}^{\mathtt{LL}} = 1,\ \mathbf{C}^{\mathtt{LT}} = [1\ \ 0],\ \mathbf{C}^{\mathtt{KR}} = [1\ \mathbf{0}_{\mathtt{N}\times 1}],\ \mathbf{C}^{\mathtt{AR}} = 1. \tag{B.2}$$

The process error covariance matrices $\mathbf{Q}$ are

$$\mathbf{Q}^{\mathtt{LL}} = (\sigma_w^{\mathtt{LL}})^2,\ \mathbf{Q}^{\mathtt{LT}} = (\sigma_w^{\mathtt{LT}})^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix},\ \mathbf{Q}^{\mathtt{KR}} = \begin{bmatrix} (\sigma_0^{\mathtt{KR}})^2 & \mathbf{0} \\ \mathbf{0} & (\sigma_1^{\mathtt{KR}})^2 \cdot \mathbf{I}_{\mathtt{N}} \end{bmatrix},\ \mathbf{Q}^{\mathtt{AR}} = (\sigma_w^{\mathtt{AR}})^2 \tag{B.3}$$

The transition matrix for the SR component of size $3\mathtt{N} + 2$ is formulated as

$$\mathbf{A}^{\mathtt{SR}} = \begin{bmatrix} \mathbf{0}_{\mathtt{N}} & 0_{1\times \mathtt{N}} & 0 & 0 & 0_{1\times \mathtt{N}} \\ \vdots & \mathbf{I}_{\mathtt{N}} & 0 & 0 & 0_{1\times \mathtt{N}} \\ \vdots & \dots & 0 & 0 & \mathbf{1}_{1\times \mathtt{N}} \\ \vdots & \dots & \dots & 0 & \mathbf{0}_{1\times \mathtt{N}} \\ sym. & \dots & \dots & \dots & \mathbf{0}_{\mathtt{N}} \end{bmatrix}. \tag{B.4}$$

The observation matrix $\mathbf{C}^{\mathtt{SR}}$ is given by

$$\mathbf{C}^{\mathtt{SR}} = [\mathbf{0}_{\mathtt{N}\times 1}^{\mathtt{T}}\ \mathbf{0}_{\mathtt{N}\times 1}^{\mathtt{T}}\ \ 0\ \ 1\ \ \mathbf{0}_{\mathtt{N}\times 1}^{\mathtt{T}}]. \tag{B.5}$$

No process error covariance matrix is considered for the SR component and is given by $\mathbf{Q}^{\mathtt{SR}} = \mathbf{0}_{3\mathtt{N}+2}$. Using Equations B.1 & B.4, the global transition matrix $\mathbf{A}$ is obtained by

arranging each of the transition matrices block diagonally shown by

$$\mathbf{A} = \text{blockdiag}\left(\overbrace{[\mathbf{A}^{\texttt{LL}},\ \mathbf{A}^{\texttt{KR}},\ \mathbf{A}^{\texttt{AR}}]}^{\texttt{CB2/3}}, \overbrace{[\mathbf{A}^{\texttt{LT}},\ \mathbf{A}^{\texttt{SR}_1}]}^{\texttt{WL1}}, \overbrace{[\mathbf{A}^{\texttt{AR}},\ \mathbf{A}^{\texttt{SR}_2}]}^{\texttt{WL2}}, \overbrace{[\mathbf{A}^{\texttt{AR}}]}^{\texttt{T-MA1}}, \overbrace{[\mathbf{A}^{\texttt{AR}}]}^{\texttt{T-MA7}}, \overbrace{[\mathbf{A}^{\texttt{AR}}]}^{\texttt{T-MA14}}, \overbrace{[\mathbf{A}^{\texttt{AR}}]}^{\texttt{T-MA28}}, \overbrace{[\mathbf{A}^{\texttt{AR}}]}^{\texttt{T-MA54}}\right), \tag{B.6}$$

where `WL1` and `WL2` refers to the average long-term trend and the mean-centered water level, and the nonlinear dependencies are modeled using the $\texttt{SR}_1$ and $\texttt{SR}_2$ components. Using equations B.2 & B.5, the global observation matrix $\mathbf{C}_t$ is given by

$$\mathbf{C} = \text{blockdiag}\left(\overbrace{[\mathbf{C}^{\texttt{LL}},\ \mathbf{C}^{\texttt{KR}},\ \mathbf{C}^{\texttt{AR}}]}^{\texttt{CB2/3}}, \overbrace{[\mathbf{C}^{\texttt{LT}},\ \mathbf{C}^{\texttt{SR}_1}]}^{\texttt{WL1}}, \overbrace{[\mathbf{C}^{\texttt{AR}},\ \mathbf{C}^{\texttt{SR}_2}]}^{\texttt{WL2}}, \overbrace{[\mathbf{C}^{\texttt{AR}}]}^{\texttt{T-MA1}}, \overbrace{[\mathbf{C}^{\texttt{AR}}]}^{\texttt{T-MA7}}, \overbrace{[\mathbf{C}^{\texttt{AR}}]}^{\texttt{T-MA14}}, \overbrace{[\mathbf{C}^{\texttt{AR}}]}^{\texttt{T-MA28}}, \overbrace{[\mathbf{C}^{\texttt{AR}}]}^{\texttt{T-MA54}}\right), \tag{B.7}$$

The $\mathbf{Q}$ and the $\mathbf{R}$ matrices are

$$\mathbf{Q} = \text{blockdiag}\left(\overbrace{[\mathbf{Q}^{\texttt{LL}},\ \mathbf{Q}^{\texttt{KR}},\ \mathbf{Q}^{\texttt{AR}}]}^{\texttt{CB2/3}}, \overbrace{[\mathbf{Q}^{\texttt{LT}},\ \mathbf{Q}^{\texttt{SR}_1}]}^{\texttt{WL1}}, \overbrace{[\mathbf{Q}^{\texttt{AR}},\ \mathbf{Q}^{\texttt{SR}_2}]}^{\texttt{WL2}}, \overbrace{[\mathbf{Q}^{\texttt{AR}}]}^{\texttt{T-MA1}}, \overbrace{[\mathbf{Q}^{\texttt{AR}}]}^{\texttt{T-MA7}}, \overbrace{[\mathbf{Q}^{\texttt{AR}}]}^{\texttt{T-MA14}}, \overbrace{[\mathbf{Q}^{\texttt{AR}}]}^{\texttt{T-MA28}}, \overbrace{[\mathbf{Q}^{\texttt{AR}}]}^{\texttt{T-MA54}}\right), \tag{B.8}$$

$$\mathbf{R} = \text{blockdiag}\left(\overbrace{[(\sigma_{V_1})^2,\ (\sigma_{V_2})^2,\ (\sigma_{V_3})^2}^{\texttt{CB2/3}}, \overbrace{(\sigma_{V_4})^2}^{\texttt{WL1}}, \overbrace{(\sigma_{V_5})^2}^{\texttt{WL2}}, \overbrace{(\sigma_{V_6})^2}^{\texttt{T-MA1}}, \overbrace{(\sigma_{V_7})^2}^{\texttt{T-MA7}}, \overbrace{(\sigma_{V_8})^2}^{\texttt{T-MA14}}, \overbrace{(\sigma_{V_9})^2}^{\texttt{T-MA28}}, \overbrace{(\sigma_{V_{10}})^2}^{\texttt{T-MA54}}]\right), \tag{B.9}$$

where $\sigma_{V_i},\ \forall i = 1:8$ refers to the standard deviation of the observation error V for each of the time series.

APPENDIX C

C.1 Proof for Lemma 1

Proof. Given that W is Gaussian, the moments of W can be derived using a Gaussian moment generating function so that

$$\begin{aligned} \mu_W &= \mathbb{E}[W] = 0, \\ \sigma_W^2 &= \mathbb{E}[(W-\mu_W)^2] = \mathbb{E}[W^2] - \mathbb{E}[W]^2, \\ &= \mathbb{E}[W^2], \\ \mathbb{E}[W^4] &= 3\mathbb{E}[W^2]^2, \end{aligned} \tag{C.1} \tag{C.2}$$

where using Equations C.1 & C.2, we can define the variance of W^2 such that

$$\text{var}(W^2) = \mathbb{E}[(W^4)] - \mathbb{E}[W^2]^2 = 2\mathbb{E}[W^2]^2. \tag{C.3}$$

If we make the approximation that $W^2 \sim \mathcal{N}(w^2; \mu^{W^2}, (\sigma^{W^2})^2)$ is a Gaussian random variable, then the PDF can be fully defined by its mean and variance,

$$\begin{aligned} \mu^{W^2} &= \mathbb{E}[W^2], \\ (\sigma^{W^2})^2 &= \text{var}(W^2) = 2\mathbb{E}[W^2]^2, \end{aligned}$$

where by using Equation C.3, the variance $\text{var}(W^2)$ can also be expressed in terms of the expected value $\mathbb{E}[W^2]$. Hence, the PDF of W^2 only depends on the unknown hyper parameter μ^{W^2} such that

$$\begin{aligned} f(w^2|\mu^{W^2}, (\sigma^{W^2})^2) &\equiv f(w^2|\mu^{W^2}), \\ &= \mathcal{N}(w^2, \mu^{W^2}, 2(\mu^{W^2})^2). \end{aligned} \tag{C.4}$$

□

C.2 Proof for Lemma 2

Proof. Let us consider that the mean parameter μ^{W^2} is described by the random variable $\overline{W^2} : \overline{w^2} \in (0, \infty)$ for which

$$f(\overline{w^2}) \sim \mathcal{N}(\overline{w^2}; \mu^{\overline{W^2}}, (\sigma^{\overline{W^2}})^2). \tag{C.5}$$

Using (C.4) and (C.5), we can rewrite the PDF of W^2 as

$$f(w^2|\overline{w^2}) = \mathcal{N}(w^2; \overline{w^2}, 2(\overline{w^2})^2). \tag{C.6}$$

Using the acyclic graph in Figure 5.1, the joint PDF of $Y_{t|t-1}$, $\boldsymbol{X}_{t|t-1}$, $W^2_{t|t-1}$, and $\overline{W^2_{t|t-1}}$ is shown by

$$\begin{aligned} f(y_t, x_t, w_t^2, \overline{w^2}_t|\boldsymbol{y}_{1:t-1}) = f(y_t|x_t, w_t^2) \cdot f(x_t|\boldsymbol{y}_{1:t-1}) \\ \cdot f(w_t^2|\overline{w^2}_t) \cdot f(\overline{w^2}_t|\boldsymbol{y}_{1:t-1}). \end{aligned} \tag{C.7}$$

Using Equation C.6 and marginalizing $Y_{t|t-1}$, $\boldsymbol{X}_{t|t-1}$, and $\overline{W^2_{t|t-1}}$ from the joint PDF defined in Equation C.7, the prior predictive PDF of $W^2_{t|t-1}$ is

$$\begin{aligned} f(w_t^2|\boldsymbol{y}_{1:t-1}) &= \int f(w_t^2|\overline{w^2}_t) \cdot f(\overline{w^2}_t|\boldsymbol{y}_{1:t-1}) d\overline{w^2}_t, \\ &= \int \mathcal{N}(w_t^2; \overline{w^2}_t, 2(\overline{w^2}_t)^2) \cdot \mathcal{N}(\overline{w^2}_t; \mu^{\overline{W^2}}_{t-1|t-1}, (\sigma^{\overline{W^2}}_{t-1|t-1})^2) d\overline{w^2}_t. \end{aligned} \tag{C.8}$$

The integration in Equation C.8 can be solved in closed-form (see Section 2.3.1). The equivalent formulation to obtain this is to represent the Gaussian random variables W^2 and $\overline{W^2}$ in terms of the standard Gaussian variable ϵ and ζ shown by

$$W^2 = \overline{W^2} + \sqrt{2}\,\overline{W^2}\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1) \tag{C.9}$$

$$\overline{W^2} = \mu^{\overline{W^2}} + \sigma^{\overline{W^2}}\zeta, \quad \zeta \sim \mathcal{N}(0, 1). \tag{C.10}$$

Using Equations C.9 & (C.10), the mean and variance of the prior predictive PDF of $W^2_{t|t-1}$ are given by

$$\begin{aligned}
\mathbb{E}[W^2_{t|t-1}] &= \mathbb{E}[\overline{W^2_{t|t-1}}] + \sqrt{2}\cancelto{0}{\mathbb{E}[\overline{W^2_{t|t-1}}\,\epsilon]}, \\
&= \mu^{\overline{W^2}}_{t-1|t-1}, \qquad \text{(C.11)} \\
\mathrm{var}(W^2_{t|t-1}) &= \mathrm{var}(\overline{W^2_{t|t-1}}) + 2\,\mathrm{var}(\overline{W^2_{t|t-1}}\,\epsilon), \\
&= (\sigma^{\overline{W^2}}_{t-1|t-1})^2 + 2(\mathrm{var}(\overline{W^2_{t|t-1}}) \cdot \cancelto{1}{\mathrm{var}(\epsilon)} \\
&+ \cancelto{1}{\mathrm{var}(\epsilon)} \cdot \mathbb{E}[\overline{W^2_{t|t-1}}]^2), \\
&= 3(\sigma^{\overline{W^2}}_{t-1|t-1})^2 + 2(\mu^{\overline{W^2}}_{t-1|t-1})^2, \qquad \text{(C.12)}
\end{aligned}$$

where the term $\mathrm{var}(\overline{W^2_{t|t-1}}\,\epsilon)$ in Equation C.12 is obtained using the GMA equations,

$$\mathrm{var}(\overline{W^2_{t|t-1}}\,\epsilon) = \mathrm{var}(\overline{W^2_{t|t-1}}) \cdot \mathrm{var}(\epsilon) + \mathrm{var}(\epsilon) \cdot \mathbb{E}[\overline{W^2_{t|t-1}}]^2.$$

Using Equations C.1 & C.11, the error variance term is given by

$$\sigma^2_W = \mu^{\overline{W^2}}_{t-1|t-1}. \qquad \text{(C.13)}$$

□

C.3 Proof for Lemma 3

Proof. Let us consider the joint PDF of $Y_{t|t-1}$, $W^2_{t|t-1}$, and $\overline{W^2_{t|t-1}}$ as shown by Figure 5.1,

$$\begin{aligned}
f(y_t, w_t^2, \overline{w^2}_t|\boldsymbol{y}_{1:t-1}) &= f(y_t|w_t^2) \cdot f(w_t^2|\overline{w^2}_t) \cdot f(\overline{w^2}_t|\boldsymbol{y}_{1:t-1}). \\
&= f(y_t|w_t^2) \cdot \frac{f(w_t^2|\overline{w^2}_t) \cdot f(\overline{w^2}_t|\boldsymbol{y}_{1:t-1})}{f(w_t^2|\boldsymbol{y}_{1:t-1})} \cdot f(w_t^2|\boldsymbol{y}_{1:t-1}), \\
&= f(y_t|w_t^2) \cdot \frac{f(w_t^2, \overline{w^2}_t|\boldsymbol{y}_{1:t-1})}{f(w_t^2|\boldsymbol{y}_{1:t-1})} \cdot f(w_t^2|\boldsymbol{y}_{1:t-1}), \\
&= f(y_t|w_t^2) \cdot f(\overline{w^2}_t|w_t^2, \boldsymbol{y}_{1:t-1}) \cdot f(w_t^2|\boldsymbol{y}_{1:t-1}). \qquad \text{(C.14)}
\end{aligned}$$

By dividing both sides in Equation C.14 by $f(y_t|\boldsymbol{y}_{1:t-1})$ we obtain

$$\begin{aligned}
\frac{f(y_t, w_t^2, \overline{w^2}_t|\boldsymbol{y}_{1:t-1})}{f(y_t|\boldsymbol{y}_{1:t-1})} &= \frac{f(y_t|w_t^2) \cdot f(w_t^2|\boldsymbol{y}_{1:t-1})}{f(y_t|\boldsymbol{y}_{1:t-1})} \cdot f(\overline{w^2}_t|w_t^2, \boldsymbol{y}_{1:t-1}), \\
f(w_t^2, \overline{w^2}_t|\boldsymbol{y}_{1:t}) &= f(w_t^2|\boldsymbol{y}_{1:t}) \cdot f(\overline{w^2}_t|w_t^2, \boldsymbol{y}_{1:t-1}), \qquad \text{(C.15)}
\end{aligned}$$

By marginalizing out W^2 from the joint PDF defined in Equation C.15, the posterior PDF of $\overline{W^2}_{t|t}$ is obtained such that

$$f(\overline{w^2}_t|\boldsymbol{y}_{1:t}) = \int f(w_t^2|\boldsymbol{y}_{1:t}) \cdot f(\overline{w^2}_t|w_t^2, \boldsymbol{y}_{1:t-1}) dw_t^2. \tag{C.16}$$

□

C.4 Proof for Lemma 4

Proof. Using the GMA equations in Section 3.2, the posterior PDF of W^2 can be shown by

$$f(w_t^2|\boldsymbol{y}_{1:t}) \sim \mathcal{N}(w_t^2; \mu_{t|t}^{W^2}, (\sigma_{t|t}^{W^2})^2), \tag{C.17}$$

where the mean and variance of $W_{t|t}^2$ are

$$\begin{aligned} \mu_{t|t}^{W^2} &= (\mu_{t|t}^{W})^2 + (\sigma_{t|t}^{W})^2, \\ (\sigma_{t|t}^{W^2})^2 &= 2(\sigma_{t|t}^{W})^4 + 4(\sigma_{t|t}^{W})^2(\mu_{t|t}^{W})^2. \end{aligned}$$

□

C.5 Proof for Proposition 2

Proof. Given that the prior predictive PDF of both W^2 and $\overline{W^2}$ are Gaussian, the joint multivariate Gaussian PDF $f(\overline{w^2}_t, w_t^2|\boldsymbol{y}_{1:t-1})$ is shown by

$$f(\overline{w^2}_t, w_t^2|\boldsymbol{y}_{1:t-1}) = \mathcal{N}\Big(\begin{pmatrix} \overline{w^2}_t \\ w_t^2 \end{pmatrix}; \boldsymbol{\mu}_{t|t-1}^{\overline{W^2},W^2}, \boldsymbol{\Sigma}_{t|t-1}^{\overline{W^2},W^2} \Big), \tag{C.18}$$

having a mean vector $\boldsymbol{\mu}_{t|t-1}^{\overline{W^2},W^2}$ and a covariance matrix $\boldsymbol{\Sigma}_{t|t-1}^{\overline{W^2},W^2}$ defined as

$$\begin{aligned} \boldsymbol{\mu}_{t|t-1}^{\overline{W^2},W^2} &= \Big[\mu_{t|t-1}^{\overline{W^2}} \; \mu_{t|t-1}^{W^2} \Big]^\intercal, \\ \boldsymbol{\Sigma}_{t|t-1}^{\overline{W^2},W^2} &= \begin{bmatrix} (\sigma^{\overline{W^2}})_{t|t-1}^2 & \text{cov}(\overline{W^2}, W^2)_{t|t-1} \\ \text{cov}(W^2, \overline{W^2})_{t|t-1} & (\sigma^{W^2})_{t|t-1}^2 \end{bmatrix}, \end{aligned} \tag{C.19}$$

and where using the transition model $w_t^2 = w_{t-1}^2$, the mean and the variance of $W_{t|t-1}^2 = W_{t-1|t-1}^2$ are given by Equations C.11 & C.12. The covariance term $\text{cov}(W^2, \overline{W^2})_{t|t-1}$ between $W_{t|t-1}^2$ and $\overline{W_{t|t-1}^2}$ in Equation C.19 is obtained using Equations C.5 & C.9, and the GMA

equations from Section 3.2 so that

$$\begin{aligned}
\text{cov}(W^2_{t|t-1}, \overline{W^2}_{t|t-1}) &= \text{cov}(W^2, \overline{W^2})_{t|t-1}, \\
&= \text{cov}(\overline{W^2} + \sqrt{2}\,\overline{W^2}\epsilon, \overline{W^2})_{t|t-1}, \\
&= \text{var}(\overline{W^2})_{t|t-1} + \sqrt{2}\text{cov}(\overline{W^2}\epsilon, \overline{W^2})_{t|t-1}, \\
&= \text{var}(\overline{W^2})_{t|t-1} + \sqrt{2}(\text{cov}(\overline{W^2}, \overline{W^2})\cancelto{0}{\mathbb{E}[\epsilon]} \\
&+ \cancelto{0}{\text{cov}(\epsilon, \overline{W^2})}\mathbb{E}[\overline{W^2}]), \\
&= (\sigma^{\overline{W^2}}_{t-1|t-1})^2.
\end{aligned}$$

Given that the joint Gaussian PDF is defined as shown by Equation C.18, the Gaussian conditional properties are used to obtain the conditional PDF $f(\overline{w^2}_t|w^2_t, \boldsymbol{y}_{1:t-1})$ which is part of the integrand shown in Equation C.16,

$$f(\overline{w^2}_t|w^2_t, \boldsymbol{y}_{1:t-1}) = \mathcal{N}(\overline{w^2}_t; \mu^{\overline{W^2}|W^2}_{t|t-1}, (\sigma^{\overline{W^2}|W^2}_{t|t-1})^2), \tag{C.20}$$

for which the conditional mean and variance are

$$\mu^{\overline{W^2}|W^2}_{t|t-1} = \mu^{\overline{W^2}}_{t|t-1} + K_t(w^2_t - \mu^{W^2}_{t|t-1}), \tag{C.21}$$

$$(\sigma^{\overline{W^2}|W^2}_{t|t-1})^2 = (\sigma^{\overline{W^2}}_{t|t-1})^2 - K^2_t(\sigma^{W^2}_{t|t-1})^2, \tag{C.22}$$

$$\begin{aligned}
K_t &= \frac{\text{cov}(W^2_{t|t-1}, \overline{W^2}_{t|t-1})}{(\sigma^{W^2}_{t|t-1})^2}, \\
&= \frac{(\sigma^{\overline{W^2}}_{t-1|t-1})^2}{(\sigma^{W^2}_{t|t-1})^2}.
\end{aligned} \tag{C.23}$$

Using Equations C.20 & C.17, Equation C.16 is rewritten as

$$f(\overline{w^2}_t|\boldsymbol{y}_{1:t}) = \int \mathcal{N}(\overline{w^2}_t; \mu^{\overline{W^2}|W^2}_{t|t-1}, (\sigma^{\overline{W^2}|W^2}_{t|t-1})^2) \cdot \mathcal{N}(w^2_t; \mu^{W^2}_{t|t}, (\sigma^{W^2}_{t|t})^2) dw^2_t. \tag{C.24}$$

Equation C.24 can be solved in closed-form having a Gaussian PDF with a random mean, i.e., $\mu^{\overline{W^2}|W^2}_{t|t-1}$, and a constant variance, i.e., $(\sigma^{\overline{W^2}|W^2}_{t|t-1})^2$, shown by Equations C.21 & C.22. Hence, the PDF $f(\overline{w^2}_t|\boldsymbol{y}_{1:t})$ is also Gaussian such that

$$f(\overline{w^2}_t|\boldsymbol{y}_{1:t}) = \mathcal{N}(\overline{w^2}_t; \mu^{\overline{W^2}}_{t|t}, (\sigma^{\overline{W^2}}_{t|t})^2),$$

for which the posterior mean and the variance are

$$\begin{aligned}
\mu_{t|t}^{\overline{W^2}} &= \mathbb{E}\left[\mu_{t|t-1}^{\overline{W^2}} + K_t(W_{t|t}^2 - \mu_{t|t-1}^{W^2})\right], \\
&= \mu_{t|t-1}^{\overline{W^2}} + K_t(\mu_{t|t}^{W^2} - \mu_{t|t-1}^{W^2}), \\
(\sigma_{t|t}^{\overline{W^2}})^2 &= (\sigma_{t|t-1}^{\overline{W^2}})^2 - K_t^2(\sigma_{t|t-1}^{W^2})^2 + K_t^2 \mathrm{var}(W_{t|t}^2), \\
&= (\sigma_{t|t-1}^{\overline{W^2}})^2 + K_t^2((\sigma_{t|t}^{W^2})^2 - (\sigma_{t|t-1}^{W^2})^2). \qquad \text{(C.25)}
\end{aligned}$$

□

APPENDIX D

D.1 Proof for Lemma 5

As described by Lemma 2 in Chapter 5 for the univariate process error, the expected value $\mathbb{E}[(W^i)^2]$ and the variance terms $\text{var}((W^i)^2), \forall i \in \{1, 2, \cdots, \mathtt{D}\}$ for the prior predictive PDF of $\boldsymbol{W^p}$ are given by

$$\begin{aligned} \mathbb{E}[(W^i)^2] &= (\mu^{\overline{(W^i)^2}}), \\ \text{var}((W^i)^2) &= 3(\sigma^{\overline{(W^i)^2}})^2 + 2(\mu^{\overline{(W^i)^2}})^2. \end{aligned}$$

Using the GMA equations, the mean and variance term of W^iW^j are

$$\begin{aligned} \mathbb{E}[W^iW^j] &= \text{cov}(W^i, W^j) = \overline{w^iw^j}, & \text{(D.1)} \\ \text{var}(W^iW^j) &= \text{var}(W^i)\text{var}(W^j) + \text{cov}(W^i, W^j)^2, \\ &= \mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{w^iw^j})^2, & \text{(D.2)} \end{aligned}$$

where using Equation C.13 from Proof C.2, $\text{var}(W^i) = \mu^{\overline{(W^i)^2}}$. Using Equations D.1 and D.2, the Gaussian random variable $W^iW^j \sim \mathcal{N}(w^iw^j; \overline{w^iw^j}, \mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{w^iw^j})^2)$ can be represented in terms of its standard Gaussian variable ϵ by

$$w^iw^j = \overline{w^iw^j} + \sqrt{\mu^{\overline{(W^i)^2}} \cdot \mu^{\overline{(W^j)^2}} + (\overline{w^iw^j})^2} \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, 1).$$

The moments of the prior predictive PDF of W^iW^j are given by

$$\begin{aligned} \mathbb{E}[W^iW^j] &= \mathbb{E}[\overline{W^iW^j}] + \mathbb{E}\left[\sqrt{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{W^iW^j})^2} \cdot \epsilon\right], \quad (\to 0) \\ &= \mu^{\overline{W^iW^j}}, & \text{(D.3)} \\ \text{var}(W^iW^j) &= \text{var}(\overline{W^iW^j}) + \text{var}(\sqrt{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{W^iW^j})^2} \cdot \epsilon), \\ &= \text{var}(\overline{W^iW^j}) + \text{var}(\sqrt{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{W^iW^j})^2}), \\ &\quad + \mathbb{E}\left[\sqrt{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{W^iW^j})^2}\right]^2, & \text{(D.4)} \end{aligned}$$

where using GMA equations the term $\text{var}(\sqrt{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{W^iW^j})^2} \cdot \epsilon)$ is obtained by

$$\text{var}(\sqrt{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{W^iW^j})^2}\cdot\epsilon) = \text{var}(\sqrt{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{W^iW^j})^2})+\mathbb{E}\left[\sqrt{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{W^iW^j})^2}\right]^2.$$

In order to simplify the notation in Equation D.4, let us consider $u = \overline{w^iw^j}$ and $t(u) = \sqrt{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\overline{w^iw^j})^2}$, so that using 1^{st} order Taylor series expansion we get

$$\begin{aligned}
\mathbb{E}[t(u)]^2 &= t(\mathbb{E}[U])^2 = \mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\mu^{\overline{W^iW^j}})^2, &\text{(D.5)}\\
\text{var}(t(u)) &= (t'(\mathbb{E}[U]))^2 \cdot \text{var}(U),\\
&= \frac{(\mu^{\overline{W^iW^j}})^2}{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\mu^{\overline{W^iW^j}})^2} \cdot (\sigma^{\overline{W^iW^j}})^2. &\text{(D.6)}
\end{aligned}$$

Hence, combining Equations D.4, D.5, and D.6 we get

$$\begin{aligned}
\text{var}(W^iW^j) &= (\sigma^{\overline{W^iW^j}})^2 + \frac{(\mu^{\overline{W^iW^j}})^2}{\mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\mu^{\overline{W^iW^j}})^2} \cdot (\sigma^{\overline{W^iW^j}})^2\\
&\quad + \mu^{\overline{(W^i)^2}}\mu^{\overline{(W^j)^2}} + (\mu^{\overline{W^iW^j}})^2.
\end{aligned}$$

Using the GMA equations, and Equations 6.2 and D.3, the covariance between the product terms W^iW^j and W^lW^m, $i, j, l, m \; \forall \in \{1, 2, \cdots \mathtt{D}\}$, is given by

$$\begin{aligned}
\text{cov}(W^iW^j, W^lW^m) &= \text{cov}(W^i, W^l)\text{cov}(W^j, W^m) + \text{cov}(W^i, W^m)\text{cov}(W^j, W^l),\\
&= \mathbb{E}[W^iW^l]\mathbb{E}[W^jW^m] + \mathbb{E}[W^iW^m]\mathbb{E}[W^jW^l],\\
&= \mu^{\overline{W^iW^l}}\mu^{\overline{W^jW^m}} + \mu^{\overline{W^iW^m}}\mu^{\overline{W^jW^l}}.
\end{aligned}$$

D.2 Proof for Lemma 6

Proof. The covariance matrix $\boldsymbol{\Sigma^W}$ in Equation 6.3 can be reformulated in terms of the random variables in $\overline{\boldsymbol{W^p}}$ given by

$$\boldsymbol{\Sigma^W} = \begin{bmatrix}
\overline{(W^1)^2} & \overline{W^1W^2} & \cdots & \overline{W^1W^{\mathtt{D}}}\\
\vdots & \overline{(W^2)^2} & \cdots & \overline{W^2W^{\mathtt{D}}}\\
\vdots & \cdots & \ddots & \vdots\\
\text{sym.} & \cdots & \cdots & \overline{(W^{\mathtt{D}})^2}
\end{bmatrix}_{t|t-1}, \qquad \text{(D.7)}$$

where using Equation D.1, $\mathbb{E}[W^i W^j] = \overline{W^i W^j}, \forall i, j \in \{1,\ 2,\ \cdots, \mathtt{D}\}$. Let us consider $\boldsymbol{L^W}$ is an upper triangular random matrix such that

$$\boldsymbol{L^W} = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1\mathtt{D}} \\ 0 & L_{22} & \cdots & L_{2\mathtt{D}} \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & L_{\mathtt{DD}} \end{bmatrix}, \tag{D.8}$$

where each of the term is assumed to be a Gaussian random variable given by $L_{ij} \sim \mathcal{N}(\mu_{L_{ij}}, \sigma^2_{L_{ij}})$. The elements of $\boldsymbol{L^W}$ can be arranged in a random vector,

$$\overrightarrow{\boldsymbol{L^W}} = [L_{11}\ L_{22}\ L_{\mathtt{DD}}\ L_{12}\ \cdots L_{ij}\ \cdots L_{\mathtt{D\text{-}1\,D}}]^\intercal,$$

such that $\overrightarrow{\boldsymbol{L^W}}$ is a Gaussian random vector given by

$$\overrightarrow{\boldsymbol{L^W}} \sim \mathcal{N}(\boldsymbol{\mu}^{\overrightarrow{\boldsymbol{L^W}}}, \boldsymbol{\Sigma}^{\overrightarrow{\boldsymbol{L^W}}}), \tag{D.9}$$

where $\boldsymbol{\mu}^{\overrightarrow{\boldsymbol{L^W}}}$ and $\boldsymbol{\Sigma}^{\overrightarrow{\boldsymbol{L^W}}}$ are the mean vector and the covariance matrix of $\overrightarrow{\boldsymbol{L^W}}$. Let us reproduce $\boldsymbol{\Sigma^W}$ using Equation D.8 such that

$$\boldsymbol{\Sigma^W} = (\boldsymbol{L^W})^\intercal \boldsymbol{L^W},$$

where each element $\overline{W^i W^j}$ of $\boldsymbol{\Sigma^W}$ defined in Equation D.7 is obtained using matrix multiplication so that

$$\overline{W^i W^j} = \sum_{k=1}^{\mathtt{D}} L_{jk} L_{ki}, \forall i, j \in \{1, \cdots, \mathtt{D}\},$$

where using Equation D.9 and the GMA equations we can determine the expected value, the variance, and the covariance terms of any element $\overline{W^i W^j}$ as follows,

$$\mathbb{E}[\overline{W^i W^j}] = \mathbb{E}\left[\sum_{k=1}^{\mathtt{D}} L_{jk} L_{ki}\right],\ \ \mathrm{var}(\overline{W^i W^j}) = \mathrm{var}(\sum_{k=1}^{\mathtt{D}} L_{jk} L_{ki}). \tag{D.10}$$

Using Equation D.10, the elements of the prior predictive PDF of $\boldsymbol{W}$ defined in Proposition 3 can be computed as

$$\mu^{\overline{W^i W^j}} = \mathbb{E}\left[\sum_{k=1}^{\mathtt{D}} L_{jk} L_{ki}\right].$$

Similarly, the covariance between the random matrices, $\boldsymbol{\Sigma}^{\boldsymbol{W}}$ and $\boldsymbol{L}^{\boldsymbol{W}}$, is equivalent to finding the covariance between the random vectors $\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}$ and $\overline{\boldsymbol{W}^{\boldsymbol{p}}}$ given by $\boldsymbol{\Sigma}_{t|t-1}^{\overrightarrow{\boldsymbol{L}^{\boldsymbol{W}}}\overline{\boldsymbol{W}^{\boldsymbol{p}}}}$, where any covariance term is obtained by

$$\text{cov}(L_{ij}, \overline{W^i W^j}) = \text{cov}(L_{ij}, \sum_{k=1}^{\mathtt{D}} L_{jk} L_{ki}).$$

□

D.3 Proof for Lemma 7

Proof. The prior knowledge of $\overline{\boldsymbol{W}^{\boldsymbol{p}}}$ is updated by using the same procedure as shown in Section 5.2.2 which employs the prior $\boldsymbol{W}_{t|t-1}^{\boldsymbol{p}}$ and the posterior $\boldsymbol{W}_{t|t}^{\boldsymbol{p}}$ knowledge of $\boldsymbol{W}^{\boldsymbol{p}}$ such that

$$f(\overline{\boldsymbol{w}_t^{\boldsymbol{p}}}|\boldsymbol{y}_{1:t}) \;=\; \mathcal{N}(\overline{\boldsymbol{w}_t^{\boldsymbol{p}}}; \boldsymbol{\mu}_{t|t}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}}, \boldsymbol{\Sigma}_{t|t}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}}),$$

where using Equations C.23-C.25, the posterior mean, variance and covariance terms of $\overline{\boldsymbol{W}^{\boldsymbol{p}}}$ are

$$\begin{aligned}
\boldsymbol{\mu}_{t|t}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}} &= \boldsymbol{\mu}_{t|t-1}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}} + \boldsymbol{K}_t(\boldsymbol{\mu}_{t|t}^{\boldsymbol{W}^{\boldsymbol{p}}} - \boldsymbol{\mu}_{t|t-1}^{\boldsymbol{W}^{\boldsymbol{p}}}),\\
\boldsymbol{\Sigma}_{t|t}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}} &= \boldsymbol{\Sigma}_{t|t-1}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}} + \boldsymbol{K}_t(\boldsymbol{\Sigma}_{t|t}^{\boldsymbol{W}^{\boldsymbol{p}}} - \boldsymbol{\Sigma}_{t|t-1}^{\boldsymbol{W}^{\boldsymbol{p}}})\boldsymbol{K}_t^{\intercal},\\
\boldsymbol{K}_t &= \boldsymbol{\Sigma}_{t|t-1}^{\boldsymbol{W}^{\boldsymbol{p}}\overline{\boldsymbol{W}^{\boldsymbol{p}}}}(\boldsymbol{\Sigma}_{t|t-1}^{\boldsymbol{W}^{\boldsymbol{p}}})^{-1},\\
\boldsymbol{\Sigma}_{t|t-1}^{\boldsymbol{W}^{\boldsymbol{p}}\overline{\boldsymbol{W}^{\boldsymbol{p}}}} &= \boldsymbol{\Sigma}_{t|t-1}^{\overline{\boldsymbol{W}^{\boldsymbol{p}}}}.
\end{aligned}$$

□

Algorithm 4 One-time step of the AGVI method for multivariate process errors

Input: $\boldsymbol{\mu}_{t-1|t-1}$, $\boldsymbol{\Sigma}_{t-1|t-1}$, $\boldsymbol{\mu}^{\overrightarrow{\boldsymbol{L^W}}}_{t-1|t-1}$, $\boldsymbol{\Sigma}^{\overrightarrow{\boldsymbol{L^W}}}_{t-1|t-1}$, $\boldsymbol{y}_t$, $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$, and $\mathbf{R}$

Prior Predictive PDF of $\boldsymbol{W}_{t|t-1} \sim \mathcal{N}(\boldsymbol{w}_t; \mathbf{0}_{t|t-1}, \boldsymbol{\Sigma}^{\boldsymbol{W}}_{t|t-1})$:

1: Any ij^{th} element of $\boldsymbol{\Sigma}^{\boldsymbol{W}}_{t|t-1}$ is obtained using $\mu^{\overline{W^i W^j}} = \mathbb{E}\left[\sum_{k=1}^{\mathtt{D}} L_{jk} L_{ki}\right]$

Prediction Step:

2: $$\boldsymbol{\mu}^{\boldsymbol{H}}_{t|t-1} = \begin{bmatrix} \mathbf{A}\boldsymbol{\mu}_{t-1|t-1} \\ \mathbf{0} \end{bmatrix}_{t|t-1}, \quad \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t-1} = \begin{bmatrix} \mathbf{A}\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{A}^\intercal + \mathbf{Q} & \boldsymbol{\Sigma}^{\boldsymbol{XW}} \\ (\boldsymbol{\Sigma}^{\boldsymbol{XW}})^\intercal & \boldsymbol{\Sigma}^{\boldsymbol{W}} \end{bmatrix}_{t|t-1},$$
$$\boldsymbol{\mu}_{\boldsymbol{Y}} = \mathbf{C}\boldsymbol{\mu}_{t|t-1}, \quad \boldsymbol{\Sigma}_{\boldsymbol{Y}} = \mathbf{C}\boldsymbol{\Sigma}_{t|t-1}\mathbf{C}^\intercal + \mathbf{R}, \quad \boldsymbol{\Sigma}_{\boldsymbol{HY}} = \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t-1}\mathbf{F}_t^\intercal, \text{ where } \mathbf{F} = [\mathbf{C} \;\; \mathbf{0}]$$

1^{st} **Update Step:**

3: $$\boldsymbol{\mu}^{\boldsymbol{H}}_{t|t} = \boldsymbol{\mu}^{\boldsymbol{H}}_{t|t-1} + \boldsymbol{\Sigma}_{\boldsymbol{HY}}\boldsymbol{\Sigma}_{\boldsymbol{Y}}^{-1}(\boldsymbol{y}_t - \boldsymbol{\mu}_{\boldsymbol{Y}}), \quad \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t} = \boldsymbol{\Sigma}^{\boldsymbol{H}}_{t|t-1} - \boldsymbol{\Sigma}_{\boldsymbol{HY}}\boldsymbol{\Sigma}_{\boldsymbol{Y}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{HY}}^\intercal$$

4: Obtain the posterior PDF, $f(\boldsymbol{w}^{\boldsymbol{p}}_t|\boldsymbol{y}_{1:t}) = \mathcal{N}(\boldsymbol{w}^{\boldsymbol{p}}_t; \boldsymbol{\mu}^{\boldsymbol{W^p}}_{t|t}, \boldsymbol{\Sigma}^{\boldsymbol{W^p}}_{t|t})$

2^{nd} **Update Step:**

5: $$\boldsymbol{\mu}^{\overline{\boldsymbol{W^p}}}_{t|t} = \boldsymbol{\mu}^{\overline{\boldsymbol{W^p}}}_{t|t-1} + \boldsymbol{K}_t(\boldsymbol{\mu}^{\boldsymbol{W^p}}_{t|t} - \boldsymbol{\mu}^{\boldsymbol{W^p}}_{t|t-1}), \quad \boldsymbol{\Sigma}^{\overline{\boldsymbol{W^p}}}_{t|t} = \boldsymbol{\Sigma}^{\overline{\boldsymbol{W^p}}}_{t|t-1} + \boldsymbol{K}_t(\boldsymbol{\Sigma}^{\boldsymbol{W^p}}_{t|t} - \boldsymbol{\Sigma}^{\boldsymbol{W^p}}_{t|t-1})\boldsymbol{K}_t^\intercal,$$
$$\boldsymbol{K}_t = \boldsymbol{\Sigma}^{\boldsymbol{W^p}\overline{\boldsymbol{W^p}}}_{t|t-1}(\boldsymbol{\Sigma}^{\boldsymbol{W^p}}_{t|t-1})^{-1}, \quad \boldsymbol{\Sigma}^{\boldsymbol{W^p}\overline{\boldsymbol{W^p}}}_{t|t-1} = \boldsymbol{\Sigma}^{\overline{\boldsymbol{W^p}}}_{t|t-1}$$

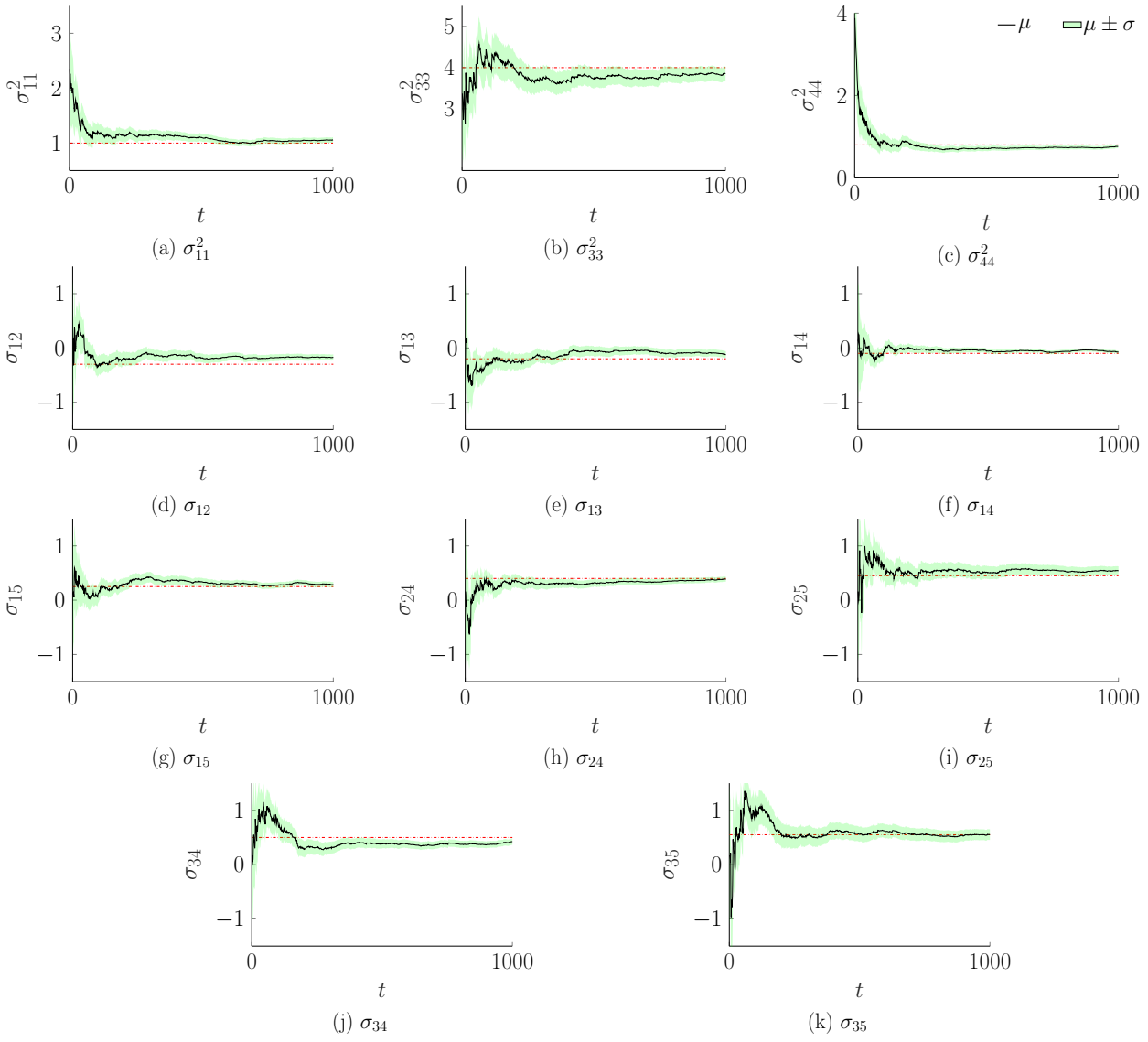
Posterior moments of $\overrightarrow{\boldsymbol{L^W}}$:

6: $$\boldsymbol{\mu}^{\overrightarrow{\boldsymbol{L^W}}}_{t|t} = \boldsymbol{\mu}^{\overrightarrow{\boldsymbol{L^W}}}_{t|t-1} + \boldsymbol{K}^{\boldsymbol{L}}_t(\boldsymbol{\mu}^{\overline{\boldsymbol{W^p}}}_{t|t} - \boldsymbol{\mu}^{\overline{\boldsymbol{W^p}}}_{t|t-1}), \quad \boldsymbol{\Sigma}^{\overrightarrow{\boldsymbol{L^W}}}_{t|t} = \boldsymbol{\Sigma}^{\overrightarrow{\boldsymbol{L^W}}}_{t|t-1} + \boldsymbol{K}^{\boldsymbol{L}}_t(\boldsymbol{\Sigma}^{\overline{\boldsymbol{W^p}}}_{t|t} - \boldsymbol{\Sigma}^{\overline{\boldsymbol{W^p}}}_{t|t-1})(\boldsymbol{K}^{\boldsymbol{L}}_t)^\intercal,$$
$$\boldsymbol{K}^{\boldsymbol{L}}_t = \boldsymbol{\Sigma}^{\overrightarrow{\boldsymbol{L^W}}\overline{\boldsymbol{W^p}}}_{t|t-1}(\boldsymbol{\Sigma}^{\overline{\boldsymbol{W^p}}}_{t|t-1})^{-1}$$

7: **return** $\boldsymbol{\mu}_{t|t}$, $\boldsymbol{\Sigma}_{t|t}$, $\boldsymbol{\mu}^{\overrightarrow{\boldsymbol{L^W}}}_{t|t}$, and $\boldsymbol{\Sigma}^{\overrightarrow{\boldsymbol{L^W}}}_{t|t}$

Table D.1 Comparison of the average RMSE values and the computational time for each method. Each of the methods are picked from different AKF categories where AGVI and SWVBAKF are Bayesian methods whereas ALMF is a covariance-matching method (CMM) and ICM is a correlation method. The variance terms and the covariance terms are represented by σ_{ii}^2 and σ_{ij}^2, $\forall i, j \in 1, \cdots, \mathtt{D}$, respectively.

Variance terms	RMSE			
	AGVI	SWVBAKF	ALMF	ICM
σ_{11}^2	0.0534	0.1215	**0.0418**	0.0814
σ_{44}^2	**0.0281**	0.1671	0.0470	0.0344
σ_{13}	0.0536	0.1772	**0.0532**	0.1470
σ_{14}	**0.0171**	0.1336	0.0226	0.0288
σ_{15}	**0.0229**	0.1002	0.0266	0.0569
σ_{23}	**0.1368**	0.3667	0.1692	0.3366
σ_{25}	**0.0851**	0.2294	0.0988	0.1562
σ_{34}	0.0802	0.3013	0.1108	**0.0683**
σ_{45}	0.0495	0.1128	0.0620	**0.0407**



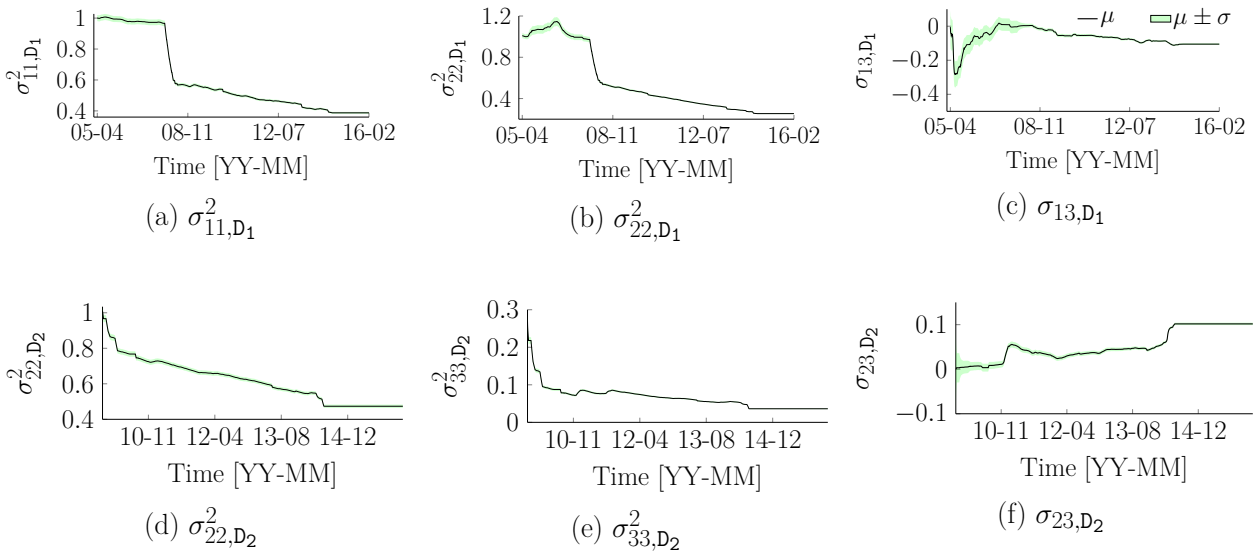
(a) σ^2_{11} (b) σ^2_{33} (c) σ^2_{44}

(d) σ_{12} (e) σ_{13} (f) σ_{14}

(g) σ_{15} (h) σ_{24} (i) σ_{25}

(j) σ_{34} (k) σ_{35}

Figure D.1 Online estimation of the error variance term and the covariance terms from the full **Q** matrix compared to their true values marked by the dashed red line. The estimated values are shown by the black solid line and their $\pm 1\sigma$ uncertainty bound is shown using the green shaded region.



(a) $\sigma^2_{11,\mathtt{D}_1}$ (b) $\sigma^2_{22,\mathtt{D}_1}$ (c) $\sigma_{13,\mathtt{D}_1}$

(d) $\sigma^2_{22,\mathtt{D}_2}$ (e) $\sigma^2_{33,\mathtt{D}_2}$ (f) $\sigma_{23,\mathtt{D}_2}$

Figure D.2 Online estimation of the error variance and covariance terms in the full $\mathbf{Q}$ matrix for both datasets $\boldsymbol{y}_{\mathtt{D}_1}$ and $\boldsymbol{y}_{\mathtt{D}_2}$; where (a) $\sigma^2_{11,\mathtt{D}_1}$, (b) $\sigma^2_{22,\mathtt{D}_1}$, and (c) $\sigma_{13,\mathtt{D}_1}$, (d) $\sigma^2_{22,\mathtt{D}_2}$, (e) $\sigma^2_{33,\mathtt{D}_2}$, and (f) $\sigma_{23,\mathtt{D}_2}$. The estimated values are shown by the black solid line and their $\pm 1\sigma$ uncertainty bound is shown using the green shaded region.

APPENDIX E

E.1 Hyperparameters for the Approximate Inference Methods

The approximate inference methods used for comparison are: PBP [10], MC-dropout [8,114], deterministic NN [15], ensemble of neural networks [9], DVI [13], PBP-MV [11], VMG [12], and TAGI [7]. This section provides the set of hyperparameters associated with each of these methods.

The code for PBP is provided in [118]. PBP tunes its hyperparameter, namely the precision (λ), automatically for which a gamma hyper-prior is considered with the scale and inverse scale parameters, $\alpha_0^\lambda = \beta_0^\lambda = 6$. The batch size used is $\mathtt{B} = 1$.

The code for MC-dropout is provided in [119]. The optimal values for the hyperparameters namely, the dropout rate d and the precision parameter τ are identified for each data split by performing grid-search over a range of (d, τ) pairs. The batch size used is $\mathtt{B} = 128$ and the Adam optimizer used is with the default learning rate built-in Keras [120]. The number of forward iterations (T) used for obtaining predictive uncertainty are 10^4.

For the Ensembles method, an ensemble of 5 neural networks is implemented which is trained with random initialization of weights and bias having a two-headed output layer that provides the mean and error variance. A softplus activation function is used as suggested in [9]. The batch size used is $\mathtt{B} = 128$, the learning rate is 0.01, and the epsilon parameter for adversarial training is set to 1%. A deterministic neural network is also implemented with only one model having the same set of hyperparameters as the Ensembles.

For DVI, the set of hyperparameters for the toy problem are provided in the code by [121], while for the regression benchmark they are provided by Wu et al. [13]. The results for PBP-MV and VMG are directly used as presented in the article by Sun et. al [11].

The implementation code for TAGI is provided in the Github repository by Goulet et al. [122]. The original work identifies the optimal error variance σ_V^2 using a 5 fold cross-validation. The batch size used is $\mathtt{B} = 10$, and the prior covariance for bias is initialized using $0.01 \cdot \boldsymbol{I}$, and for the weights using the Xavier's approach.

E.2 Initialization of the Neural Network's Parameters

Using He's approach [16], the prior covariance matrix for the weights $\boldsymbol{W}^{(j)}$ and bias $\boldsymbol{B}^{(j)}$, in any hidden layer j are given by $\boldsymbol{\Sigma}_{\boldsymbol{W}}^{(j)} = \boldsymbol{\Sigma}_{\boldsymbol{B}}^{(j)} = \frac{1}{n_{j-1}} \cdot \boldsymbol{I}$, where n_{j-1} represents the number of hidden units in the previous layer $j-1$. The He's approach was modified by introducing a scaling factor α [15] for the weights such that the new prior covariance matrix is $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{W}}^{(j)} = \frac{\alpha}{n_{j-1}} \cdot \boldsymbol{I}$ that are initialized according to the data. Also, a different scaling factor β is considered for the weights connected to the 2^{nd} output node providing the error variance. The best pair of values for these two hyperparameters (α and β) are identified using a validation set and a grid-search procedure. The list of hyperparameter values over which the grid-search is carried out are

$$\begin{aligned} \text{patience} &: \{3, 5, 10\}, \\ \alpha &: \{0.1, 0.5, 1\}, \\ \beta &: \{0.1, 0.01, 0.001\}, \end{aligned}$$

where patience is the hyperparameter for the early-stopping procedure. Table E.1 shows the optimal values for the hyperparameters used for each dataset. Figures E.1 and E.2 shows the learning curves for TAGI-V using both the original and the modified He's approach for parameter initialization. The results shows that the modified He's approach provides better predictive accuracy in most datasets.

Table E.1 Optimized set of hyperparameters identified using grid-search procedure. The parameters α and β, and patience are associated with the modified He's approach and early-stopping procedure, respectively. The grid-search is carried out using a validation set obtained from the original training set by a $80-20$ split ratio. The total computational time (in s.) required for the grid-search procedure is also provided.

Datasets	α	β	Patience	Total Time (in s.)
Boston	0.5	0.01	5	591.63
Concrete	0.5	0.01	5	1225.83
Energy	0.5	0.01	5	420.30
Kin8nm	1	0.01	5	3562.74
Naval	0.5	0.01	3	19570.24
Power	0.5	0.001	10	4805.74
Protein	0.5	0.1	10	23299.31
Wine	0.1	0.01	10	353.71
Yacht	0.1	0.1	5	648.60

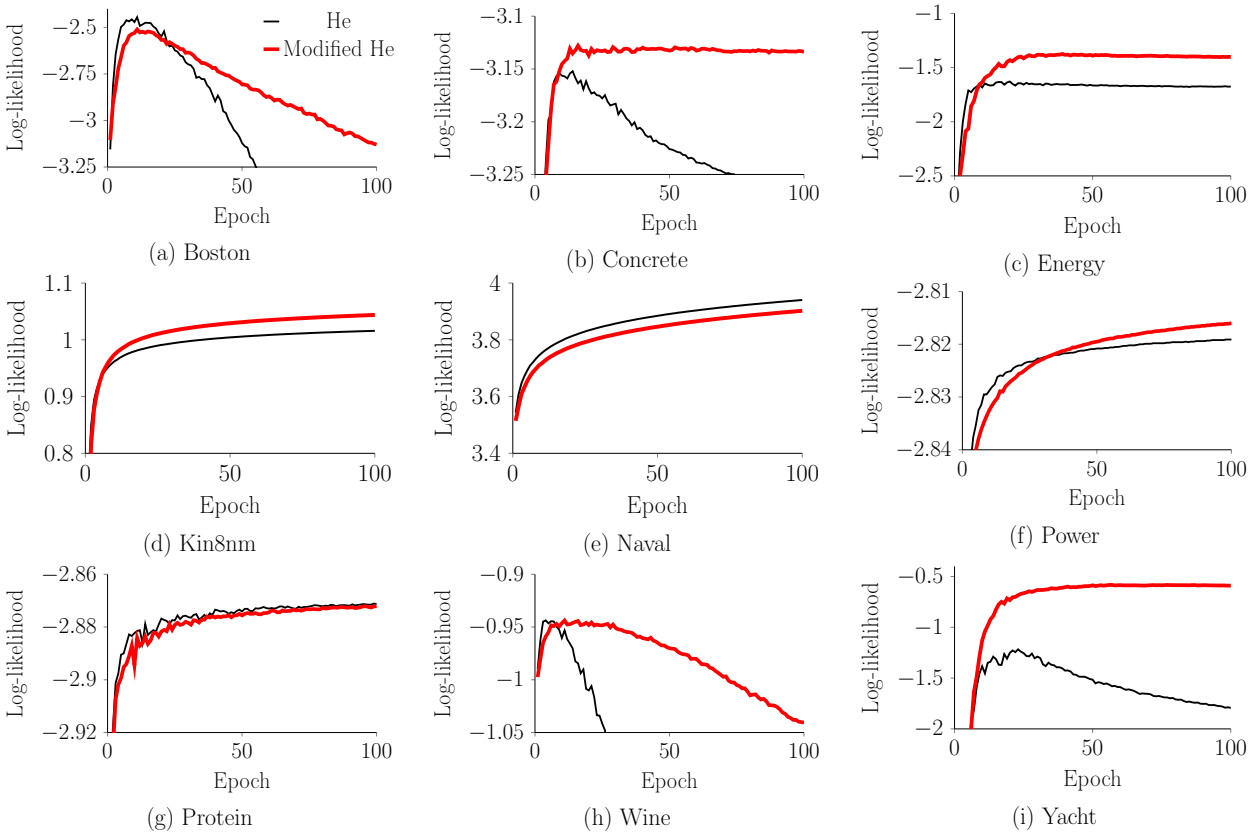


Figure E.1 The learning curves for test log-likelihood showing the comparative performance between the original and modified He's approach for parameter initialization. The black and red solid line represents the performance using the original and modified He's approach, respectively. In the original He's approach [16], the scaling factors are set to $\alpha = \beta = 1$, but for the modified He's approach the scaling factors are tuned for each dataset using a grid-search procedure over possible set of hyperparameter values [15].

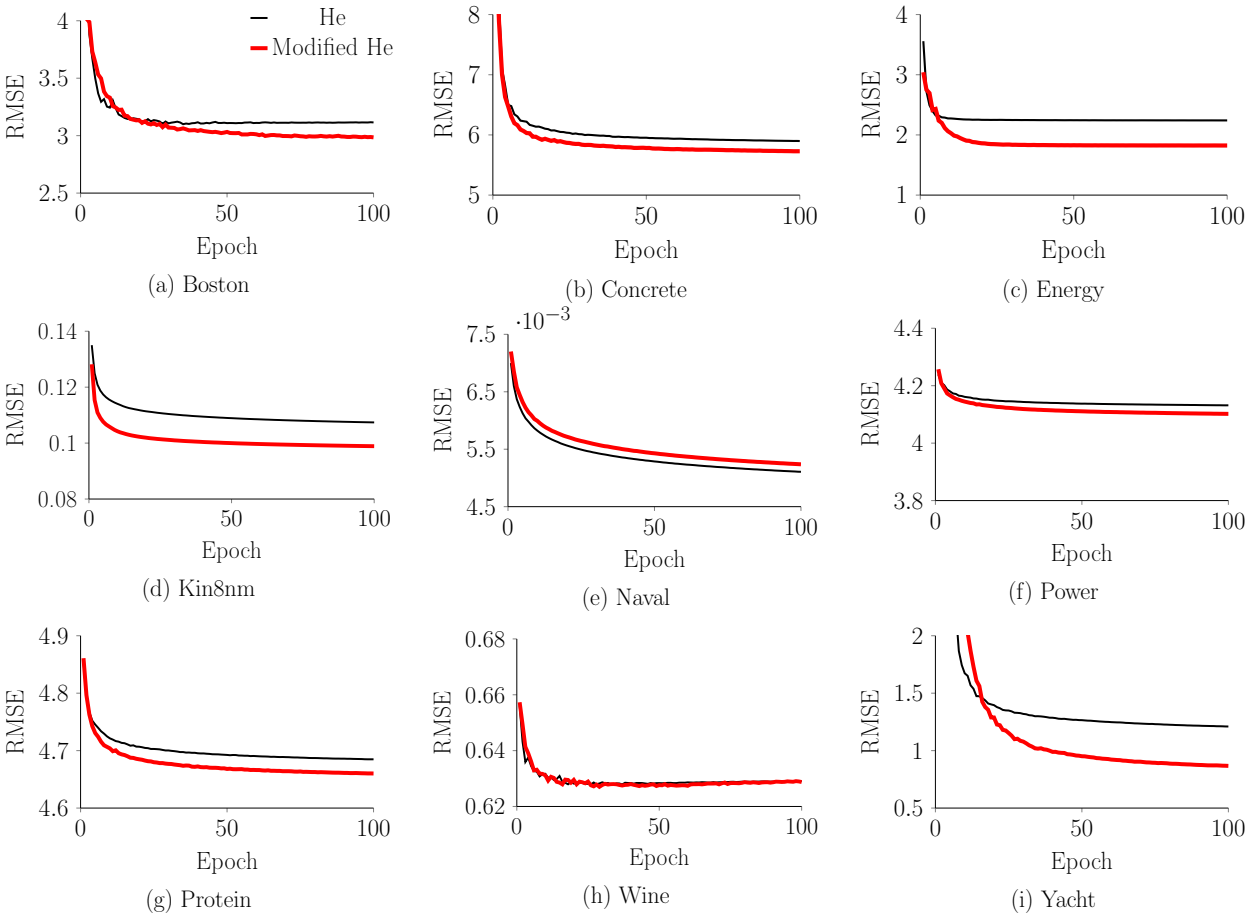


Figure E.2 The learning curves for test RMSE showing the performance using the original and modified He's approach for parameter initialization. The black and red solid line represents the performance using the original and modified He's approach, respectively. In the original He's approach [16], the scaling factors are set to $\alpha = \beta = 1$, but for the modified He's approach, the scaling factors are tuned for each dataset using a grid-search procedure over possible set of hyperparameter values [15].

E.3 Learning Curves for TAGI-V Under Epoch Setting Using Early-Stopping.

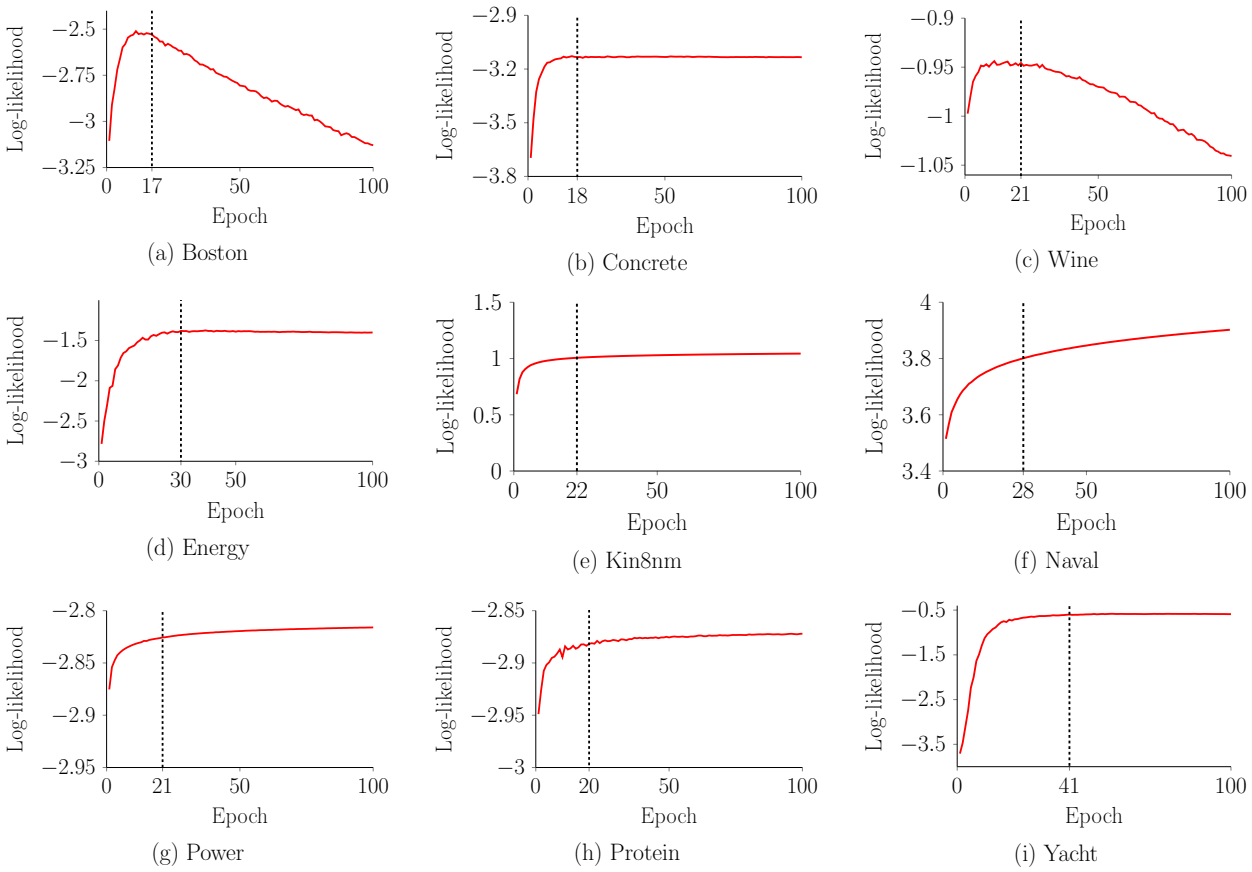


Figure E.3 The learning curves for TAGI-V under epoch setting showing the test log-likelihood for the datasets Energy, Kin8nm, Naval, Power, Protein, and Yacht. The optimal epoch is highlighted by the black dotted line found using early-stopping procedure.

E.4 Learning Curves Showing Test Log-likelihood and Test RMSE Under the Epoch and Time Setting.

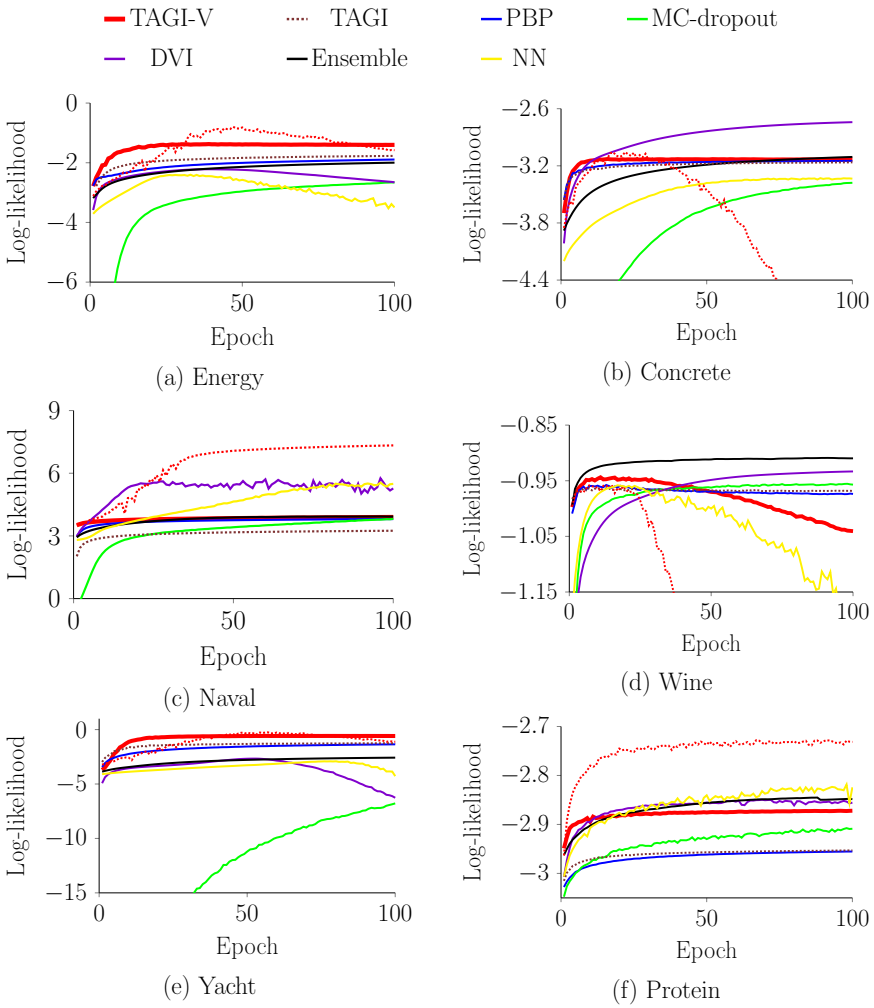


Figure E.4 Learning curves showing the test log-likelihood under the epoch setting. The horizontal axis shows the number of epochs and the vertical axis shows the test loglikelihood. The colored line plots are : TAGI-V (red solid line), PBP (blue solid line) [10], MC-dropout (green solid line) [8], DVI (purple solid line) [13], deterministic NN (yellow solid line) [15], Ensemble (black solid line) [9], TAGI (brown dotted line) [7], and TAGI-V 2L (red dotted line) that represents a TAGI-V network of two layers and 100 hidden nodes.

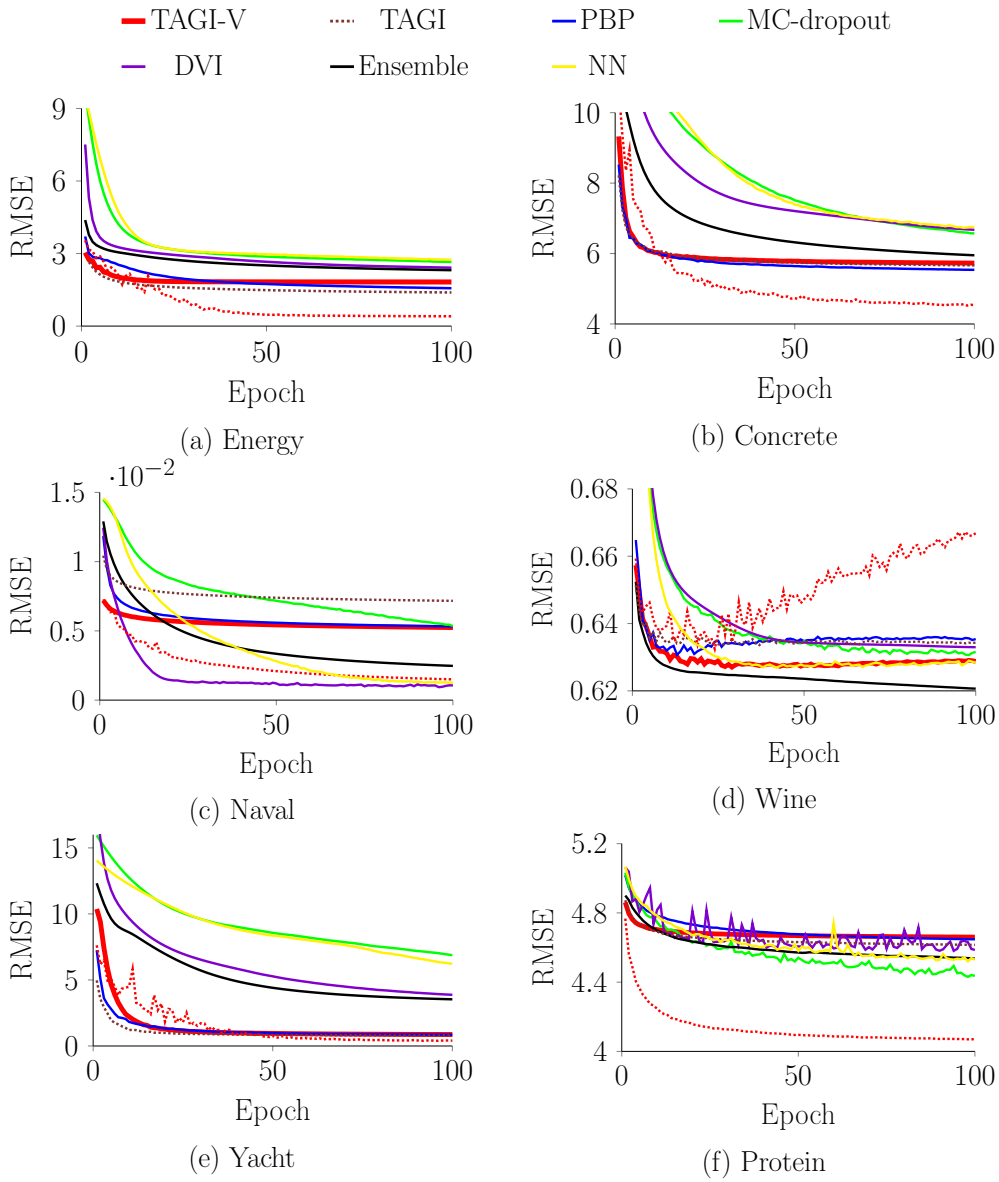


Figure E.5 Learning curves showing the test RMSE under the epoch setting. The horizontal axis shows the number of epochs and the vertical axis shows the test RMSE. The colored line plots are : TAGI-V (red solid line), PBP (blue solid line) [10], MC-dropout (green solid line) [8], DVI (purple solid line) [13], deterministic NN (yellow solid line) [15], Ensemble (black solid line) [9], TAGI (brown dotted line) [7], and TAGI-V 2L (red dotted line) that represents a TAGI-V network of two layers and 100 hidden nodes.

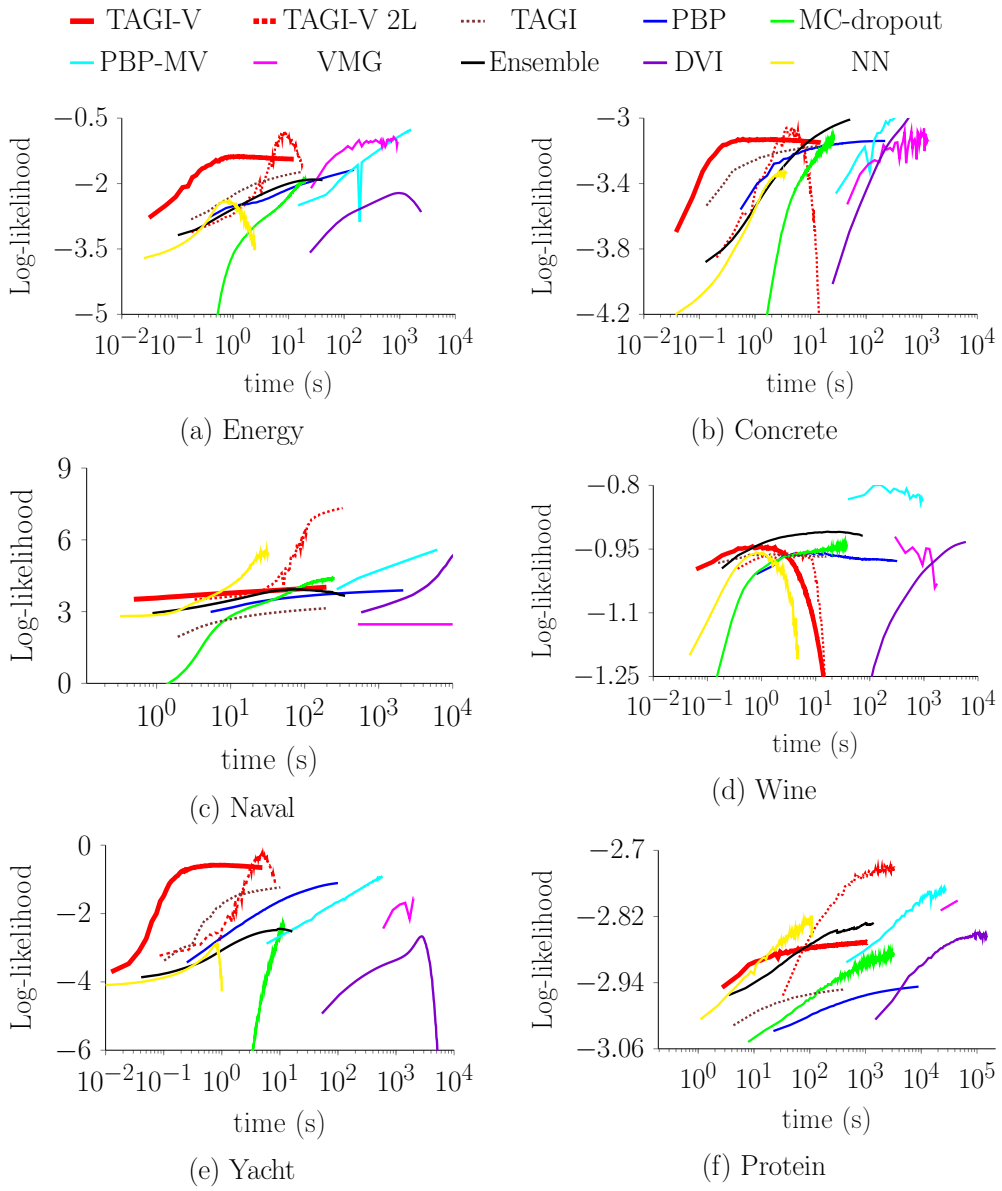


Figure E.6 Learning curves showing the test log-likelihood under the time setting. The horizontal axis represents training time (in s.) in log scale (base 10) and the vertical axis represents the test log-likelihood in linear scale. The colored line plots are : TAGI-V (red solid line), PBP (blue solid line) [10], MC-dropout (green solid line) [8], DVI (purple solid line) [13], deterministic NN (yellow solid line) [15], Ensembles (black solid line) [9], TAGI (brown dotted line) [7], TAGI-V 2L (red dotted line) that represents a TAGI-V network of two layers and 100 hidden nodes, PBP-MV (cyan solid line) [11], and VMG (magenta solid line) [12]. The learning curves for PBP-MV and VMG are obtained directly from the original article by Sun et al. [11].

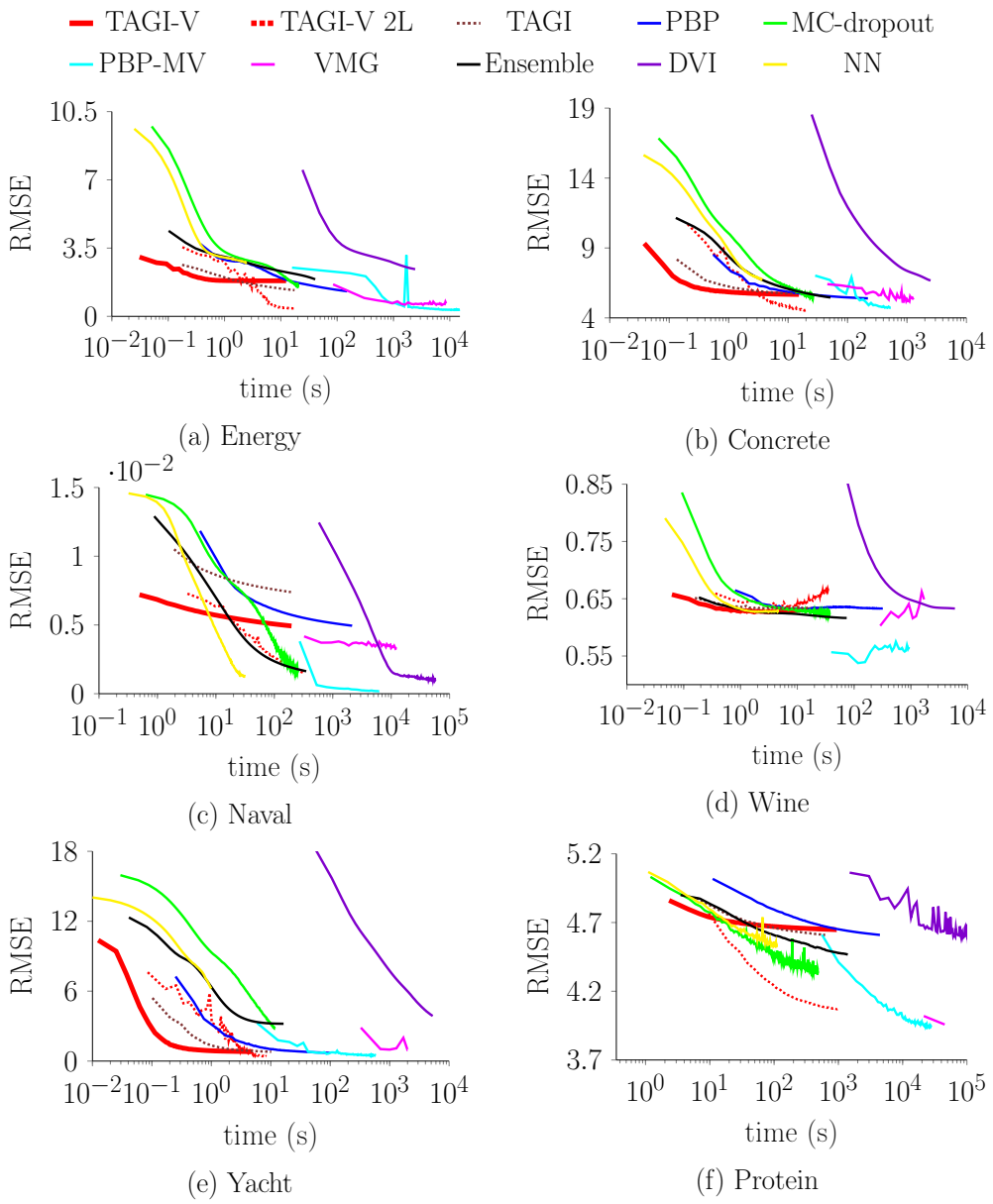


Figure E.7 Learning curves showing the test RMSE under the time setting. The horizontal axis represents training time (in s.) in log scale (base 10) and the vertical axis represents the test RMSE in linear scale. The colored line plots are : TAGI-V (red solid line), PBP (blue solid line) [10], MC-dropout (green solid line) [8], DVI (purple solid line) [13], deterministic NN (yellow solid line) [15], Ensembles (black solid line) [9], TAGI (brown dotted line) [7], TAGI-V 2L (red dotted line) that represents a TAGI-V network of two layers and 100 hidden nodes, PBP-MV (cyan solid line) [11], and VMG (magenta solid line) [12]. The learning curves for PBP-MV and VMG are obtained directly from the original article by Sun et al., 2017 [11].

E.5 Comparison for Computational Time Between the Approximate Inference Methods.

Table E.2 Comparison between the approximate inference methods for average training time (in s.) per epoch (Rank legend: **first**). All the experiments are carried out using 12 core 3GHz CPU. For TAGI-V and TAGI, the codes are in MATLAB, and all others are written in Python.

Datasets	TAGI-V	TAGI	MC-Dropout	Deep Ensembles	PBP	PBP-MV	VMG	DVI	NN
Boston	0.025	0.099	0.041	0.061	0.25	37	18.6	12.86	**0.012**
Concrete	0.038	0.134	0.066	0.129	0.55	28.57	35.71	24.91	**0.037**
Energy	0.031	0.177	0.051	0.102	0.375	14.7	18.38	24.26	**0.025**
Kin8nm	0.309	0.814	0.434	0.642	3.65	158.73	222.22	277.31	**0.194**
Naval	0.493	1.934	0.631	0.881	5.375	271.26	271.26	579.689	**0.320**
Power	0.32	0.783	0.496	0.701	4.20	181.82	981.82	363.816	**0.28**
Protein	2.327	6.23	1.193	3.506	11.075	556	21296.00	1498.41	**1.10**
Wine	0.062	0.157	0.093	0.185	0.80	39.68	166.67	59.53	**0.047**
Yacht	0.012	0.102	0.0295	0.041	0.250	5.56	327.78	53.01	**0.010**

E.6 Comparison for TAGI-V's Predictive Performance.

Table E.3 Comparison between the inference methods for average test RMSE's as mentioned in the original work for TAGI [7], MC-dropout [8], Deep ensembles [9], PBP [10], PBP-MV [11], VMG [12], and DVI [13] (Rank legend: **first**). The $\pm\sigma$ represents one standard deviation computed over 20 splits. The results for TAGI-V are also averaged over 5 random seeds. The results for DVI is left empty as it is not provided in the respective article by Wu et al. [13].

Datasets	TAGI-V	TAGI	MC-dropout	Deep Ensembles	PBP	PBP-MV	VMG	DVI
Boston	2.99 ± 0.86	2.98 ± 0.86	$\mathbf{2.83 \pm 0.17}$	3.28 ± 1.00	3.01 ± 0.18	3.11 ± 0.15	3.18 ± 0.19	–
Concrete	5.94 ± 0.51	5.72 ± 0.52	$\mathbf{4.93 \pm 0.14}$	6.03 ± 0.58	5.66 ± 0.093	5.08 ± 0.14	5.18 ± 0.16	–
Energy	1.94 ± 0.28	1.46 ± 0.22	1.08 ± 0.03	2.09 ± 0.29	1.804 ± 0.048	$\mathbf{0.45 \pm 0.01}$	0.48 ± 0.01	–
Kin8nm	0.102 ± 0.01	$0.10 \pm$ 1e-03	0.09 ± 0.00	0.09 ± 0.00	0.098 ± 0.00	$\mathbf{0.07 \pm 0.00}$	$\mathbf{0.07 \pm 0.00}$	–
Naval	0.005 ± 0.00	$0.01 \pm$ 6e-03	0.00 ± 0.00	0.00 ± 0.00	0.006 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	–
Power	4.11 ± 0.15	4.12 ± 0.16	4.01 ± 0.04	4.11 ± 0.17	4.12 ± 0.034	3.91 ± 0.04	$\mathbf{3.87 \pm 0.05}$	–
Protein	4.70 ± 0.03	4.70 ± 0.02	4.27 ± 0.02	4.71 ± 0.06	4.73 ± 0.013	3.94 ± 0.02	$\mathbf{3.90 \pm 0.02}$	–
Wine	0.63 ± 0.04	0.63 ± 0.04	$\mathbf{0.62 \pm 0.01}$	0.64 ± 0.04	0.64 ± 0.008	0.64 ± 0.01	0.64 ± 0.01	–
Yacht	0.88 ± 0.40	1.02 ± 0.42	$\mathbf{0.67 \pm 0.05}$	1.58 ± 0.48	1.02 ± 0.054	0.81 ± 0.06	0.87 ± 0.08	–

Table E.4 Comparison between the inference methods for average test log-likelihood's as mentioned in the original work for TAGI [7], MC-dropout [8], Deep ensembles [9], PBP [10], PBP-MV [11], VMG [12], and DVI [13] (Rank legend: **first**). The $\pm\sigma$ represents one standard deviation computed over 20 splits. The results for TAGI-V are also averaged over 5 random seeds.

Datasets	TAGI-V	TAGI	MC-Dropout	Deep Ensembles	PBP	PBP-MV	VMG	DVI
Boston	-2.51 ± 0.30	-2.58 ± 0.45	$\mathbf{-2.40 \pm 0.04}$	-2.41 ± 0.25	-2.57 ± 0.09	-2.54 ± 0.08	-2.71 ± 0.12	-2.41 ± 0.02
Concrete	-3.13 ± 0.13	-3.17 ± 0.09	$\mathbf{-2.93 \pm 0.02}$	-3.06 ± 0.18	-3.16 ± 0.02	-3.04 ± 0.03	-3.07 ± 0.04	-3.06 ± 0.01
Energy	1.54 ± 0.40	-1.81 ± 0.14	-1.21 ± 0.01	-1.38 ± 0.22	-2.04 ± 0.02	$\mathbf{-1.01 \pm 0.01}$	-0.91 ± 0.01	$\mathbf{-1.01 \pm 0.06}$
Kin8nm	1.00 ± 0.04	0.88 ± 0.04	1.14 ± 0.01	1.20 ± 0.02	0.90 ± 0.01	$\mathbf{1.28 \pm 0.01}$	1.24 ± 0.00	1.13 ± 0.00
Naval	3.82 ± 0.05	2.10 ± 0.57	4.45 ± 0.00	5.63 ± 0.05	3.73 ± 0.01	4.85 ± 0.06	2.47 ± 0.00	$\mathbf{6.29 \pm 0.04}$
Power	-2.82 ± 0.04	-2.83 ± 0.04	-2.80 ± 0.01	-2.79 ± 0.04	-2.84 ± 0.01	$\mathbf{-2.78 \pm 0.01}$	$\mathbf{-2.78 \pm 0.01}$	-2.80 ± 0.00
Protein	-2.88 ± 0.03	$-2.97 \pm$ 4e-03	-2.87 ± 0.00	-2.83 ± 0.02	-2.97 ± 0.00	$\mathbf{-2.77 \pm 0.01}$	-2.78 ± 0.01	-2.85 ± 0.01
Wine	-0.95 ± 0.09	-0.96 ± 0.06	-0.93 ± 0.01	-0.94 ± 0.12	-0.97 ± 0.01	-0.97 ± 0.01	-0.99 ± 0.02	$\mathbf{-0.90 \pm 0.01}$
Yacht	-0.51 ± 0.30	-1.49 ± 0.45	-1.25 ± 0.01	-1.18 ± 0.21	-1.63 ± 0.02	-1.64 ± 0.02	-1.46 ± 0.02	$\mathbf{-0.47 \pm 0.03}$

E.7 Hyperparameter Tuning for Large UCI Regression Datasets

The hyperparameters that needs to be learnt are the parameters α and β associated with the modified He's approach and patience for the early-stopping procedure. The list of hyperpa-

rameter values over which the grid-search is carried out are as follows:

$$\begin{aligned} \text{patience} &: \{3, 5, 10\}, \\ \alpha &: \{0.1, 0.5\}, \\ \beta &: \{0.1, 0.01, 0.001, 0.0001\}, \end{aligned}$$

where patience is the hyperparameter for the early-stopping procedure. Note that any combination of hyperparameters causing numerical overflow errors are omitted from the grid-search procedure. Table E.1 shows the optimal values for the hyperparameters used for each dataset.

Table E.5 Optimized set of hyperparameters identified using grid-search procedure. The parameters α and β, and patience are associated with the modified He's approach and early-stopping procedure, respectively. The grid-search is carried out using a validation set obtained from the original training set by a $80 - 20$ split ratio.

Datasets	α	β	Patience
Elevators	0.1	0.1	10
KeggD	0.1	10^{-4}	3
KeggU	0.1	0.1	10
Pol	0.5	10^{-3}	3
Skillcraft	0.1	0.1	3