



THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Innovative Data-Driven Modeling and Control for Nonlinear Systems using Volterra Series

JEREMY G. STODDARD
BEng (Hons) (Elec.)

A thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy
in Electrical Engineering

August 2019

*This research was supported by an Australian Government Research Training Program
(RTP) Scholarship*

Declaration

I hereby certify that the work embodied in this thesis is my own work, conducted under normal supervision. The thesis contains no material which has been accepted, or is being examined, for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made. I give consent to the final version of my thesis being made available worldwide when deposited in the University's Digital Repository, subject to the provisions of the Copyright Act 1968.

The content of this thesis is the result of original research, and all associated publications have been coauthored with my principal supervisor, A/Prof. James Welsh. Chapters 4 and 5 are based on collaborative work with Prof. Håkan Hjalmarsson. Chapter 8 is based on collaborative work with Dr. Georgios Birpoutsoukis and Prof. Johan Schoukens. Chapter 9 is based on collaborative work with Dr. Georgios Birpoutsoukis and Dr. John Lataire.

Jeremy G. Stoddard

BEng (Hons) (Elec.)

August, 2019

Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa. Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae

tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Contents

Acknowledgements	i
Contents	iii
List of Abbreviations	ix
List of Symbols	xi
1 Introduction	1
1.1 Background	2
1.2 Motivation	4
1.3 Thesis outline and contributions	5
1.4 Publication list	8
I Preliminaries	11
2 System Identification and Control	13
2.1 Dynamical systems	14
2.2 The identification process	15
2.2.1 Experiment design	16
2.2.2 Choosing a model class	17
2.2.3 Choosing the fit criterion	18
2.2.4 Model estimation	18
2.2.5 Validation	18
2.3 The impulse response model	19
2.4 Classical estimators	20
2.4.1 The least squares approach	21
2.4.2 Maximum likelihood estimation	21
2.4.3 Estimator properties	22
2.5 Regularized least squares and the Bayesian perspective	24
2.5.1 Regularization penalty options	25
2.5.2 The quadratic penalty	26

2.5.3	Bayesian interpretation	27
2.6	Model-based control	29
2.6.1	MPC formulation	30
2.6.2	MPC stability	31
2.7	Summary	32
3	The Volterra Series	33
3.1	Introduction	34
3.2	Discrete-time representations	35
3.2.1	Time domain representation	35
3.2.2	Frequency domain representations	36
3.3	Properties and assumptions	38
3.4	Relationship to common block structures	41
3.4.1	Wiener model	41
3.4.2	Hammerstein model	41
3.4.3	Wiener-Hammerstein model	42
3.5	History of Volterra series identification	42
3.6	Regularized Volterra series estimation	44
3.7	Summary	48
II	Time Domain Identification	53
4	EM-Based Hyperparameter Optimization for Regularized Volterra Series Estimation	55
4.1	Introduction	56
4.2	The expectation-maximization algorithm	57
4.3	Hyperparameter optimization using EM	58
4.3.1	The expectation step	59
4.3.2	The maximization step	61
4.3.3	Algorithm for implementation	62
4.3.4	Computation time scaling	62
4.4	Numerical examples	63
4.4.1	Simulation Settings	63
4.4.2	Results	65
4.5	Conclusion	67
5	Volterra Kernel Identification using Regularized Basis Function Expansions	69
5.1	Introduction	70
5.2	Orthonormal basis functions in identification	71
5.2.1	Laguerre basis functions	72

5.2.2	Kautz basis functions	72
5.3	Basis function expansions of Volterra models	72
5.3.1	Optimal selection of Laguerre poles	74
5.3.2	Optimal selection of Kautz poles	75
5.4	Regularized identification of basis function kernels	76
5.4.1	Theoretical justification	77
5.4.2	Modified EM for pole selection and hyperparameter optimization	79
5.4.3	Computational complexity and parameter reduction	80
5.5	Numerical examples	81
5.5.1	Simulation Settings	81
5.5.2	Results	84
5.6	Conclusion	86
5.A	Long division of Laguerre basis function filters	86
6	Practical Case Studies in Time Domain Identification	89
6.1	Introduction	90
6.2	Cascaded tanks	90
6.2.1	System description	91
6.2.2	Measurement process	91
6.2.3	Problem statement	92
6.2.4	Estimating unknown initial states	93
6.2.5	Identification results	94
6.3	Coupled electric drives	97
6.3.1	System description	98
6.3.2	Measurement process	99
6.3.3	Problem statement	100
6.3.4	Identification results	100
6.4	Conclusion	102
III	Frequency Domain Identification	105
7	Gaussian Process Regression in the Frequency Domain - the Parallel Hammerstein Case	107
7.1	Introduction	108
7.2	Real/complex normal distributions	109
7.3	The NOFRF model	110
7.3.1	The parallel Hammerstein case	112
7.4	Identification using multi-level excitation	113
7.5	Gaussian process regression for NOFRFs	114
7.5.1	Deriving the MAP estimate	115
7.5.2	Hyperparameter tuning	117

7.6	GPR estimation in the presence of transients	117
7.7	GPR estimation using multiple input realizations	118
7.8	Simulation examples	120
7.8.1	Simulation settings	120
7.8.2	Comparison of traditional and GPR methods	121
7.8.3	Transient estimation test	124
7.9	Conclusion	124
8	Gaussian Process Regression for the Estimation of Generalized Frequency Response Functions	127
8.1	Introduction	128
8.2	Problem formulation	128
8.3	Gaussian process regression for GFRFs	130
8.3.1	GFRFs as real/complex Gaussian vectors	130
8.3.2	The Gaussian assumption	131
8.3.3	Prior covariance design	132
8.3.4	Deriving the MAP estimate	134
8.3.5	Hyperparameter tuning	135
8.3.6	Comparison with the indirect time domain approach	136
8.4	Simulation examples	137
8.4.1	Study 1: rank deficient noise-free case	137
8.4.2	Study 2: limited frequency band with measurement noise . .	138
8.5	Conclusions	140
9	Frequency Domain Response of Nonlinear Systems under Arbitrary Excitation	141
9.1	Introduction	142
9.2	Preliminaries and problem formulation	142
9.2.1	Frequency domain analysis	143
9.2.2	Volterra series formulation	144
9.2.3	Problem statement	144
9.2.4	Linear System Response	145
9.3	The second order Volterra kernel case	145
9.4	The general Volterra series case	148
9.5	Numerical examples	149
9.6	Insights for system identification	152
9.6.1	Transients in a parallel Hammerstein system	152
9.6.2	GFRF estimation in the presence of transients	153
9.7	Conclusion	154
9.A	Transients proof for a second order diagonal kernel	154

IV Model Predictive Control	157
10 Computationally Efficient MPC for Arbitrary Order Volterra-Laguerre Models	159
10.1 Introduction	160
10.2 History of Volterra series model-based control	160
10.3 A flexible state space representation	162
10.4 Model predictive control design	164
10.4.1 State space prediction model	165
10.4.2 Minimizing the objective function	165
10.4.3 Handling input constraints	167
10.4.4 Disturbance model for offset-free control	168
10.5 Notes on stability	169
10.5.1 Linear FIR case	169
10.5.2 Numerical example	171
10.5.3 General case	171
10.6 Conclusion	172
11 Application Examples of Volterra-Laguerre MPC in Water Level Control	175
11.1 Introduction	176
11.2 Simulation study on the cascaded tanks	176
11.2.1 Experiment methodology	176
11.2.2 Volterra-Laguerre model estimates	177
11.2.3 MPC results	178
11.3 Physical test on a coupled tanks apparatus	180
11.3.1 Experiment methodology	181
11.3.2 Identification results	182
11.3.3 MPC results	183
11.4 Conclusion	185
12 Conclusions and Future Work	187
12.1 Summary of contributions	188
12.1.1 Part II - Time domain identification	188
12.1.2 Part III - Frequency domain identification	188
12.1.3 Part IV - Model predictive control	189
12.2 Considerations for future work	190
Bibliography	193

CONTENTS

List of Abbreviations

DC	Diagonal/Correlated
DFT	Discrete Fourier Transform
EM	Expectation-Maximization
FIR	Finite Impulse Response
FRF	Frequency Response Function
GFRF	Generalized Frequency Response Function
GOBFs	Generalized Orthonormal Basis Functions
GPR	Gaussian Process Regression
i.i.d.	Independent and Identically Distributed
IDFT	Inverse Discrete Fourier Transform
IIR	Infinite Impulse Response
KBFs	Kautz Basis Functions
LBFs	Laguerre Basis Functions
LPM	Local Polynomial Method
LRM	Local Rational Method
LS	Least Squares
LTI	Linear Time-Invariant
m-DTFT	m-dimensional Discrete-Time Fourier Transform
MAP	Maximum A Posteriori
MC	Monte Carlo
MIMO	Multiple-input-multiple-output
MLE	Maximum Likelihood Estimation
MLM	Marginal Likelihood Maximization
MPC	Model Predictive Control
MSE	Mean Square Error
NARMAX	Nonlinear Auto-Regressive Moving-Average with eXogenous inputs
NMSE	Normalized Mean Square Error
NOFRF	Nonlinear Output Frequency Response Function
NRMSE	Normalized Root Mean Square Error

PEM	Prediction Error Method
PID	Proportional-Integral-Derivative
PRBS	Pseudorandom Binary Sequence
PRMLS	Pseudo-Random Multilevel Sequence
RCG	Real/Complex Gaussian
ReLS	Regularized Least Squares
RMS	Root Mean Square
SISO	Single-input-single-output
SNR	Signal-to-Noise Ratio
TC	Tuned/Correlated
TIFM	Time-Invariant and Fading Memory

List of Symbols

System Identification

\mathcal{S}_0	True system
\mathcal{M}	Set of all possible models in the model class
θ	Model parameters
θ_0	True parameter vector
$V(\theta)$	Fit criterion for identification
$\hat{\theta}$	Estimate of the model parameters
h	Impulse response of a linear system
Y	Vector of output measurements
Φ	Regressor matrix
E	Vector of errors
σ^2	Variance
P	Quadratic regularization penalty or Bayesian prior covariance
η	Hyperparameters for Bayesian regularization
$\hat{\eta}^{(k)}$	EM hyperparameter sequence
Σ_Y	Covariance of output vector Y
m	Nonlinear Volterra series order
M	Maximum nonlinear Volterra series order
h_0	Constant offset/0 th order Volterra kernel
h_m	Volterra kernel of order m
n_m	Memory length of kernel h_m
d_m	Number of unique coefficients in a kernel h_m
$h_m^{\mathcal{V}}$	Vector containing all unique coefficients from h_m
P_m	Prior covariance for kernel parameters $h_m^{\mathcal{V}}$
H_m	GFRF of order m
$H_m^{\mathcal{V}}$	RCG vector containing all unique coefficients from H_m
$G.$	NOFRF or transfer function
α_0	Constant offset/0 th order basis function kernel
α_m	Basis function expanded kernel of order m

\mathcal{B}_m	Number of basis functions in the α_m expansion
-----------------	-------------------------------------------------------

Control

u^*	Optimal MPC input sequence
μ	MPC move horizon
d_o	Output disturbance term
Γ_u	Scaling parameter for MPC input penalty

Signals

y^0	True system output
y_m	Output contribution from m^{th} nonlinear order
$u(t)$	Measured input at sample/time t
$y(t)$	Measured output at sample/time t
$r(t)$	Reference signal
$U(k)$	Input spectrum at DFT bin k
$U_m(k)$	DFT spectra of $u^m(t)$
$Y^0(k)$	True output spectrum at DFT bin k
$Y(k)$	Measured output spectrum at DFT bin k
Y_m	Output spectrum contribution from m^{th} nonlinear order
T_m	Output transient at the m^{th} nonlinear order
\mathbf{k}_u	Set of DFT frequencies at which the input spectrum is excited
\mathbf{k}_y	Set of DFT frequencies at which the true output spectrum is nonzero
\mathbf{k}_e	Set of DFT frequencies for frequency domain estimation
ω_k	k^{th} discrete frequency
N	Number of samples in input/output measurements

Statistics

$\mathcal{L}(i; j)$	Likelihood function for i given parameters j
$p(i j)$	Conditional distribution of i given the observation j
$\mathcal{N}(\mu, \Sigma)$	Normal (Gaussian) distribution with mean μ and covariance Σ
$\mathcal{RCN}(\mu, \Sigma)$	Real/complex normal distribution with augmented mean μ and augmented covariance Σ
$\mathbf{E}\{x\}$	Expected value of random variable x

Fields

\mathbb{N}	Set of natural numbers
\mathbb{Z}	Set of integer numbers
\mathbb{Z}^+	Set of positive integers
\mathbb{R}	Set of real numbers
\mathbb{R}^+	Set of positive real numbers
\mathbb{C}	Set of complex numbers
\mathbb{F}	Arbitrary field
$\mathbb{F}^{n_1 \times \dots \times n_m}$	Arbitrary field with dimension $n_1 \times \dots \times n_m$

Operators

q, q^{-1}	Forward and backwards shift
z	z -transform variable
Var	Variance
Cov	Covariance
vec	Vectorization
log	Natural logarithm
det	Determinant
min	Minimum
max	Maximum
diag	Diagonal matrix from a vector
Tr	Trace
$\ \cdot\ _i$	Euclidean i -norm
$\bar{\cdot}$	Complex conjugate
$(\cdot)^T$	Transpose
$(\cdot)^H$	Hermitian transpose
*	Linear convolution
$\langle \cdot, \cdot \rangle$	Inner product
\circ	Hadamard (element-wise) product
\otimes	Kronecker product
$[\cdot]^{\otimes m}$	Kronecker exponentiation by $m \in \mathbb{N}$
\Re	Real part
\Im	Imaginary part
\tilde{X}	Augmented vector of RCG vector X
$\frac{\partial}{\partial x}$	Partial derivative with respect to x

Functions and Constants

δ	Unit impulse function
I	Identity matrix of appropriate dimension
j	Imaginary unit ($\sqrt{-1}$)
$\binom{a}{b}$	binomial coefficient a choose b
\mathcal{K}_m	Kronecker permutation-sum function for m matrix arguments

CHAPTER

1

Introduction

1.1 Background

The dynamical system is a powerful and ubiquitous concept, capable of describing a vast range of natural and man-made processes. While mechanical, electrical and hydraulic systems are classic examples, more complex phenomena such as biological or environmental systems and abstract concepts like economics can also be described by dynamical systems. In all of these cases, to gain an understanding it is useful to construct a mathematical description of the system, which we will call the system *model*. A model has many potential uses, e.g. it can be used to: provide insights into system behaviour, predict future behaviour, simulate the response to hypothetical scenarios, or design a controller for the system. Regardless of its intended use, a model can be obtained in one of three ways:

1. **Phenomenologically:** The mathematical description is built from the underlying physical laws governing each part of the system. This is sometimes referred to as white-box modeling.
2. **Black-box modeling:** A purely data-driven approach, where input and output data are collected experimentally from the system, and a statistical method is applied to transform the system data into a model estimate.
3. **Grey-box modeling:** A combination of the previous two approaches, where the structure of the model is obtained from a phenomenological analysis, but some of the model parameters are estimated in a data-driven fashion.

Phenomenological modeling has limited utility, particularly when the system is prohibitively complex, or when the physical parameters are impossible to accurately quantify. Data-driven modeling, on the other hand, is a far more versatile approach. The application of data-driven modeling to *dynamical* systems is commonly referred to as ‘system identification’, a term we will also adopt here.

As a field of research, system identification began to emerge in the 1960s. Papers by Ho and Kalman [55] and Åström and Bohlin [3] in 1965 became foundational works in identification, paving the way for two main methods of approach: subspace identification and prediction error methods (PEMs) respectively. In the following decades structured frameworks were developed for identification of linear systems, with PEMs receiving a significant majority of the community’s focus [33]. There are many comprehensive textbooks from this era, including [48], [72] and [117], which provide statistical properties and practical considerations for common linear identification methods. While these texts focus chiefly on estimation using time domain data, identification in the frequency domain has also been studied extensively [96].

While the identification of linear dynamical systems is now considered a mature field, the story for nonlinear systems is still, relatively, in its infancy. The first and

most obvious issue when considering nonlinear systems is the vast array of possible models which can be used. In particular, when there is a lack of prior knowledge regarding the behaviour of the system, choosing an appropriate model structure is a significant challenge. Nevertheless, many nonlinear identification methods have been developed by imposing a known model class on the system of interest [6]. In identification literature, three general approaches emerged in the latter half of the 20th century: block-oriented modeling [7], Volterra series estimation [107] and NARMAX¹ modeling [69]. The block-oriented approach combines linear dynamic and static nonlinear blocks in a fixed structure, thus requiring the strongest assumptions on model class. The latter two approaches apply to a much broader class of nonlinear systems, however at the time of their introduction the experimental and computational requirements were prohibitive for any non-trivial problem.

Prior to the emergence of system identification, the Volterra series had already gained some popularity as a tool for nonlinear system analysis [135], [57]. The concept is attractive from a theoretical standpoint, as it is a natural extension of the widely known, and used, Taylor series. In practice, however, the Volterra series was too complicated to be used in analyzing all but the most simple of systems, and it was not until the widespread availability of high performance computing in very recent years that it gained significant momentum in the literature for systems theory [26]. The same perspectives on Volterra series were then mirrored in the system identification community. Again, the series was an appealing option in theory due to its ability to model *any* nonlinear system which is time invariant and has fading memory [16], all but eliminating the need for prior knowledge of the system under study. In reality, however, the multidimensional nature of the series implies a requirement for large numbers of parameters and long data records, and hence a high computational burden to match.

There has been some progress in developing strategies to mitigate the computational and data length requirements associated with Volterra series parameter estimation. Some approaches require specifically designed input signals, such as Gaussian white noise [108] or a pseudorandom multilevel sequence (PRMS) [88]. Another common technique is to expand the series using a set of orthogonal or wavelet basis functions [26], thereby reducing the number of parameters to be estimated. Most recently, a Bayesian identification method from the linear system literature was extended to Volterra series in [11], which employs a computationally intensive optimization procedure to dramatically increase estimation accuracy for short data records.

Like any system model, Volterra series models have many uses, but control is undeniably the most pertinent application in an engineering context. Control refers to the practice of manipulating system inputs in order to obtain a desired output trajectory, where a mathematical model of the system is a useful tool in optimizing

¹Nonlinear Auto-Regressive Moving-Average with eXogenous inputs

the control strategy. Traditionally, linear models were used to tune negative feedback control structures such as the Proportional-Integral-Derivative (PID) and state feedback architectures still in wide-scale use today (see e.g. [49]). Another type of model-based control, known as Model Predictive Control (MPC), emerged almost in isolation from academic literature, through necessity in the process industries [78]. Due to the highly complex, multivariable, and often nonlinear nature of many engineering processes, and the large number of hard control constraints, traditional feedback structures were infeasible in many industrial settings. On the other hand, MPC was a conceptually simple and easily implementable alternative which found remarkable success in many engineering applications, and has since been adopted and studied extensively by the academic world [73], [101].

The use of Volterra series models in control design has, to this point, been extremely limited. This is of course unsurprising, since the identification of such models has historically been impractical in most real engineering settings, and the complicated nature of the models adds significant theoretical and computational complexity to online control calculations. In the late 1990s and early 2000s, some theoretical work was produced on feedback control and MPC using Volterra series, the majority of which can be found in [34], however the results are limited to low order models.

1.2 Motivation

Nonlinear system identification is a difficult task, but one where the Volterra series representation shows great theoretical promise. Historically, the series was too complex to be useful in practice for analysis or identification, however rapid developments in both computing technology and estimation algorithms have left Volterra series models poised on the edge of mainstream adoption.

There are still several remaining challenges in enabling the widespread use of Volterra series for modeling and control of nonlinear systems. Three of these challenges will form the motivational basis for this thesis, and are listed below:

1. *In the most general identification setting, where data records can be short, noisy and arbitrary in nature, Volterra series identification is still not feasible for higher order models (3rd order and higher). The Bayesian technique used in [11] requires too many parameters in the representation, and its associated optimization problem scales poorly with series order, yielding prohibitively large computational and memory requirements.*
2. *While the method described in [11] constitutes significant progress for time domain identification, equivalent techniques have not yet been developed for identification of Volterra series models directly in the frequency domain.*

3. *No framework exists for designing efficient model predictive control algorithms using generic high order Volterra models.*

1.3 Thesis outline and contributions

In this thesis, the three central challenges for Volterra series modeling and control highlighted in Section 1.2 will be explored in detail. The thesis is organized in four parts, with one part used to establish some necessary preliminaries, and separate parts dedicated to each of the three challenges.

Following the introduction, Part I contains two preliminary chapters with the following structure:

- Chapter 2 introduces some fundamental concepts in system identification and control, with a focus on the estimation methods and control paradigms that will feature prominently in later chapters. In particular, Bayesian regularization techniques and the MPC framework are discussed in depth.
- Chapter 3 presents some necessary background on the Volterra series representation of dynamical systems, in both the time and frequency domain. General properties are discussed, and some overarching assumptions are established which will remain valid in the sequel. The extension of Bayesian regularization techniques to the Volterra series case, developed in [11], is also explained since this concept is crucial to many of the contributions in subsequent chapters.

Part II contains all contributions pertaining to the first challenge, i.e. the feasibility of high order Volterra series identification in the time domain. The contributions are organized as follows:

- Chapter 4 introduces the statistical method known as expectation-maximization (EM) [32], and derives the associated E- and M-steps for the case of Volterra series identification in a Bayesian setting. The application of EM is shown to reduce the large optimization problem required in [11] into a set of smaller problems in an iterative scheme, thereby improving the computation time scaling of the identification method with Volterra series order. The results are demonstrated through a simulation example.
- Chapter 5 investigates the use of orthonormal basis function expansions to reduce the number of parameters needing estimation in a Volterra series model. In particular, the possibility of applying Bayesian regularization directly to the basis function expansions is considered. The results from Chapter 4 are extended to produce a modified EM algorithm capable of optimizing both the original identification hyperparameters as well as the parameters which

generate basis functions for each nonlinear order. Numerical examples are used to illustrate how the combination of Bayesian regularization, basis functions and EM-based optimization can produce accurate Volterra series estimates up to the 5th order under extreme conditions, whilst the method from [11] becomes infeasible after the 2nd order.

- Chapter 6 complements the work contained in Chapters 4 and 5, by assessing their contributions through two practical nonlinear example systems. Both of the systems presented are established nonlinear benchmarks in the identification community, and provide only short data records for estimation, making them a perfect test bed for addressing the challenge considered in this part of the thesis.

The first benchmark system consists of two cascaded water tanks [113], combining nonlinear fluid dynamics with a hard saturation nonlinearity from tank overflow. The identification method developed in Chapter 5 is compared against the method from [11] to highlight the reduction in parameters and computation time which can be achieved using the techniques proposed in this thesis.

The second system consists of two coupled electric drives [136] connected via a flexible belt and spring to form a resonant configuration. An absolute value nonlinearity is present in the output sensor. The accuracy of a second order Volterra model estimated using the method of Chapter 5 is shown to be superior to many other models estimated using established identification routines.

Part III addresses the second motivating challenge of this thesis, that Bayesian regularization techniques have not yet been developed for direct frequency domain identification of Volterra series models. Three chapters are contained within this part, with the following contributions:

- In Chapter 7, the identification problem is considered for the specific case of parallel Hammerstein systems; a subset of the block-oriented model structures. In this case, the standard frequency domain Volterra series can be reduced to a set of one-dimensional frequency functions known as Nonlinear Output Frequency Response Functions (NOFRFs). While the NOFRFs are used to describe each nonlinear order of the system, it is shown that the Hammerstein structure causes each frequency function to behave like a linear filter. This leads to the development, in this thesis, of a Bayesian estimation method extended from techniques in the linear frequency domain literature [67].
- Chapter 8 considers the more general identification problem, and provides the theoretical framework for an equivalent frequency domain method to that of [11]. The frequency domain terms of a Volterra series model, commonly

known as Generalized Frequency Response Functions (GFRFs), are shown to be identified accurately from just one period of a steady-state data record, despite severe rank deficiency in the regression problem.

- Chapter 9 removes the steady-state assumption imposed in Chapter 8, and examines the behaviour of nonlinear Volterra systems in the frequency domain when their input excitation is arbitrary. In this case, the system response contains both steady-state and transient contributions. The behaviour of the transient contribution has not been well studied in the literature for nonlinear systems, hence we derive analytic expressions in the chapter which describe the entire set of response contributions produced by a Volterra system in a highly structured manner. In doing so, important implications are revealed for frequency domain identification when steady-state measurements cannot be used.

Part IV shifts the focus of the thesis towards control in addressing the final challenge; efficient MPC using high order Volterra models. Theory and implementation are discussed in the following two chapters:

- In Chapter 10, motivated by the success of the identification algorithm designed in Chapter 5, a model predictive control algorithm is developed for arbitrary order Volterra models described by Laguerre basis functions. A nonlinear state space model framework is proposed, which is then used to construct a typical MPC objective. Solving this objective at any given sample time is shown to reduce to an optimization problem that is computationally simple and can be easily implemented online, with or without input constraints. Some notes on stability are also presented for the algorithm, highlighting the importance of the input penalty term in the objective function.
- Chapter 11 demonstrates the efficacy of the MPC method proposed in Chapter 10, using two water level control problems. The first problem considers level control for the cascaded water tanks benchmark introduced in Chapter 6, however the control performance is assessed here in simulation using a nonlinear state space model for the true system. To complement this, identification and control experiments are also performed on a real coupled water tanks apparatus. Both the simulation and experimental results show superior control performance from the proposed MPC when compared to controllers using simplified or linearized model structures.

Finally, a concluding summary of contributions is provided at the end of the thesis in Chapter 12. A number of promising pathways for future investigation are also discussed.

1.4 Publication list

All publications resulting from the research conducted in this thesis are listed below, along with the chapters which they relate to. The works are categorized according to their publication status.

Published conference proceedings:

- J. G. Stoddard and J. S. Welsh. Regularized basis function estimation of Volterra kernels for the cascaded tanks benchmark. In *Proceedings of the 18th IFAC Symposium on System Identification*, pages 413–418, 2018.
(Chapter 6)
- J. G. Stoddard and J. S. Welsh. Data-driven modeling of a coupled electric drives system using regularized basis function Volterra kernels. In *Intelligent Robotics and Applications*, pages 475–485. Springer International Publishing, 2018.
(Chapter 6)
- J. G. Stoddard and J. S. Welsh. Frequency domain estimation of parallel Hammerstein systems using Gaussian process regression. In *Proceedings of the 18th IFAC Symposium on System Identification*, pages 1014–1019, 2018.
(Chapter 7)
- J. G. Stoddard and J. S. Welsh. Improved nonlinear model predictive control with Volterra-Laguerre models. In *Proceedings of the 2019 American Control Conference*, pages 1014–1019, 2019.
(Chapters 10,11)

Published journal papers:

- J. G. Stoddard, J. S. Welsh, and H. Hjalmarsson. EM-based hyperparameter optimization for regularized Volterra kernel estimation. *IEEE Control Systems Letters*, 1(2):388–393, 2017.
(Chapter 4)
- J. G. Stoddard, G. Birpoutsoukis, J. Schoukens, and J. S. Welsh. Gaussian process regression for the estimation of generalized frequency response functions. *Automatica*, 106:161–167, 2019.
(Chapter 8)
- J. G. Stoddard, G. Birpoutsoukis, J. Lataire, and J. S. Welsh. Frequency domain response under arbitrary excitation for fading memory nonlinear systems. *Automatica*, 107:327–332, 2019.
(Chapter 9)

Journal papers under review:

- J. G. Stoddard, J. S. Welsh, and H. Hjalmarsson. Volterra kernel identification using regularized orthonormal basis functions. *IEEE Transactions on Automatic Control*, Under Review, 2018.

(Chapter 5)

PART

I

Preliminaries

CHAPTER

2

System Identification and Control

This chapter contains a brief introduction to concepts within the fields of system identification and control. Dynamical systems and the process of identification are discussed broadly, with particular focus on the impulse response model and its associated regularized least squares estimator. Some background on control theory is provided, along with a description of the model predictive control (MPC) framework adopted in this thesis.

2.1 Dynamical systems

A system can be defined as any transformation between an input signal and output signal. Denoting the system by \mathcal{S}_0 , and the input/output signals by u_0 and y_0 respectively, we will leave the nature of these terms unspecified for the moment. In order to interact with the system, and in particular take measurements of the input and output quantities, the presence of measurement noises should also be acknowledged. Thus, a complete picture of the system and its measured signals is shown in the block diagram of Figure 2.1, where v_u and v_y denote noise signals which corrupt the measurement of u_0 and y_0 respectively, with u and y representing the noisy measured signals.

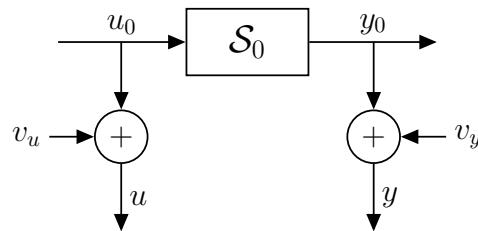


Figure 2.1: Generic block diagram of a system with input and output measurement noise

Before considering identification or control, the nature of the system under study and its associated signals must be further clarified. There are a number of system characterizations which must be determined (or chosen), which are listed and defined below:

1. **Dynamic/Static:** A system which is dynamic has the property that the current output, denoted by $y(t)$, depends not only on the current input, $u(t)$, but also on past values of the input and/or output, $u(t - \tau)$ and $y(t - \tau)$ for $\tau > 0$. It is equivalent to say that dynamic systems possess memory. Conversely, a static system is a memory-less system, where the current output is dependent only on the current input.
2. **Linear/Nonlinear:** A system is defined as linear if it satisfies the properties of homogeneity and additivity, i.e. the following condition is true:

$$\mathcal{S}_0 : u_1 \rightarrow y_1 \text{ and } \mathcal{S}_0 : u_2 \rightarrow y_2 \implies \mathcal{S}_0 : \alpha u_1 + \beta u_2 \rightarrow \alpha y_1 + \beta y_2 \quad \forall \alpha, \beta. \quad (2.1)$$

If this condition is not satisfied in general, the system is nonlinear.

3. **Time-invariant/Time-varying:** A time-invariant system does not change the nature of its input-output transformation over time, while for time-varying

systems, the time at which a given input is applied can change the resulting output.

4. **Discrete-time/Continuous-time:** A system which is discrete-time performs transformations using sampled input and output signals. This means that the signals $u(t)$ and $y(t)$ are defined only for specific values of t . When the samples are regular (evenly spaced), the argument t is typically restricted to integer values to represent the sample number. Continuous-time systems, as the name suggests, consider input signals which are continuous functions and produce a corresponding continuous output signal. It is worthwhile to note that while virtually all physical systems are continuous in nature, the discrete-time framework is useful in our digital world where measurements are taken and stored as discrete samples.
5. **SISO/MIMO:** A single-input-single-output (SISO) system possesses scalar input and output signals. Conversely, a multiple-input-multiple-output (MIMO) system may have multiple signals at the input and output ports. This means that the signals $u(t)$ and $y(t)$ at any instance of t are *vectors* containing multiple scalar components.

Many of the above characterizations should be considered as mathematical approximations, rather than concrete physical properties of the system. For example, it would be fair to say that no real system is truly linear, however there are many systems for which a linear model is satisfactory in capturing the important dynamic behaviour for prediction or control purposes [96]. Indeed, the entire pursuit of prescribing a mathematical model to a real process should be considered an abstraction from reality, since an exact description is never attainable. This philosophical perspective is eloquently presented in [72]:

“In a sense, there is an impenetrable but transparent screen between our world of mathematical descriptions and the real world. We can look through this window and compare certain aspects of the physical system with its mathematical description, but we can never establish an exact connection between them.”

(Lennart Ljung)

Nevertheless, all systems must be characterized in a mathematical sense prior to modeling. In this thesis, we will focus on SISO systems which are well approximated by a discrete-time, dynamic, nonlinear, time-invariant description.

2.2 The identification process

When it comes to the identification of a dynamical system from experimental data, there are a number of distinct steps in the process. Naturally, much of the attention

in system identification literature is focussed on estimation algorithms, and how to transform the measured data into an accurate system model. However, before the estimation stage there are other considerations to be made. These considerations include the experimental procedure which should be designed in order to obtain informative data from the system, the model structure which must be chosen for estimation, and a metric which should be constructed to quantify the performance of candidate models during estimation. Even after a model estimate has been obtained, there is further work in validating the model on separate data to ensure it meets specifications for the intended use.

An overview of the full identification process is depicted in Figure 2.2. Note that many of the stages can be informed by prior knowledge of the system, as well as previous unsuccessful estimation attempts. The latter point highlights the iterative nature of system identification. It can be difficult to choose the experimental conditions, model class, etc. in an optimal way without yet having a mathematical system description, however those initial attempts which do not meet specifications can provide information for revising and improving the process.

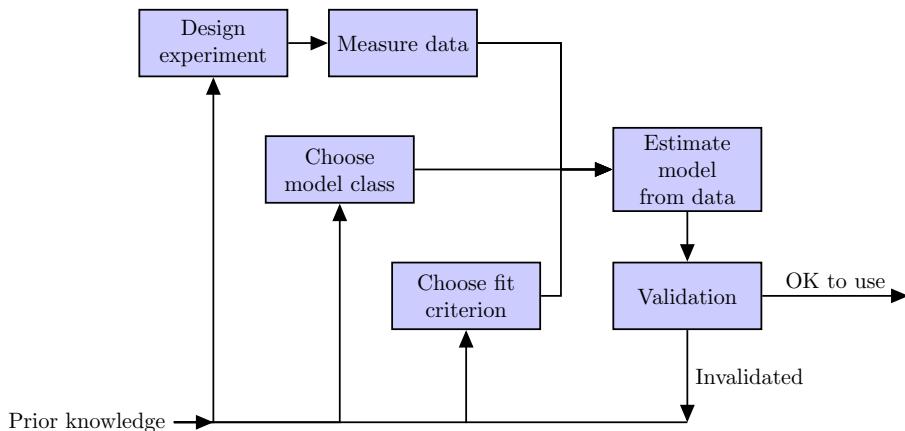


Figure 2.2: An overview of the identification process

Background will be provided in the following subsections for each stage in the identification process. These sections will also provide some context regarding each stage in terms of this thesis.

2.2.1 Experiment design

Broadly speaking, the objective when designing an identification experiment is to ensure that the measured data provides maximal information with respect to the intended use of the system model [47]. The experiment design should cater to any practical constraints which are present, and consider factors such as experiment duration, presence of stabilizing feedback, and most importantly, the nature of the excitation signal. Optimal input design has itself become a rich sub-field, which

considers how best to choose the class of signal used for excitation, as well as its power spectrum and range [13], [54].

Experiment design is not a crucial element of this thesis. Several signal classes are used for excitation, including Gaussian white noise, pseudo-random binary sequence (PRBS), and random phase multisine. White noise and PRBS have an approximately flat power spectrum across the entire frequency band, while multisines allow precise control over the power contained at each frequency [111], being explicitly constructed from the sinusoid summation:

$$u_0(t) = A_0 + \sum_{i=1}^F A_i \cos(2\pi i f_0 t + \phi_i), \quad (2.2)$$

where A_0 gives the constant offset, f_0 is the fundamental frequency (in Hz), F is the maximum harmonic excited, and ϕ_i are randomly selected phase shifts.

2.2.2 Choosing a model class

A plethora of model classes exist for describing dynamical systems, and a selection here has consequences for the estimation methods which can be used as well as the suitability of the estimated model for its intended use. The first choice which should be made is whether to use a parametric or nonparametric representation. While parametric models have a fixed and finite number of parameters, nonparametric models are characterized by a parameter set which is infinite in theory and in practice can grow indefinitely with increasing data length. Another crucial question is whether to construct a model in the time or frequency domain, i.e. whether to express the input to output transformation in terms of time domain or frequency domain signals.

For discrete linear systems, popular model classes include the parametric AR-MAX, Box-Jenkins and state space structures, as well as the nonparametric impulse response model and its frequency domain equivalent, the frequency response function (FRF). For nonlinear systems, as discussed in Chapter 1, there are a number of representations available. NARMAX and block-oriented models can be fully or partially parametric, while the Volterra series and its related expansions are nonparametric. The Volterra series also has a frequency domain equivalent formulation comprised of GFRFs. This thesis assumes a preference for nonparametric modeling, and deals with Volterra series estimation in both time and frequency domain. Other nonlinear model classes, e.g. block-oriented structures, are sometimes used to construct a simulated ‘true system’ on which identification is performed.

When a model class has been selected, it defines a set, \mathcal{M} , of all possible models for the system. The set is defined as

$$\mathcal{M} := \{\mathcal{S}(\theta) | \theta \in \Theta \subset \mathbb{R}^n\} \quad (2.3)$$

where $\theta \in \mathbb{R}^n$ is a vector of parameters completely describing the model, Θ is the vector space of all possible parameter vectors, and $\mathcal{S}(\theta)$ is the parameterized system model.

2.2.3 Choosing the fit criterion

Once experimental data has been collected and a model class chosen for a system, a metric $V(\theta)$ is defined in order to differentiate the performance of model candidates in the estimation stage. This metric will be labelled the ‘fit criterion’, which for any given θ assigns a numeric value representing how well the corresponding model fits the observed experimental data.

The fit criterion should be chosen based on the context in which the model will be used, but care must be taken since this choice will also have implications for computational complexity in the estimation phase. Typically, the criterion $V(\theta)$ is based on some measure of distance between the measured output and the model output predicted using $\mathcal{S}(\theta)$. Alternatively (and sometimes equivalently), $V(\theta)$ can be formulated as a statistical likelihood function which provides the likelihood of seeing the observed data given a model parameterized by θ and some assumed measurement noise distributions.

The estimation methods developed in this thesis use fit criterions which can be framed both in terms of a distance measure and a statistical function, with the relationship between the two criterions discussed in Section 2.5.

2.2.4 Model estimation

The specific details of model estimation will depend heavily on the model structure and fit criterion which have been chosen in the previous stages. In the most general case, model estimation can be framed simply as the optimization of a criterion $V(\theta)$ over the allowable space Θ . For example, if the criterion is a distance measure between the measured and predicted outputs, our model can be defined by the minimization problem:

$$\hat{\theta} = \arg \min_{\theta} V(\theta). \quad (2.4)$$

Depending on the complexity of the model and fit criterion, solving the optimization problem can be as simple as an analytic solution. Conversely, $V(\theta)$ may be highly non-convex, requiring techniques that can efficiently locate a global optimum. The estimation methods in this thesis will feature minimization problems of the latter type, and the efficiency of finding solutions is addressed in Chapter 4.

2.2.5 Validation

After obtaining a model estimate, it is important to test the model’s prediction accuracy on a new set of experimental data, known as validation data, which is

uncorrelated with the data used for estimation. This allows a user to assess whether the model meets the specifications for its intended purpose, and will also detect problems like overfitting, where some of the model parameters are inadvertently used to capture the noise in the estimation data rather than the underlying process.

2.3 The impulse response model

The impulse response model is a foundation stone of linear systems theory. For a discrete-time SISO system, it is defined as the output response to a unit impulse applied at sample time $t = 0$, i.e. $u(t) = \delta(t)$ where $\delta(t)$ denotes the unit impulse function,

$$\delta(t) = \begin{cases} 1 & \text{for } t = 0 \\ 0 & \text{for } t \neq 0 \end{cases} \quad (2.5)$$

This impulse response is denoted by $h(t)$, which can be infinite in length. The infinite impulse then forms a unique description for any linear system, since the system response to any arbitrary input is just a convolution of the input with the impulse response,

$$y(t) = \sum_{\tau=-\infty}^{\infty} h(\tau)u(t - \tau) = h(t) * u(t), \quad (2.6)$$

where $*$ is the convolution operator. It is clear that $h(t)$ represents a nonparametric model structure for linear systems, which will be denoted the infinite impulse response (IIR) model.

Models of infinite dimension are not useful in a practical setting, however there are some generic properties of a linear impulse response which are independent of the generating system, and these properties can be exploited to produce a finite-parameter model. First, we impose an assumption of causality (no system can predict the future), such that $h(t) = 0$ for $t < 0$. Thus, the negative tail of the impulse can be disregarded. We will also restrict our scope to the set of stable linear systems, which brings an additional implication that the impulse response tends to 0 over time:

$$h(t) \xrightarrow{t \rightarrow \infty} 0. \quad (2.7)$$

Under these conditions, it follows that any impulse response model can be truncated to a finite length, $t = 0, 1, \dots, n - 1$, where $h(n)$ is the first negligible component as dictated by some user-specified threshold. The truncation length n , will be referred to as the ‘effective memory length’ of the system. The truncated impulse forms a new nonparametric description which will be denoted the finite impulse response (FIR) model, and written mathematically as,

$$y(t) = \sum_{\tau=0}^{n-1} h(\tau)u(t - \tau). \quad (2.8)$$

Examples of truncated impulse responses are shown in Figure 2.3 for arbitrary linear systems.

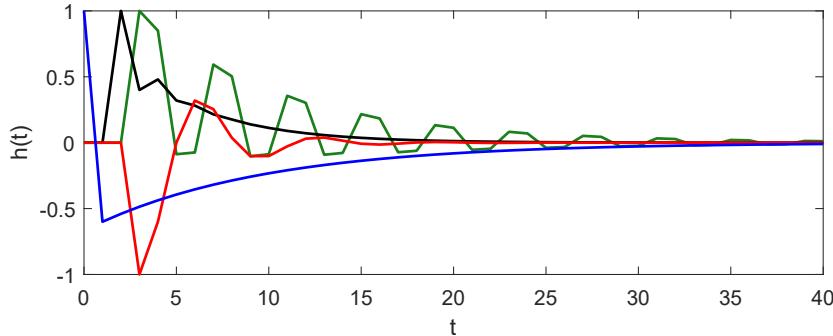


Figure 2.3: Example impulse responses for randomly generated discrete linear filters

2.4 Classical estimators

For the identification of dynamical systems, it is often possible to formulate the estimation problem such that classical estimation techniques can easily be applied. An example of this is when the system model is ‘linear-in-the-parameters’. Assuming we have some parameter vector $\theta \in \mathbb{R}^n$ which is a column vector, a linear-in-the-parameters model has output response at sample t given by:

$$y(t) = \phi(t)^T \theta + e(t) \quad (2.9)$$

where T is the vector transpose operator. The $e(t)$ term captures any error between the measured output and model, and $\phi(t) \in \mathbb{R}^n$ is known as the regressor vector, which can contain any mixture of past output and input samples depending on the model structure. If we consider the FIR model from (2.8), which is linear-in-the-parameters, then (2.9) holds with

$$\theta = [h(0) \ h(1) \ \dots \ h(n-1)]^T, \quad (2.10)$$

$$\phi(t) = [u(t) \ u(t-1) \ \dots \ u(t-n+1)]^T. \quad (2.11)$$

While (2.9) explains the output response for a single sample, t , the representation can be extended to model an entire output data record. Assuming N samples of input and output data ($t = 0, \dots, N-1$) are measured from a system, and considering the FIR model (2.8), an output vector $Y = [y(n-1) \ y(n) \ \dots \ y(N-1)]^T$ is constructed and modeled as,

$$Y = \Phi^T \theta + E \quad (2.12)$$

where E is a vector containing the errors $e(t)$, and $\Phi \in \mathbb{R}^{n \times (N-n+1)}$ is the regressor matrix, given by

$$\Phi = [\phi(n-1) \ \phi(n) \ \dots \ \phi(N-1)]. \quad (2.13)$$

2.4.1 The least squares approach

In order to estimate θ from (2.12), a fit criterion $V(\theta)$ is required. The least squares (LS) approach employs a sum of squared errors between the model and the measurement as such a criterion. Mathematically, the LS criterion can be expressed as,

$$V_{LS}(\theta) = E^T E \quad (2.14)$$

$$= (Y - \Phi^T \theta)^T (Y - \Phi^T \theta) \quad (2.15)$$

$$= \| (Y - \Phi^T \theta) \|_2^2, \quad (2.16)$$

where $\| \cdot \|_2^2$ denotes the squared 2-norm operator. The resulting optimization problem to obtain our estimate $\hat{\theta}$, is given by

$$\hat{\theta} = \arg \min_{\theta} \| (Y - \Phi^T \theta) \|_2^2, \quad (2.17)$$

which is easily shown to be a convex problem with the analytic solution,

$$\hat{\theta} = (\Phi \Phi^T)^{-1} (\Phi Y). \quad (2.18)$$

For the least squares approach, the interpretation of ‘error’ is quite general. it can represent output measurement noise, or model errors due to a mismatch between the model and true system, or a combination of both. Of course, the true source of these errors will have an effect on the properties of the least squares estimator, however before discussing this, the maximum likelihood estimator will be derived for our linear-in-the-parameters structure.

2.4.2 Maximum likelihood estimation

In maximum likelihood estimation (MLE), our fit criterion is chosen as the likelihood function, $\mathcal{L}(Y; \theta)$, which gives the likelihood of observing the output vector Y given some model structure parameterized by θ . In order to express the likelihood analytically, the system model should have some statistical properties imposed, and in particular for our linear-in-the-parameters representation (2.12), the stochastic behaviour of the error vector E should be defined.

One common assumption for E , and the one which will be discussed here, is that E is a vector of independent and identically distributed (i.i.d.) random variables each with a zero mean normal distribution. Observing Figure 2.1, this assumption

implies that we have Gaussian white measurement noise at the output ($v_y(t) = e(t)$), and no input noise ($v_u = 0$). This produces a stochastic model description for Y , i.e.

$$Y \sim \mathcal{N}(\Phi^T \theta, \sigma^2 I), \quad (2.19)$$

where σ^2 is the variance of the errors in E , and I is an identity matrix of appropriate size. An analytic expression for the likelihood, and thus the fit criterion, can be written using the multivariate normal distribution, yielding

$$V_{ML}(\theta) = (2\pi\sigma^2)^{-\frac{N-n+1}{2}} \exp\left(-\frac{1}{2\sigma^2} \left\| (Y - \Phi^T \theta) \right\|_2^2\right). \quad (2.20)$$

The optimization problem for estimating θ is simply to maximize the likelihood function,

$$\hat{\theta} = \arg \max_{\theta} V_{ML}(\theta). \quad (2.21)$$

It is trivial to show that under the assumption above on the error, and substituting (2.20) into (2.21), the problem will reduce to (2.17). Thus the maximum likelihood approach shares the same analytic solution (2.18) as the least squares approach for this case. This explains the popularity of the Gaussian measurement noise assumption in MLE, since it produces a familiar analytic solution. However, the estimator performance will degrade if there is significant input measurement noise, or the output noise is a correlated (non-white) sequence. In these circumstances, a more complex stochastic model may be required.

2.4.3 Estimator properties

Definition 2.1 (Estimator). *For a model set \mathcal{M} , an estimator is an operator which maps the measured data u and y to a parameter vector estimate $\hat{\theta}$.*

To analyze the properties of an estimator, the philosophical truth that real systems can never be fully described by abstract models will be abandoned out of necessity, and replaced with the pretence that there exists a true *mathematical* description of the system, S_0 . Under the further condition that this true system lies within our model set, i.e. $S_0 \in \mathcal{M}$, the notion of a true parameter vector can be introduced.

Definition 2.2 (True parameters). *The true parameter vector, θ_0 , is a vector satisfying $S(\theta_0) = S_0$.*

Any estimate $\hat{\theta}$ produced by an estimator is a function of input and output measurements obtained from the system, which contain stochastic noise in general. The estimate itself should therefore be treated as a random variable, and it is the statistical properties of this variable which define the performance of the estimation method. There are three properties in particular which, when combined, provide an

appreciation of the performance of the estimator. They are the bias, variance and mean square error (MSE).

Definition 2.3 (Bias). *The bias of an estimator is the difference between the expected value of the estimate and the true parameters:*

$$\text{Bias}(\hat{\theta}) = \mathbf{E}\{\hat{\theta}\} - \theta_0. \quad (2.22)$$

An unbiased estimator is one for which $\mathbf{E}\{\hat{\theta}\} = \theta_0$.

Definition 2.4 (Variance). *The variance of an estimator is its variability around the expected value, described by:*

$$\text{Var}(\hat{\theta}) = \mathbf{E}\left\{(\hat{\theta} - \mathbf{E}\{\hat{\theta}\})(\hat{\theta} - \mathbf{E}\{\hat{\theta}\})^T\right\}. \quad (2.23)$$

Definition 2.5 (MSE). *The MSE of an estimator combines the effects of bias and variance into a total error metric, given by:*

$$\text{MSE}(\hat{\theta}) = \mathbf{E}\left\{(\hat{\theta} - \theta_0)(\hat{\theta} - \theta_0)^T\right\} \quad (2.24)$$

$$= \text{Bias}(\hat{\theta}) \text{Bias}(\hat{\theta})^T + \text{Var}(\hat{\theta}). \quad (2.25)$$

Observing these three properties, it is clear that an accurate estimator should have the lowest possible bias and variance in order to minimize its MSE. It would seem natural then, to consider only unbiased estimators such that $\text{Bias}(\hat{\theta}) = 0$. Indeed, the LS and MLE estimators, (2.17) and (2.21), will be unbiased under the following conditions:

1. $\mathcal{S}_0 \in \mathcal{M}$
2. No input measurement noise
3. The output error E is any zero mean stochastic process

Note that in the MLE case, the Gaussian i.i.d. noise assumption need not be met to ensure unbiasedness. Variance, on the other hand, will decrease further under the correct noise distribution. There is a fundamental restriction on how far the variance can decrease in unbiased estimators, and this is known as the Cramér-Rao lower bound [96]. An unbiased estimator with variance obtaining the lower bound is called an *efficient* estimator.

When considering models which require many parameters, such as the nonparametric impulse response, the Cramér-Rao lower bound can present a problem. For example, when the number of samples N used in the estimation is low, or measurement noise is significant, even an efficient estimator will have very high variance

and hence is likely to produce an inaccurate model. In these cases, it can be beneficial to introduce a small bias into the estimator in such a way that the variance decreases dramatically along with the MSE. Typically, this bias reflects some prior belief on how the parameter vector should behave, and can be introduced by adding an additional penalty to the fit criterion (known as ‘regularization’), or by adopting a Bayesian estimation approach. These two techniques are intrinsically related and will form the core of all identification methods presented in this thesis.

2.5 Regularized least squares and the Bayesian perspective

The least squares approach to model estimation is attractive due to its simplicity, with an analytic solution guaranteed for any model structure that is linear-in-the-parameters as long there is enough measurement data available. In practice however, and particularly for nonparametric structures such as the FIR model, estimation using least squares will lead to poor results under many circumstances. The issues relevant to FIR estimation are:

- The estimator will have high variance if the number of data points N does not greatly exceed the number of parameters, n .
- The estimator will have high variance when output measurement noise is significant.
- The estimation problem will be ill-conditioned if the input excitation is not sufficiently exciting.
- The estimation problem does not have a unique solution when $N < 2n - 1$.

Observing these issues, it is clear that a long, well designed and relatively noise-free experiment is required to achieve good impulse response estimates, particularly when the impulse has a large effective memory length, n . There are many practical identification settings where such ideal experimental conditions are not possible, and consequently will require more advanced estimation techniques.

As an example, FIR model estimates are obtained for a simulated linear system using least squares on two sets of experimental data, where each experiment has Gaussian excitation $u(t) \sim \mathcal{N}(0, 1)$ and Gaussian measurement noise $e(t) \sim \mathcal{N}(0, \sigma^2)$. The first experiment records $N = 2000$ data samples and σ^2 is chosen such that the Signal-to-Noise Ratio (SNR)¹ is 40dB. The second experiment records only $N = 200$ samples with a decreased SNR of 20dB. The least squares estimated impulse response for each experiment is plotted in the left half of Figure 2.4, along with the true system, showing the performance degradation that comes with small

¹SNR is defined using the variance ratio: $\text{SNR} = \text{Var}[y_0]/\text{Var}[e]$

data lengths and high noise levels. The right half of Figure 2.4 plots *regularized* estimates for the same experiments, which are seen to be far less sensitive to decreases in data length and SNR. The mathematical techniques for performing regularized FIR estimation will now be discussed.

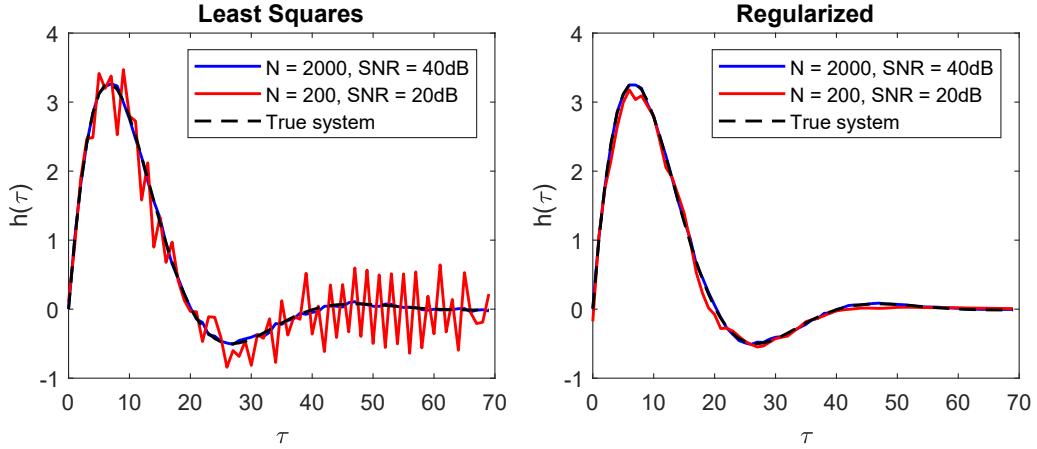


Figure 2.4: Least squares (left) and regularized (right) FIR estimates for an example linear system using different data lengths and SNRs

2.5.1 Regularization penalty options

In its most general form, regularization can be described as the inclusion of an additional term in the fit criterion of an estimator. The regularized least squares (ReLS) criterion is then given by,

$$V_{ReLS}(\theta) = \left\| (Y - \Phi^T \theta) \right\|_2^2 + T(\theta), \quad (2.26)$$

where $T(\theta)$ is an additional penalty term acting directly on the parameter vector. Such an addition can be used to impose certain expected behaviours on the estimated parameter vector, thereby reducing the effect of noisy measurements and improving the estimation accuracy. In ill-posed estimation problems, the added penalty can also improve conditioning.

Various forms have been proposed for T , starting with the early results of Tikhonov [128], for which the term ‘Tikhonov regularization’ was coined. The penalty for this type of regularization takes the general form:

$$T(\theta) = \|\Gamma \theta\|_2^2, \quad (2.27)$$

where Γ is a Tikhonov matrix chosen to impose certain parameter behaviours. When the Tikhonov matrix is chosen as a multiple of the identity matrix, i.e.

$$T(\theta) = \gamma \|\theta\|_2^2, \quad (2.28)$$

for some $\gamma \in \mathbb{R}^+$, then the regularization method is known within the statistics community by the term ‘ridge regression’ [56]. This scheme is particularly useful for parameter vectors which are expected to be sparse, since parameters are heavily penalized for being too far from zero.

A more recent proposal for regularized estimation of sparse parameter vectors is the lasso (Least Absolute Shrinkage and Selection Operator) algorithm [127], which replaces the 2-norm penalty with a 1-norm, giving

$$T(\theta) = \gamma \|\theta\|_1. \quad (2.29)$$

The lasso approach has been shown to exhibit some nice properties with respect to capturing the set of zero elements in the true parameter vector, θ_0 [129].

Yet another similar technique is nuclear norm regularization [40], [83], which uses penalty term

$$T(\theta) = \gamma \|H(\theta)\|_*, \quad (2.30)$$

with $H(\theta)$ denoting the Hankel matrix of θ , and $\|\cdot\|_*$ being the nuclear norm operator, defined as the sum of singular values of its matrix argument. The nuclear norm has been shown to be a convex approximation of the matrix rank function [40], and for θ given by an impulse response model (as in (2.10)), penalizing the Hankel matrix rank is equivalent to penalizing the order (i.e. complexity) of the linear system.

2.5.2 The quadratic penalty

One penalty term in particular has garnered a significant amount of attention in recent decades, first in the machine learning field for function estimation [100], before being introduced to the system identification community as a method of low variance impulse response estimation [94]. The penalty is quadratic in nature, having the form,

$$T(\theta) = \gamma \theta^T P^{-1} \theta. \quad (2.31)$$

Comparison with (2.27) reveals this term to be just a special case of Tikhonov regularization, where $\gamma P^{-1} = \Gamma^T \Gamma$.

The total regularized estimation problem with the quadratic penalty (2.31) can be stated as

$$\hat{\theta} = \arg \min_{\theta} \left\| (Y - \Phi^T \theta) \right\|_2^2 + \gamma \theta^T P^{-1} \theta, \quad (2.32)$$

which is a convex problem with an analytic solution,

$$\hat{\theta} = (P \Phi \Phi^T + \gamma I)^{-1} P \Phi Y. \quad (2.33)$$

The terms γ and P are tunable, with P enforcing certain behaviours in the estimated parameter vector and γ scaling the effect of the regularization, much like

generic Tikhonov regularization. It is the specific techniques for designing and tuning these terms, however, which has recently received an explosion of interest [95]. To understand how these techniques emerged, it is important to first look at the MSE property of our regularized estimator (2.33).

It will be assumed that we have the linear-in-the-parameters model (2.12) with true parameter vector θ_0 , and i.i.d. errors such that $\mathbf{E}\{EE^T\} = \sigma^2 I$. Under these conditions, it can be shown that the MSE of (2.33) is minimized (in the matrix sense) by the following choices [25]:

$$\gamma = \sigma^2, \tag{2.34}$$

$$P = \theta_0 \theta_0^T. \tag{2.35}$$

It is clear that an optimal choice of P relies on knowing the true parameter vector, a fact which is not particularly useful in an estimation setting. The form of (2.35) does, however, lead to two useful mathematical interpretations for tuning P .

The first method interprets P as the reproducing kernel of a Hilbert space containing all allowable model parameterizations (see e.g. [110]), and in this context the estimation process is labelled ‘kernel-based regularization’. The second method, and the one which will be adopted in this thesis, is a Bayesian interpretation of θ_0 and therefore P . If instead of viewing the system as deterministic, θ_0 is expressed as a random variable with some covariance $\mathbf{E}\{\theta_0 \theta_0^T\} = \Pi$, then (2.35) can be replaced with $P = \Pi$. Tuning of the covariance can then be performed via established Bayesian methods, hence this form of regularization is often termed Bayesian regularization, or Gaussian Process Regression (GPR) when normal distributions are employed.

2.5.3 Bayesian interpretation

Having established that a Bayesian perspective may be useful in the regularized least squares problem (2.32), the problem can be reframed as a Bayesian estimation problem. In doing so, the first step is typically to reinterpret the true parameter vector θ_0 as a zero mean Gaussian process:

$$\theta_0 \sim \mathcal{N}(0, P). \tag{2.36}$$

As is the case for Bayesian estimation in general, this assumption does not indicate that the physical system is truly stochastic in nature. Rather, it is a convenient mathematical abstraction which can be used to express our prior belief on how the system should behave. It is for this reason that P will be labelled the *prior covariance* matrix.

When the system is represented by a FIR model structure, there are two behaviours in particular which are encoded into the Gaussian covariance, P :

1. The impulse response decays exponentially towards zero over time.
2. The impulse response is a smooth function.

The decay behaviour can be represented by an exponential decrease in parameter variance along the diagonal of P , while smoothness is induced by increasing the off-diagonal correlation components of parameters which are in close proximity to each other. Several covariance structures of varying complexity have been proposed in the literature to capture such behaviour [94], [24], where the two most typical choices are,

Tuned/Correlated (TC):

$$\begin{aligned} P(x, y) &= c\lambda^{\max(x,y)}, \\ c \geq 0, \quad 0 \leq \lambda < 1, \quad \eta_P &= [c, \lambda]. \end{aligned} \tag{2.37}$$

Diagonal/Correlated (DC):

$$\begin{aligned} P(x, y) &= c\lambda^{(x+y)/2}\rho^{|x-y|}, \\ c \geq 0, \quad 0 \leq \lambda < 1, \quad |\rho| \leq 1, \quad \eta_P &= [c, \lambda, \rho]. \end{aligned} \tag{2.38}$$

In the expressions (2.37) and (2.38), $P(x, y)$ denotes the x, y^{th} element of P , and η_P is a vector containing all tunable hyperparameters for the covariance matrix.

Considering once again the linear-in-the-parameters structure (2.12), the error vector E is assumed, for mathematical convenience, to be white Gaussian measurement noise, i.e. $E \sim \mathcal{N}(0, \sigma^2 I)$. The result is a fully Gaussian framework where the joint distribution of the output and parameter vector can be easily found as,

$$\begin{bmatrix} \theta \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} P & P\Phi \\ \Phi^T P & \Phi^T P\Phi + \sigma^2 I \end{bmatrix}\right). \tag{2.39}$$

Assuming for the moment that P is known, producing an estimate for θ via the Bayesian approach requires a posterior distribution conditioned on the observed data, $p(\theta|Y)$. Forming conditional distributions for jointly Gaussian variables is a well-established practice which yields the following normal posterior,

$$p(\theta|Y) \sim \mathcal{N}\left((P\Phi\Phi^T + \sigma^2 I)^{-1}P\Phi Y, P - P\Phi(\Phi^T P\Phi + \sigma^2 I)^{-1}\Phi^T P\right) \tag{2.40}$$

The maximum a posteriori (MAP) estimate for θ , conditioned on Y , is simply the mean of the distribution,

$$\hat{\theta} = (P\Phi\Phi^T + \sigma^2 I)^{-1}P\Phi Y, \tag{2.41}$$

which is identical to the regularized least squares solution (2.33) with $\gamma = \sigma^2$. Thus, a fully Bayesian approach provides the same analytic solution as quadratically penalized least squares.

We can now proceed one step further and take advantage of our Bayesian framework to tackle the problem of appropriately tuning P . Noting that the noise variance, σ^2 , is also typically unknown, the most common method is to collect all tunable hyperparameters into a single vector $\eta = [\eta_P, \sigma^2]$ and consider the marginal likelihood, $\mathcal{L}(Y; \eta)$, where θ has been marginalized out:

$$\mathcal{L}(Y; \eta) \sim \mathcal{N}(0, \Sigma_Y(\eta)), \quad (2.42)$$

where $\Sigma_Y(\eta) = \Phi^T P(\eta_P) \Phi + \sigma^2 I$ from (2.39). By maximizing the marginal likelihood function, hyperparameters can be tuned purely on the observed system data, a technique known as empirical Bayes.

Maximizing (2.42) reduces to the minimization problem,

$$\hat{\eta} = \arg \min_{\eta} Y^T \Sigma_Y^{-1} Y + \log \det \Sigma_Y, \quad (2.43)$$

which is in general a non-convex optimization problem with multiple local minima [100]. Methods to improve robustness and computation time have been explored in the literature, in particular by employing QR factorization to avoid the inversion of large matrices [22]. Once η has been tuned, the model estimate $\hat{\theta}$ follows immediately via (2.41).

2.6 Model-based control

The field of control has a long and interesting history spanning back to ancient times, however the development of modern control theory began in the 20th century. In particular, the World War era and subsequent space race provided the catalyst for rapid advancements in our understanding of feedback control [49]. To effectively design and analyze feedback control architectures, mathematical models of the dynamical systems were required, and so the fields of system identification and control evolved in closely related trajectories.

While model-based feedback controllers are now ubiquitous in everything from manufacturing and transport to telecommunications and finance, there are still applications where such control schemes are difficult or impossible to implement effectively. For example, control problems which involve strong nonlinear dynamics, large numbers of inputs and outputs, or complex constraints are not well suited to traditional feedback control. In the process industries, where such complexities are common, the Model Predictive Control (MPC) approach emerged, and has since received an explosion of academic interest [18]. In this thesis, we consider model-based control from the perspective of Volterra series modeling, where the nonlinear

dynamics may be arbitrarily dominant. Thus, the MPC framework will be utilized to achieve efficient Volterra series model-based control.

2.6.1 MPC formulation

There are a large number of MPC variations which have been developed in the literature, all with different benefits and intended application areas. These variations fall into several broad categories including explicit, robust and stochastic MPC. In this section, a simple generalized formulation will be introduced which is sufficient for tackling the Volterra series control problem in Part IV of this thesis.

Consider a time-invariant discrete-time dynamical system which can be described by the state space equations,

$$x(t+1) = f(x(t), u(t)), \quad (2.44)$$

$$y(t) = g(x(t)), \quad (2.45)$$

where $x(t)$ is a vector of internal states at sample t , $u(t)$ and $y(t)$ are the system input and output respectively, f is a function describing the state evolution, and g is a static function mapping the states to the output. Equation (2.44) is known as the *state equation*, while (2.45) is labelled the *output equation*. The functions f and g can be nonlinear in general, as will be the case in this thesis.

The typical control objective is considered, i.e. for the output to track a desired reference signal, $r(t)$. In MPC, such an objective is achieved through the use of a cost function which is repeatedly optimized over a finite receding prediction horizon.

Suppose that some penalty function has been designed to describe the desired behaviour of the input and output signals, $\ell(y(t), u(t), r(t)) = \ell(g(x(t)), u(t), r(t))$, in a given sample. Looking first at the sample time $t = 0$, the MPC optimization problem for this case can be written as,

$$\begin{aligned} u^* = & \arg \min_{u(\cdot)} \sum_{t=1}^p \ell(y(t)|x(0), u(t), r(t)) \\ & \text{subject to } y(t) = g(x(t)), x(t+1) = f(x(t), u(t)), \\ & u(t) = u(t-1) \forall t > \mu, u(t) \in \mathcal{U} \forall t, x(t) \in \mathcal{X} \forall t. \end{aligned} \quad (2.46)$$

The optimal control sequence, u^* , is found by minimizing a sum of model-predicted penalty functions, ℓ , over the prediction horizon, p . Some additional constraints may also be present, for example there may be a move horizon $\mu < p$ after which the input remains constant until the end of the prediction horizon. The input and states may also be constrained to some allowable subsets, \mathcal{U} and \mathcal{X} , of their total vector space. Note that the initial states, $x(0)$, are assumed known here, such that robust MPC principles are not required.

Following the MPC optimization in (2.46), only the *first* input in the optimal

sequence, $u^*(1)$, is applied to the system. Then, at the next sample time, $t = 1$, the prediction horizon recedes to sample $p + 1$, and the same optimization procedure is repeated to find a new optimal control sequence. Once again, only the first value of this optimal sequence is applied, and the process continues ad infinitum. Since the optimization at any sample time t is performed identically to the case in (2.46), we can assume without loss of generality that $t = 0$ at every sample time, and use (2.46) to compute the next input sequence. A diagram is provided in Figure 2.5 to illustrate the optimization process occurring at any given sample time.

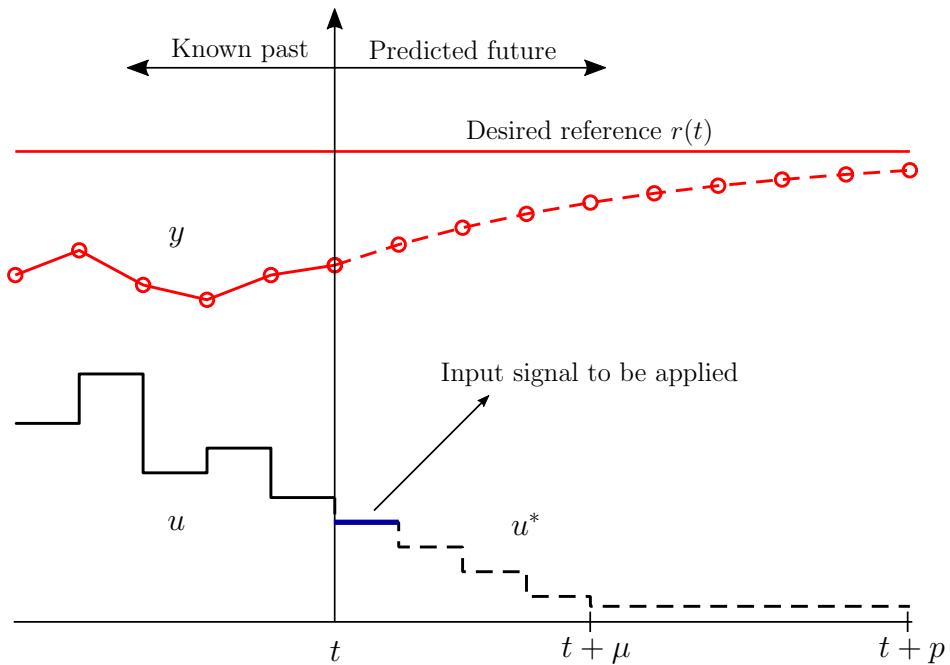


Figure 2.5: A visual example of the MPC optimization problem at sample time t . The optimal input sequence is restricted to a move horizon, μ , and the corresponding model-predicted output is also shown.

2.6.2 MPC stability

Stability is a crucial property in control system design. Generically, and assuming a constant reference $r(t) = r$, a stable control system is one in which the output can be *guaranteed* to remain bounded for all time. For a complete definition of stability, there are sometimes more subtle considerations to be made, for example the internal states of the system should also remain bounded, as well as the control input.

In the earliest days of MPC implementation, the stability of the algorithms was not well understood. While there is certainly a strong relationship between MPC and the more classical feedback control concepts of infinite horizon control and dynamic programming, the now finite horizon and input/state constraints add to the complexity of theoretical analysis [79].

For the simplest case of unconstrained linear MPC, it is sometimes possible to construct an explicit feedback control law from the receding horizon optimization in (2.46). Thus, stability can be assessed in the same way as for traditional feedback control, by performing an eigenvalue analysis of the linear dynamic state equation in (2.44). For all other cases, stability arguments can be made using the *optimal value function*,

$$V_p(x(0)) = \min_{u(\cdot)} \sum_{t=1}^p \ell(y(t)|x(0), u(t), r(t)). \quad (2.47)$$

One common approach is to prove that, $\forall x(0) \in \mathcal{X}$, the optimal value function monotonically decreases with each iteration. The receding horizon nature of the problem implies that it is sufficient to show the decrease in any one iteration, i.e.

$$V_p(f(x(0), u^*(1))) \leq V_p(x(0)). \quad (2.48)$$

If such a condition is satisfied, and the penalty function $\ell(y(t), u(t), r(t))$ is positive definite and equal to zero only at the desired input/output setpoint, then stability can be easily shown [79],[52].

2.7 Summary

Some fundamental concepts in system identification and control were presented in this chapter. The notion of a dynamical system was introduced, and the various stages involved in identifying such systems were discussed. Particular focus was given to the nonparametric impulse response model, which is used to motivate the advantages and drawbacks of common estimation methods such as least squares and maximum likelihood. Some popular regularization techniques were described, with an extensive coverage of quadratically penalized least squares as seen from a Bayesian perspective, which will feature prominently throughout the thesis. Finally, a generic model predictive control formulation was outlined for use in the final part of this thesis.

Many of the concepts introduced in this chapter will be extended in Chapter 3, where the linear impulse response model is generalized to higher dimensions to produce the Volterra series model. In particular, a method for extending the Bayesian regularization method to Volterra series terms will be explained.

CHAPTER

3

The Volterra Series

This chapter introduces the Volterra series model for nonlinear dynamical systems. Time and frequency domain representations are defined for the discrete-time case, where some assumptions and properties are also stated. Previous contributions to the Volterra series identification problem are summarized, with a comprehensive explanation of the regularized estimation method as extended from linear identification literature.

3.1 Introduction

For linear time-invariant systems, the nonparametric impulse response model gives a complete and intuitive description of the system space. As a consequence, its truncated FIR form is the model structure of choice in many modern regularized estimation procedures, which were discussed in Chapter 2. When broadening our scope to *nonlinear* time-invariant systems, an analogous nonparametric representation is known as the Volterra series [109].

In much the same way that linear functions are just one term in the Taylor series description of nonlinear static functions, the linear impulse response becomes one term in a series of multi-dimensional impulse responses, which together form the Volterra series description of nonlinear dynamical systems. Each successive term in the series contains one additional dimension in its impulse response, and represents a single ‘nonlinear order’ of the system. For example, the next term after the linear impulse is a 2-dimensional impulse response which describes all 2nd order (quadratic) nonlinear behaviour in the system. In theory, the Volterra series has both infinite terms and infinite memory lengths within each term. The series is typically defined in the time domain, since this is the domain of an impulse response, however there are also equivalent frequency domain representations [109], [64], [65] which can be obtained using appropriate Fourier transformations. Furthermore, all representations of the Volterra series can be described using continuous- or discrete-time models.

In practice, for data-driven modeling and control, the Volterra series must be truncated such that there is a finite maximum nonlinear order, and each impulse response term has finite memory length. It has been shown in [16] that a truncated, finite memory Volterra series model can approximate any fading memory nonlinear system arbitrarily well. The fading memory condition is somewhat similar to a stability condition in linear systems, requiring that the current system output not be influenced by input signals applied in the remote past. This will also exclude nonlinear system classes which exhibit, for example, hysteresis or non-unique steady-states. The fading memory condition still leaves an incredibly broad set of systems which can be universally approximated, along with many systems which can be approximated locally within a small region of some stable equilibrium point.

The extreme versatility of Volterra series models in representing nonlinear systems will motivate the use of such models in this thesis. In particular, the identification of nonlinear systems can be performed with very little prior knowledge, requiring user selections only for the maximum nonlinear order and memory lengths. Due to the multidimensional nature of the series, however, the model requires large numbers of parameters to be estimated, and this number grows rapidly with memory length and series order. From an identification perspective, large numbers of parameters means high variance and MSE in the resulting estimates, particularly when the

input/output data length is low or measurement noise is significant. There are also implications on the computational and memory requirements when considering a large parameter space. Such complications have historically restricted the use of classical estimators on Volterra series, and only in the most recent decades have new techniques emerged which enable more widespread use of the series in data-driven modeling. Some of these techniques will be discussed in Sections 3.5 and 3.6.

3.2 Discrete-time representations

While the continuous-time Volterra series is a useful concept in systems theory, the nature of sampled data and nonparametric estimation dictates that we use discrete-time models for estimation and control. Thus, we will restrict our focus in this thesis to the discrete-time definitions of Volterra series and its related expansions. Essentially, the only difference between these definitions and their continuous-time equivalents found in e.g. [109] is the use of summations in place of continuous integrals. Naturally, the series terms also change from being continuously defined, to being defined only at integer values of their input arguments. The time and frequency domain representations are defined below.

3.2.1 Time domain representation

The Volterra series is most often expressed in the time domain, where its discrete-time representation can be given as,

$$y^0(t) = h_0 + \sum_{m=1}^M y_m(t), \quad (3.1)$$

$$y_m(t) = \sum_{\tau_1=0}^{n_m-1} \dots \sum_{\tau_m=0}^{n_m-1} h_m(\tau_1, \dots, \tau_m) \prod_{\tau=\tau_1}^{\tau_m} u(t - \tau), \quad (3.2)$$

where $u(t)$ and $y^0(t)$ denote the true system input and output respectively, $h_0 \in \mathbb{R}$ is a constant offset, and $y_m(t)$ is the output contribution from the m^{th} nonlinear order, up to some maximum series order, M . These output contributions are formed via the m -dimensional convolution in (3.2), where $h_m(\tau_1, \dots, \tau_m)$ is the m^{th} term of the Volterra series, τ_i is the i^{th} lag variable for the term, and n_m is its memory length.

The series terms, h_m , are commonly referred to as *Volterra kernels*, where the m^{th} kernel is an m -dimensional impulse response. It is these kernels which must be estimated for a Volterra series model. It should be noted that the 1st order contribution, $y_1(t)$, is nothing more than the FIR model from (2.8), and the corresponding Volterra kernel $h_1(\tau_1)$ behaves like a linear impulse response.

A visual depiction of the time domain series is provided in Figure 3.1 for the first four terms, showing the increase in dimension and complexity with each successive

kernel.

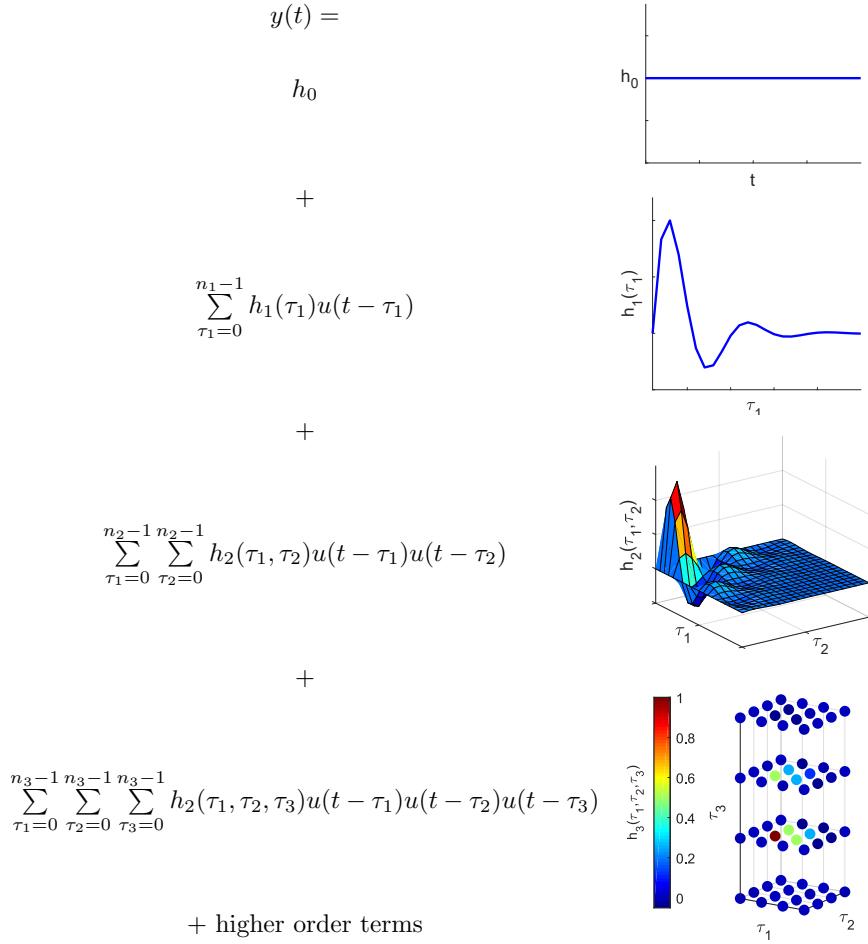


Figure 3.1: A visual example of the Volterra series for the first four terms

3.2.2 Frequency domain representations

The Volterra series can also be expressed in the frequency domain, with a number of competing representations available in the literature [26]. In their discrete-time form, all of these representations require an assumption that the system is in steady state. The assumption implies that an N -periodic input signal has been applied for a sufficiently long time so that the output is also N -periodic and free of transients. Then, for some N -sample measurement window, we can express the system input and output spectra, $U(k)$ and $Y^0(k)$ for $k = 0, 1, \dots, N - 1$, as the Discrete Fourier Transform (DFT) of their corresponding N -sample time domain signals $u(t)$ and $y^0(t)$. The relationship between these two steady state spectra will constitute the frequency domain model.

While each frequency domain structure has its own unique benefits, the most

natural extension from the time domain series is a Generalized Frequency Response Function (GFRF) model. First derived in [46], the GFRF model is obtained via multidimensional Fourier transforms on each time domain series term. In discrete-time, the mathematical description is given by,

$$\begin{aligned} Y^0(k) &= H_0(k) + \sum_{m=1}^M Y_m(k), \\ Y_m(k) &= \frac{1}{N^{m-1}} \sum_{k_1+\dots+k_m=k} H_m(k_1, \dots, k_m) \prod_{i=1}^m U(k_i), \end{aligned} \quad (3.3)$$

where $H_0(k) = h_0\delta(k)$ is a zero frequency component from the constant offset h_0 , $H_m(k_1, \dots, k_m)$ is the m^{th} order GFRF term, and k_i are discrete frequency variables. The GFRFs, H_m , are related to their equivalent m^{th} order time domain kernel, h_m , via multidimensional DFTs, i.e.

$$H_m(k_1, \dots, k_m) = \sum_{\tau_1=0}^{n_m-1} \dots \sum_{\tau_m=0}^{n_m-1} h_m(\tau_1, \dots, \tau_m) e^{-j2\pi k_1 \tau_1} \dots e^{-j2\pi k_m \tau_m}. \quad (3.4)$$

Much like the time domain Volterra series, each successive term in the GFRF series has one extra frequency dimension. Continuing the analogy, the first order contribution, $Y_1(k)$, is equivalent to a linear Frequency Response Function (FRF) model, and the corresponding GFRF, $H_1(k_1)$, exhibits the same characteristics as a FRF.

Frequency domain models are often used for abstract analysis and understanding of a system's behaviour. One disadvantage of the GFRF representation is that such analysis is complex and unintuitive when using (3.3) and multidimensional frequency functions [8]. Indeed, the GFRFs cannot even be efficiently visualized past the second or third nonlinear order. For this reason, alternate representations have been developed which provide greater insight and ease of analysis, such as the Nonlinear Output Frequency Response Functions (NOFRFs) [64], Output Frequency Response Functions (OFRFs) [65] and Associated Frequency Response Functions (AFRFs) [41], [42]. This thesis in particular will consider the NOFRF model structure, which is an alternate frequency domain extension of the Volterra series containing a one-dimensional frequency function at every nonlinear order. The NOFRF at any given order can then be viewed and analyzed in a similar fashion to a linear FRF.

Mathematically, the NOFRF model expresses the output spectra as,

$$Y_0(k) = H_0(k) + \sum_{m=1}^M Y_m(k) \quad (3.5)$$

$$Y_m(k) = G_m(k)U_m(k) \quad (3.6)$$

where $G_m(k)$ is the m^{th} order NOFRF at frequency bin k , and $U_m(k)$ is defined as the DFT of the N -sample input raised to the m^{th} power, i.e. $u^m(t)$. An equivalent block

structure for the model is shown in Figure 3.2. For a given input sequence, there is a mathematical relationship between the GFRF and NOFRF at each nonlinear order [26], which can be given as

$$G_m(k) = \frac{\sum_{k_1+\dots+k_m=k} H_m(k_1, \dots, k_m) \prod_{i=1}^m U(k_i)}{\sum_{k_1+\dots+k_m=k} \prod_{i=1}^m U(k_i)}. \quad (3.7)$$

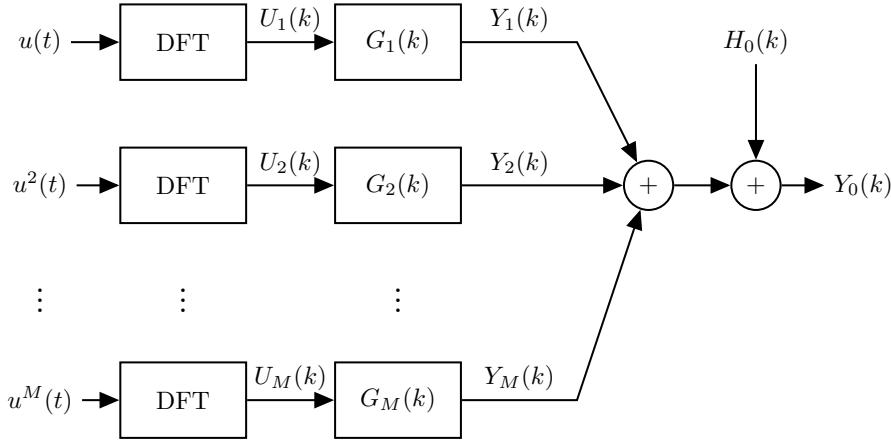


Figure 3.2: Equivalent block structure of the NOFRF model

As can be seen in (3.7), the NOFRF series is an input dependent model unlike its GFRF counterpart. The dependence arises because the NOFRFs act on exponents of the input rather than the input directly, and this property limits their utility in system analysis. Nevertheless, in Chapter 7 we will examine a special case where the NOFRFs are input independent and pseudo-linear.

3.3 Properties and assumptions

Some important properties and assumptions for Volterra series models will be established in this section and considered true in the sequel. The first of these properties concerns causality.

Definition 3.1 (Causal Volterra kernel). *A Volterra kernel $h_m(\tau_1, \dots, \tau_m)$ is said to be causal if it satisfies*

$$h_m(\tau_1, \dots, \tau_m) = 0 \quad \forall \tau_i < 0, \quad i = 1, \dots, m. \quad (3.8)$$

The implication of causality is that the output contribution produced by a causal kernel cannot be influenced by future input samples.

Another issue which requires careful consideration is the uniqueness of Volterra series models.

Property 3.2. *In general, a Volterra kernel $h_m(\tau_1, \dots, \tau_m)$ is non-unique for $m > 1$, i.e. there are other kernels which will produce the same input to output mapping.*

While non-uniqueness is certainly undesirable from an identification standpoint, there is a special subset of kernels which are unique within their set. These are the symmetric Volterra kernels.

Definition 3.3 (Symmetry). *A symmetric kernel $h_m(\tau_1, \dots, \tau_m)$ is one in which the value of the kernel is independent of the ordering of the lag variables, i.e.*

$$h_m(\tau_1, \tau_2, \dots, \tau_m) = h_m(\tau_2, \tau_1, \dots, \tau_m) = \dots = h_m(\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_m}) \quad (3.9)$$

$$i_j \neq i_k, \quad i_1, i_2, \dots, i_m \in (1, 2, \dots, m)$$

A symmetric kernel is unique in the sense that there exists no other *symmetric* kernel that exhibits the same input to output mapping. For this reason, assuming symmetry has become the common convention when dealing with Volterra series models [26]. Even in the case where an asymmetric kernel is provided, a symmetric kernel can be easily generated [109].

Property 3.4. *Any non-unique asymmetric Volterra kernel $h_m^*(\tau_1, \dots, \tau_m)$ can be used to generate a unique symmetric equivalent kernel, h_m , by averaging all kernels obtained from the set, S_m , of all possible permutations of $\boldsymbol{\tau} = [\tau_1, \dots, \tau_m]$, i.e.*

$$h_m(\tau_1, \dots, \tau_m) = \frac{1}{m!} \sum_{\boldsymbol{\tau} \in S_m} h_m^*(\boldsymbol{\tau}). \quad (3.10)$$

Symmetric kernels not only solve the uniqueness issue for Volterra series, they also reduce the number of parameters to be estimated in a given kernel. Since the order of the lags is not important, we need only identify the parameter once for each combination of lag values, instead of each permutation. As a result, the expression for the number of *unique* parameters requiring estimation in an m^{th} order symmetric kernel with memory length n_m is given by,

$$d_m = \binom{n_m + m - 1}{m} = \frac{\prod_{i=0}^{m-1} (n_m + i)}{m!}. \quad (3.11)$$

While symmetry does provide a reduction in the number of estimated parameters, the series is still plagued by the ‘curse of dimensionality’, a phrase used to describe the rapid increase in parameter space as problems are moved to higher dimensions. Indeed, (3.11) shows the number of parameters will increase exponentially with nonlinear order (dimension) m , and combinatorially with memory length n_m . This parameter growth is visualized in Figure 3.3.

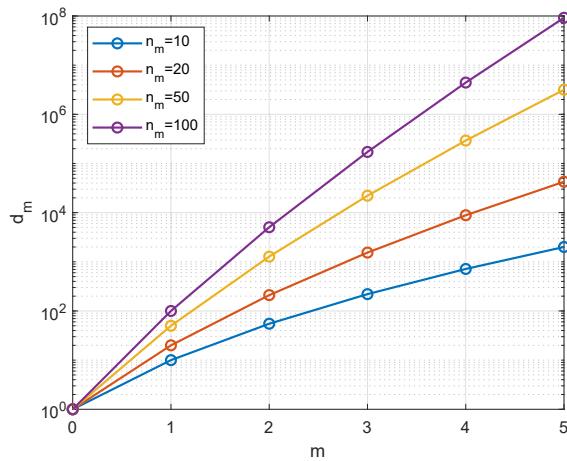


Figure 3.3: Number of unique kernel parameters for varying dimension and memory length

The final consideration which must be made is stability. In the case of linear systems, bounded-input bounded-output (BIBO) stability can be defined easily using the impulse response representation. In particular for discrete-time models, a necessary and sufficient condition is that the impulse response be absolutely summable:

$$\sum_{\tau=-\infty}^{\infty} |h(\tau)| < \infty. \quad (3.12)$$

For Volterra series models, the situation is similar but not quite as straightforward.

Property 3.5. *For a Volterra kernel $h_m(\tau_1, \dots, \tau_m)$, the kernel is BIBO stable if*

$$\sum_{\tau_1=-\infty}^{\infty} \dots \sum_{\tau_m=-\infty}^{\infty} |h_m(\tau_1, \dots, \tau_m)| < \infty. \quad (3.13)$$

The condition that the kernel be absolutely summable is sufficient to give BIBO stability, however it is no longer a necessary condition for $m > 1$. Furthermore, a method for constructing stable kernels which violate this condition can be found in [109]. While such special cases exist, in practice one would expect the kernels to decay to zero in all directions of increasing lag, implying that we have absolute summability.

Having established some important properties of the Volterra series and its associated kernels, the following assumption will hold true for the remainder of the thesis.

Assumption 3.6. *All nonlinear systems considered in this thesis can be represented by Volterra series models which contain causal, symmetric, and absolutely summable kernels.*

3.4 Relationship to common block structures

While block-oriented models are not the focus of this thesis, some of the more common block structures can be related to an equivalent Volterra series representation in an explicit sense. These relationships have been explored in e.g. [134] and [60], and will be used in the sequel to generate simulated Volterra systems with which to test the proposed algorithms. First, some assumptions are placed on the static nonlinear and linear dynamic blocks.

Assumption 3.7. *In all block structures, the static nonlinear block can be represented by a polynomial nonlinearity¹,*

$$F(x(t)) = \beta_0 + \beta_1[x(t)] + \dots + \beta_M[x(t)]^M.$$

Assumption 3.8. *In all block structures, the linear blocks are stable rational filters which can be represented by their corresponding impulse response, $g_i(\tau)$ for some integer subscript i .*

With these assumptions in place, three of the most well known block-oriented models are introduced, and their relationship to the Volterra series representation is defined.

3.4.1 Wiener model

The Wiener block structure consists of a single linear filter followed by a static nonlinear block to generate the output. A block diagram is provided in Figure 3.4.

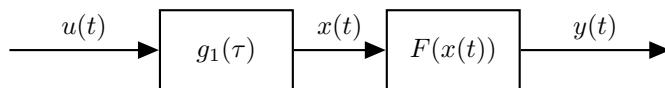


Figure 3.4: Block structure for a Wiener model

For the Wiener model, equivalent Volterra kernels are generated as follows,

$$\begin{aligned} h_0 &= \beta_0, \\ h_m(\tau_1, \dots, \tau_m) &= \beta_m \cdot g_1(\tau_1) \cdot \dots \cdot g_1(\tau_m). \end{aligned} \quad (3.14)$$

3.4.2 Hammerstein model

The Hammerstein model contains the same two blocks as the Wiener model, but in the opposite order, i.e. a static nonlinear block followed by a linear filter. The structure is represented in Figure 3.5.

¹The polynomial assumption on $F(x(t))$ is not particularly restrictive, since a Taylor series expansion can easily be generated for non-polynomial nonlinearities.

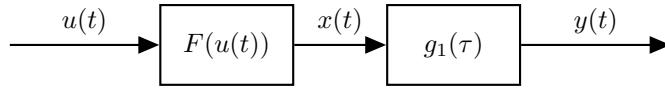


Figure 3.5: Block structure for a Hammerstein model

Hammerstein systems can also be represented using a series of Volterra kernels, i.e.

$$h_0 = \beta_0,$$

$$h_m(\tau_1, \dots, \tau_m) = \begin{cases} \beta_m g_1(\tau_1) & \tau_1 = \dots = \tau_m, \\ 0 & \text{otherwise.} \end{cases} \quad (3.15)$$

The kernels are seen to be nonzero only on their diagonal elements.

3.4.3 Wiener-Hammerstein model

The Wiener-Hammerstein model has the structure shown in Figure 3.6, which is seen to be a combination of the previous two models. In such systems, the static nonlinear block is both preceded and followed by a linear filter. Clearly, Wiener-Hammerstein models are capable of describing a broader range of systems than Wiener and Hammerstein models, since the latter two are just special cases of the former.

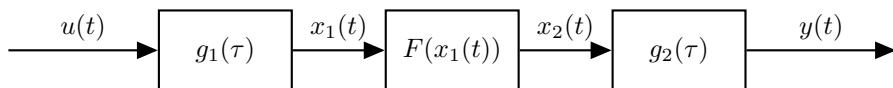


Figure 3.6: Block structure for a Wiener-Hammerstein model

The Volterra kernels corresponding to the Wiener-Hammerstein model are found via

$$h_0 = \beta_0,$$

$$h_m(\tau_1, \dots, \tau_m) = \beta_m \sum_{\tau=-\infty}^{\infty} g_2(\tau) \cdot g_1(\tau_1 - \tau) \cdot \dots \cdot g_1(\tau_m - \tau). \quad (3.16)$$

3.5 History of Volterra series identification

Despite the curse of dimensionality, the earliest results on Volterra series identification attempted to use classical methods from statistics involving input-output cross-correlations. Such methods were made more feasible by the development of

the Wiener series [108], which is a variation of the Volterra series that is orthogonal (in discrete-time) for Gaussian white noise input. Estimation of the Wiener series could then be performed individually for each Wiener kernel by computing cross-correlations from experimental data [68].

Of course, the assumption of a Gaussian white noise input is too strict to be useful in many practical settings, and significant effort was put towards broadening the applicability of the cross-correlation approach. An extension for coloured (correlated) Gaussian inputs was proposed in [61], and extensions to other input distributions can be found in [89] and [116]. Despite these efforts, estimation using cross-correlations was not a very efficient tool, requiring very large sample sizes for proper convergence, even in small, low-order problems [44].

Apart from using inputs with a known distribution, researchers specifically designed the identification input to maximize the accuracy of their chosen estimator. Some examples of these inputs include pseudorandom binary sequences (PRBS) in [126] and [102], and pseudorandom multilevel sequences (PRMLS) in [88]. These techniques are only applicable if the input used for identification can be precisely controlled, which is not always a realistic assumption.

More recently, the strategy of expanding Volterra series models in terms of a finite set of basis functions has come to the forefront. Such methods can provide a dramatic reduction in the number of parameters requiring estimation, providing the basis functions are chosen appropriately. An early example can be found in [76], where Laguerre basis function expansions are employed for nonlinear biological systems. The popularity of the approach gained momentum following papers on the optimal Laguerre [19] and Kautz [30] basis function expansions for Volterra models. As well as the classical sets of orthonormal functions, wavelets have also been explored as a basis due to their desirable properties of orthogonality and compact support [26]. Some notable examples can be found in [86], [98] and [99], however these methods are limited to kernels of order three and lower.

The rapid increase of available computing power in recent decades has allowed more computationally intensive methods to enter the field of Volterra series estimation. A method based on particle swarm optimization was proposed in [21], while a relationship between the series and neural networks was discussed in [137] and used to propose an identification scheme. Regularized linear regression was explored in [44] using ridge regression. Finally, the Bayesian regularization approach [11] of interest in this thesis is made possible via intensive global optimization techniques.

The discussion so far has only focussed on time domain estimation, since frequency domain kernels can be obtained indirectly from transformation of the time domain terms. Direct frequency domain estimation has its own history, however, which runs in parallel and follows the same trends as the time domain. Estimation of a system's GFRFs was originally considered for specific and strict experimental conditions. Methods in [5] required harmonic or Gaussian noise inputs, while [58]

required special harmonic inputs as well as readily available difference equations for the system. Many results employed specially designed periodic multisine inputs, such as [17], [27] and [39]. More recently, an interpolation method was provided in [85] which exploits the (assumed) local smoothness of GFRFs, and another method has been proposed in [70] which directly uses harmonic time domain data for estimation.

3.6 Regularized Volterra series estimation

Inspired by the success of Bayesian regularization techniques in linear impulse response identification [94], an extension of the method to the Volterra series was first proposed in [12], and further improved in [11] and [10]. The approach is conceptually similar, since each Volterra kernel is a multidimensional impulse response, however the extra dimensions add complexity when designing prior covariances for the kernels. While the properties of smoothness and decay are still integral to the process, they must now be imposed across the entire (hyper)surface of each kernel.

First, assumptions must be placed on the model structure for the system.

Assumption 3.9. *The system can be represented by a Volterra series model with known deterministic input and white Gaussian measurement noise, $e(t) \sim \mathcal{N}(0, \sigma^2)$, added directly at the output. The measured output, y , is then given by*

$$y(t) = y^0(t) + e(t), \quad (3.17)$$

where $y^0(t)$ is the noise-free Volterra series output from (3.1).

From the assumed model structure, a linear regression formulation of the problem can now be considered, i.e.

$$Y = \Phi^T \theta + E, \quad (3.18)$$

where $Y, E \in \mathbb{R}^{N-n+1}$ are the vectors of output measurements and measurement noise respectively, and $\theta = [h_0, h_1^T, h_2^T, \dots, h_M^T]^T \in \mathbb{R}^D$ is the parameter vector containing all unique coefficients for each Volterra kernel. For the multi-dimensional kernels, $h_m^V \in \mathbb{R}^{d_m}$ refers to a vector containing the unique kernel coefficients in $h_m(\tau_1, \dots, \tau_m)$. The regressor matrix, $\Phi \in \mathbb{R}^{D \times (N-n+1)}$, should be constructed based on the vectorization scheme used in θ , and must take into account the assumed symmetry of kernels in the series. While each kernel's coefficients are free to be ordered in an arbitrary fashion, a structured approach to vectorization is detailed in [11] for the second order case. A small illustrative example will be given here.

Consider a 2nd order Volterra series model, with memory lengths $n_1 = n_2 = n = 3$ for the kernels. The parameters requiring estimation in this model are depicted in Figure 3.7, where black circles indicate unique coefficients to be estimated, and red circles indicate non-unique coefficients which will be estimated. The gray dashed

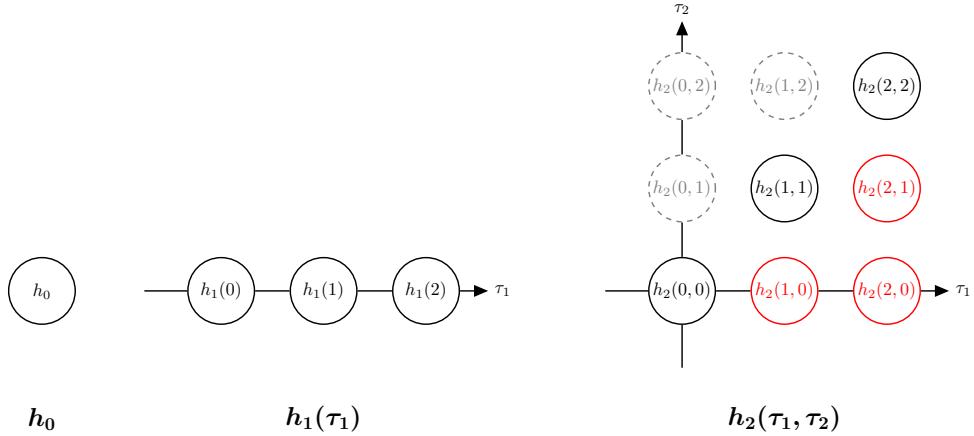


Figure 3.7: Estimation parameters in a 2nd order Volterra series with memory length 3

circles give the non-unique coefficients which will not be estimated, but rather inferred using their symmetrical red counterparts. The resulting regression structure is then given by,

$$\begin{bmatrix} y(2) \\ y(3) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ u(2) & u(3) & \dots & u(N-1) \\ u(1) & u(2) & \dots & u(N-2) \\ u(0) & u(1) & \dots & u(N-3) \\ u(2) \cdot u(2) & u(3) \cdot u(3) & \dots & u(N-1) \cdot u(N-1) \\ 2u(1) \cdot u(2) & 2u(2) \cdot u(3) & \dots & 2u(N-2) \cdot u(N-1) \\ 2u(0) \cdot u(2) & 2u(1) \cdot u(3) & \dots & 2u(N-3) \cdot u(N-1) \\ u(1) \cdot u(1) & u(2) \cdot u(2) & \dots & u(N-2) \cdot u(N-2) \\ 2u(0) \cdot u(1) & 2u(1) \cdot u(2) & \dots & 2u(N-3) \cdot u(N-2) \\ u(0) \cdot u(0) & u(1) \cdot u(1) & \dots & u(N-3) \cdot u(N-3) \end{bmatrix}^T \begin{bmatrix} h_0 \\ h_1(0) \\ h_1(1) \\ h_1(2) \\ h_2(0,0) \\ h_2(1,0) \\ h_2(2,0) \\ h_2(1,1) \\ h_2(2,1) \\ h_2(2,2) \end{bmatrix} + E.$$

Note the non-unique coefficients (highlighted in red) in the parameter vector, which have their regressor entries multiplied by 2 since they represent the contribution of 2 symmetric coefficients in the second order kernel.

Given the model structure assumption in (3.17), it was shown in Chapter 2 that both the least squares and maximum likelihood estimate of θ will be given by the analytic solution,

$$\hat{\theta}_{LS} = (\Phi\Phi^T)^{-1}\Phi Y. \quad (3.19)$$

Since the least squares estimation problem is likely to be ill-conditioned or even underdetermined for a Volterra model with large numbers of parameters, we regularize the estimate with a Bayesian-tuned quadratic penalty. The Bayesian method proposed in [11] treats each Volterra kernel in the model as an independent Gaussian

process with a separate prior covariance, outlined in the following assumption.

Assumption 3.10 (Gaussian priors). *Every vector of kernel coefficients, $h_m^{\mathcal{V}}$, is a zero mean Gaussian process:*

$$h_m^{\mathcal{V}} \sim \mathcal{N}(0, P_m) \quad m = 1, \dots, M. \quad (3.20)$$

Furthermore, all pairs of coefficient vectors, $h_i^{\mathcal{V}}$ and $h_j^{\mathcal{V}}$, are independent for $i \neq j$.

The constant offset is also Gaussian, i.e. $h_0 \sim \mathcal{N}(0, P_0)$, and is independent of all other kernels.

This Bayesian assumption frames the total parameter vector as a Gaussian process, i.e. $\theta \sim \mathcal{N}(0, P)$, where P is the total prior covariance constructed as a block diagonal matrix,

$$P = \begin{bmatrix} P_0 & & & 0 \\ & P_1 & & \\ 0 & & \ddots & \\ & & & P_M \end{bmatrix}, \quad (3.21)$$

with $P_m \in \mathbb{R}^{d_m \times d_m}$ being the prior covariance for the m^{th} Volterra kernel.

Individually, the prior covariance matrices should be designed to impose smoothness and exponential decay on their multi-dimensional kernels, as in the FIR case. For P_1 , the TC structure in (2.37) or DC structure in (2.38) can be directly applied, since h_1 is effectively an impulse response. For kernel dimensions greater than one, the TC or DC structure must be incorporated into a more complex covariance matrix in order to impose smoothness and stability along the entire (hyper)surface. The approach suggested in [11] is to consider m perpendicular regularizing directions for the kernel h_m , where the vector $(1, \dots, 1)$ is always included as one of the directions. As an example, the regularizing directions for a second order kernel are shown in Figure 3.8. A rotated coordinate system can be formed from the set of regularizing vectors, which we will denote (v_m^1, \dots, v_m^m) .

Remark 3.11. *A suitable set of orthogonal regularizing directions for the m^{th} non-linear order can be obtained via the following standard form (for $m \geq 3$):*

$$\begin{aligned} v_m^1 &= (1, 1, \dots, 1), \\ v_m^2 &= (-1, \dots, -1, m-1) \\ v_m^3 &= (-1, \dots, -1, m-2, 0) \\ &\dots \\ v_m^m &= (-1, 1, 0, \dots, 0) \end{aligned} \quad (3.22)$$

The rotated coordinate system is used to apply the chosen covariance structure along each regularizing direction, forming a set of m partial covariance matrices.

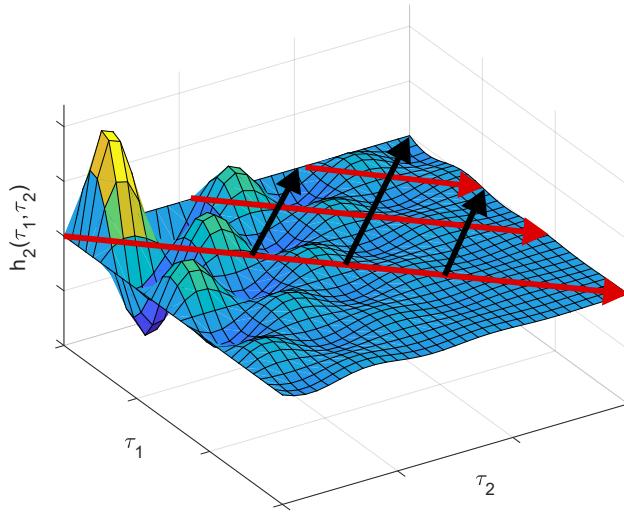


Figure 3.8: The two perpendicular regularizing directions, (1,1) (red) and (-1,1) (black), for a second order Volterra kernel

For the regularizing direction v_m^j , the corresponding partial covariance matrices for the TC and DC structures are,

$$\text{TC structure:} \quad P_m^j(x, y) = (\lambda_m^j)^{\max(x', y')}, \quad (3.23)$$

$$\text{DC structure:} \quad P_m^j(x, y) = (\lambda_m^j)^{(x'+y')/2} (\rho_m^j)^{|x'-y'|}, \quad (3.24)$$

where x' is the coordinate of $h_m^V(x)$ on the v_m^j axis. If $h_m^V(x)$ is associated with lag values $\boldsymbol{\tau} = (\tau_1, \dots, \tau_m)$, then $x' = \langle \boldsymbol{\tau}, v_m^j \rangle$. An Hadamard product of the partial covariances is required to produce the total prior covariance for the kernel, i.e.

$$P_m = c_m \cdot P_m^1 \circ \dots \circ P_m^m, \quad m = 1, \dots, M. \quad (3.25)$$

where c_m is a normalization hyperparameter. The covariance for h_0 is simply given by

$$P_0 = c_0. \quad (3.26)$$

There are clearly a large number of hyperparameters which require empirical tuning to form our regularization penalty, P . More specifically, there are $m+1$ hyperparameters per kernel when using a TC structure, which include the normalization constant, c_m , and m decay constants, λ_m^j . Using the DC structure increases this to $2m+1$ hyperparameters per kernel, by including the ρ_m^j set. In either case, the suggested method for hyperparameter tuning in [11] is to use the standard marginal likelihood maximization of (2.43). While the optimization is still applicable in the Volterra series case, the large dimension of the search space and non-convexity of the problem can yield prohibitively long computation times, with a high risk of settling in sub-optimal local minima. These complications motivate the research in Chapter 4 of this thesis.

Following the hyperparameter tuning (which may also include noise variance σ^2), P can be constructed, and the regularized estimate of θ may be obtained analytically from (2.41).

As an illustration of the profound benefit of Bayesian regularization in Volterra series estimation, parameter estimates have been obtained for a simple 2nd order example system, whose true kernels are shown in Figure 3.9. Estimates of these kernels were obtained using

1. A simple least squares estimator (3.19)
2. A regularized estimate (2.41) with empirically tuned TC structure for P via (3.21), (3.23) and (2.43)

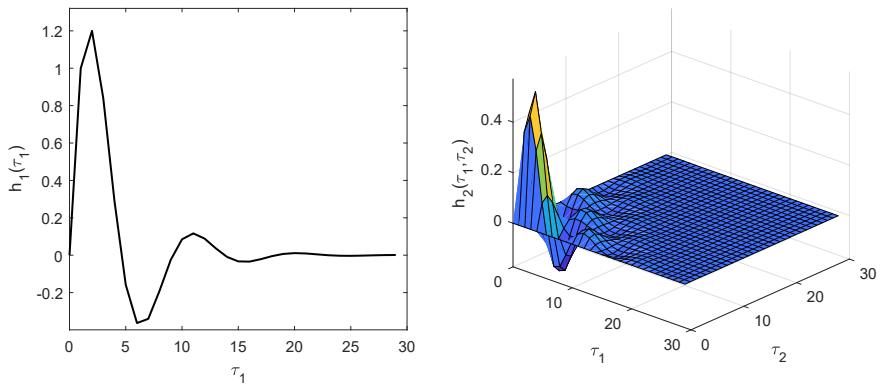


Figure 3.9: True Volterra kernels for the comparative simulation example

The first set of estimates, shown in Figure 3.10, were obtained from a simulation of the system in Figure 3.9 for a dataset with length $N = 1000$, noise variance $\sigma^2 = 1$, and a white Gaussian noise input. In this case, the input signal is sufficiently exciting, however the least squares estimate still suffers from high variance due to the large number of parameters being estimated. The regularized estimator, however, produces smooth, accurate estimates of the Volterra kernels due to the construction of prior covariances which impose the desired behaviour.

The second set of estimates, shown in Figure 3.11, were obtained using the same specifications, except that the input is now *filtered* Gaussian noise. In this case the input does not have energy in the full frequency band (i.e. it is not sufficiently exciting), hence the least squares estimator is ill-conditioned and the resulting estimate is unusable. The regularized estimate, on the other hand, is relatively unaffected by the change.

3.7 Summary

The Volterra series model for nonlinear dynamical systems was introduced in this chapter, with discrete-time representations defined in both the time and frequency

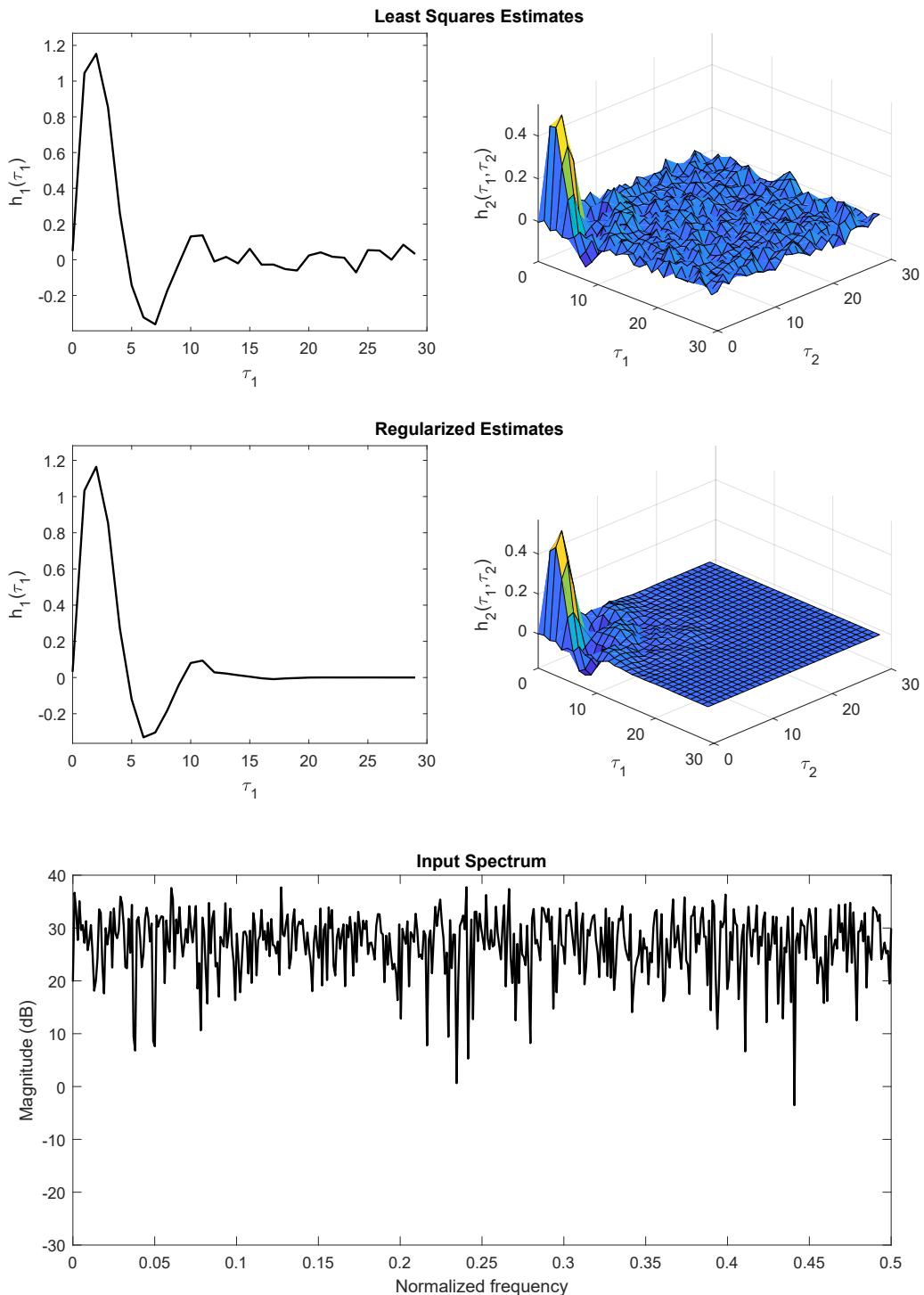


Figure 3.10: Least squares (top) and regularized (middle) kernel estimates for a white Gaussian input (bottom)

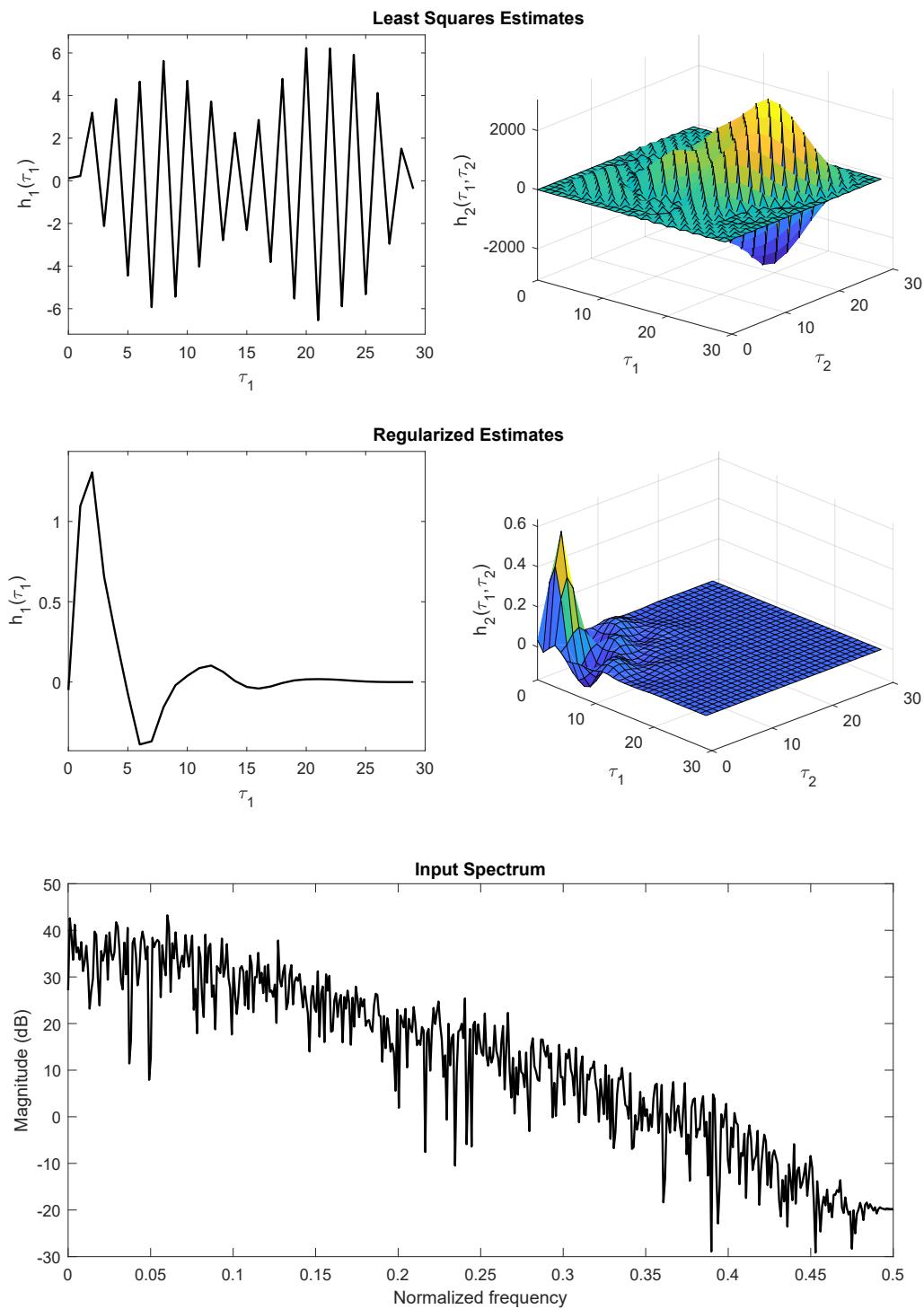


Figure 3.11: Least squares (top) and regularized (middle) kernel estimates for a filtered Gaussian input (bottom)

domain. Assumptions of causality, stability and symmetry were established for each Volterra kernel in the series, and the parameter growth issue (i.e. the curse of dimensionality) was discussed. Previous results on the identification of Volterra series models were summarized, and particular attention was paid to the Bayesian regularization method proposed recently in the literature. The regularized approach was developed for time domain identification, and will serve as a foundation for the extensions proposed in Part II of this thesis, as well as inspiration for many of the frequency domain contributions in Part III.

PART

II

Time Domain Identification

CHAPTER

4

EM-Based Hyperparameter Optimization for Regularized Volterra Series Estimation

The Bayesian regularization method for estimating Volterra series models suffers from poor computation time scaling with respect to the series order. The scaling problem arises in the marginal likelihood maximization used for hyperparameter tuning, where the dimension of the search space grows quadratically with series order. A further problem also arises in the fact that the cost function can have a large number of local minima. In this chapter, the hyperparameter optimization is reformulated using an expectation-maximization (EM) approach, which reduces the single large optimization problem to a set of smaller problems in an iterative scheme. The maximum search space dimension of these smaller problems increases linearly with the order of the Volterra series, thereby improving the computation time scaling of regularized estimation. Both the traditional marginal likelihood and proposed EM approach are applied to numerical examples to confirm the superior computation time scaling of the EM method.

4.1 Introduction

The Volterra series provides a linear-in-the-parameters representation for a broad class of nonlinear systems, however we have seen in Chapter 3.6 that the estimation of its associated Volterra kernels is ill-suited to a standard least squares framework. This is because the nonparametric nature of the model implies a requirement for large numbers of parameters, which in turn produces large variances in least squares estimates obtained from a finite and noisy data record.

In order to reduce the variance of kernel estimates, a regularized least squares approach was developed in [11] and [10]. As discussed in Chapter 3, the method uses Bayesian techniques to impose smoothness and decay along the (hyper)surfaces of each Volterra kernel, which is a nonlinear extension of Bayesian impulse response identification as introduced in [94]. The regularized estimation employs a set of tunable hyperparameters to describe the smoothness and decay rates of each kernel, where traditionally such hyperparameters have been optimized through a maximization of the associated marginal likelihood expression, which is typically non-convex [22]. The dimension of the search space for this optimization increases quadratically with the order of the Volterra series model, since decay hyperparameters are required for each dimension of each kernel.

When compared to standard least squares and other more traditional estimation methods, the main disadvantages of the regularization approach in [11] are:

- Increased computational cost,
- Poor computation time scaling with the maximum series order, M ,
- Risk of converging to a local minimum in the hyperparameter optimization

These disadvantages are directly acknowledged in the concluding remarks of [11], which states:

“Several issues should be addressed in the future, such as efficient ways to optimize the hyperparameters of the regularization matrix in order to reduce the risk of resulting in a local minimum of the marginal likelihood.”

To address the issues of computation time scaling and non-convexity, an alternative hyperparameter tuning method is developed in this chapter. The new method is proposed within the expectation-maximization (EM) framework [32], and is inspired by results from [15] and [14] which deal with linear system identification. When applied to Volterra series estimation, the approach will be shown to trade the single large optimization problem in [11] for a set of smaller problems in an iterative scheme, where the maximum dimension of the search space increases linearly with the series order.

Numerical examples in Section 4.4 confirm that computation times for the EM-based formulation do not increase as rapidly with series order, while still converging to the same hyperparameters as the marginal likelihood optimization traditionally used in these problems.

4.2 The expectation-maximization algorithm

‘Expectation-maximization’ refers to a family of computation methods designed to find maximum likelihood estimates in cases where there are latent (unobserved) variables [82]. These latent variables may represent real unobserved data, or (as is the case in this thesis) they can be ‘nuisance parameters’ resulting from the reformulation of an estimation problem.

In the general case, consider a data vector, X , which is a concatenation of two subvectors:

$$X = \begin{bmatrix} Y \\ Z \end{bmatrix}, \quad (4.1)$$

where Y is the vector of *observed* data, while Z represents the *unobserved* variables. Assume further that X is drawn from some distribution, $p_c(Y, Z; \eta)$, which is described by the unknown parameters, η . Under these conditions, the maximum likelihood estimation of η is the problem of interest, i.e.

$$\hat{\eta} = \arg \max_{\eta} L_c(Y, Z; \eta), \quad (4.2)$$

where $L_c(Y, Z; \eta) = \log p_c(Y, Z; \eta)$ is the log likelihood function for the complete data vector, X . However since the Z portion of data is not observed, (4.2) cannot be used directly.

One possible solution is to produce a marginal likelihood function by integrating out the latent variables from our distribution, yielding a marginal likelihood maximization (MLM) problem:

$$\hat{\eta} = \arg \max_{\eta} \log \int_Z p_c(Y, Z; \eta) dZ \quad (4.3)$$

$$= \arg \max_{\eta} \log p(Y; \eta). \quad (4.4)$$

Indeed, this is the typical pathway for regularized estimation problems employing empirical Bayes techniques [94], [11], [67].

There are many circumstances where a marginal likelihood approach becomes undesirable. For example, the marginalization process may be impossible or computationally intractable to perform, or it may be that the resultant optimization problem (4.4) is extremely difficult to solve. The latter problem is relevant to regularized Volterra series estimation, where the marginal likelihood approach implies searching through a high-dimensional non-convex function. In these situations, EM

methods provide an alternate pathway which often resolves the issues associated with marginalization.

In essence, the EM approach moves back to the complete data problem, (4.2), but replaces the unobserved data, Z , with its expected value given Y and some guess of η . While there is no reason to believe this guess is accurate, the procedure can be repeated iteratively with nice convergence properties for η .

Conceptually, the iterative procedure for the EM method consists of two steps. The first is the *expectation* step, where the complete data likelihood function is approximated using parameters, $\hat{\eta}^{(k)}$, from the previous iteration, i.e.

$$Q(\eta, \hat{\eta}^{(k)}) = \mathbf{E}\{L_c(Y, Z; \eta)\}, \quad (4.5)$$

where Q is the likelihood approximation, and the expectation is taken with respect to the posterior distribution $p(Z|Y; \hat{\eta}^{(k)})$.

The second step consists of a *maximization*, where the parameters are updated by maximizing the function (4.5) calculated in the first step. The new hyperparameters are thus given by,

$$\hat{\eta}^{(k+1)} = \arg \max_{\eta} Q(\eta, \hat{\eta}^{(k)}). \quad (4.6)$$

The chain of successive parameter estimates, $\hat{\eta}^{(k)}$, is called an *EM sequence*, and there are some useful properties of this sequence which can be exploited. Noting first that $Q(\eta, \hat{\eta}^{(k)})$ is a lower bound on the marginal likelihood, $\log p(Y; \eta)$, it can be shown that each new iteration of the EM sequence corresponds to a higher point in the marginal likelihood function. If we impose the reasonable assumption that the marginal likelihood is itself upper bounded, then it is straightforward to show that the EM sequence will converge to a stationary point of $p(Y; \eta)$ [138]. While some artificial examples have been constructed in the literature which converge to saddle points or local minima [82], a local maximum is almost guaranteed in practice. Thus, the EM method provides an alternate route to MLM without the need for explicit marginalization.

There is another property of the EM method which is not supported by formal results, but has been noted anecdotally. In comparison to standard MLM, the equivalent EM approach is often found to be more robust to poor initialization, and therefore less likely to settle in a sub-optimal local maximum. Such behaviour is particularly important in the Volterra series context, where MLM can involve many local maxima in the high-dimensional search space [11].

4.3 Hyperparameter optimization using EM

For the Bayesian estimation problem outlined in Chapter 3.6, the parameter vector, θ , is framed as a random variable which is jointly distributed with the output data, Y . The problem of hyperparameter optimization can thus be framed as likelihood

maximization with latent variables. The original parameter vector, θ , becomes the unobserved data, and we require an estimate of all hyperparameters η describing the joint distribution, i.e.

$$\hat{\eta} = \arg \max_{\eta} L_c(Y, \theta; \eta), \quad (4.7)$$

where $L_c(Y, \theta; \eta)$ is the joint log likelihood of Y and θ .

This interpretation has already led to the MLM approach defined in (2.43), both for the linear FIR and Volterra series cases. In this section, the EM solution will be derived for hyperparameter tuning in Volterra kernels with Gaussian priors. Relevant modeling assumptions will be (re)stated below for convenience.

Assumption 4.1. *The data generating model is a Volterra system with white Gaussian additive noise (3.17), and uses the vectorized regression structure given in (3.18), i.e.*

$$Y = \Phi^T \theta + E, \quad (4.8)$$

with $\theta = [h_0, h_1^T, h_2^T, \dots, h_M^T]^T$.

Assumption 4.2. *The parameter vector is a Gaussian random variable, $\theta \sim \mathcal{N}(0, P)$, with block diagonal prior covariance (3.21). Individual kernel covariances are defined using the TC structure, via (3.23) and (3.25).*

While the derivation is performed here for the TC covariance structure, the DC structure and others can be derived in an almost identical fashion.

4.3.1 The expectation step

Under assumptions 4.1 and 4.2, the lower bound function $Q(\eta, \hat{\eta}^{(k)})$ in (4.5) can be constructed analytically from the normal distributions of Y and θ . First, the function is separated into two components,

$$Q(\eta, \hat{\eta}^{(k)}) = \mathbf{E}\{\log p(Y|\theta; \eta)\} + \mathbf{E}\{\log p(\theta; \eta)\}. \quad (4.9)$$

It can be seen from (4.8) that $Y|\theta; \eta \sim \mathcal{N}(\Phi^T \theta, \sigma^2 I)$, hence,

$$\begin{aligned} \mathbf{E}\{\log p(Y|\theta; \eta)\} &= \mathbf{E}\left\{a_1 - \frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (Y^T Y - 2Y^T \Phi^T \theta + \theta^T \Phi \Phi^T \theta)\right\} \\ &= a_1 - \frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (Y^T Y - 2Y^T \Phi^T \mathbf{E}\{\theta\} + \text{Tr}[\Phi \Phi^T \mathbf{E}\{\theta \theta^T\}]), \end{aligned} \quad (4.10)$$

where a_1 is a real number, and $\sigma^2 \in \eta$ is the only hyperparameter present in the expression. The remaining expected values in (4.10) are taken with respect to $p(\theta|Y; \hat{\eta}^{(k)})$, with analytic expressions obtained via the joint distribution of θ and

Y ,

$$\begin{bmatrix} \theta \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} P & P\Phi \\ \Phi^T P & \Phi^T P\Phi + \sigma^2 I \end{bmatrix}\right). \quad (4.11)$$

Using the standard expressions for Gaussian conditional distributions, and applying the matrix inversion lemma, we can arrive at numerically robust expressions for the desired expected values,

$$\mathbf{E}\{\theta\} = \mathbf{E}\{\theta|Y\} = P(\hat{\eta}^{(k)})[\Phi\Phi^T P(\hat{\eta}^{(k)}) + [\hat{\sigma}^2]^{(k)}I]^{-1}\Phi Y, \quad (4.12)$$

$$\mathbf{Cov}\{\theta|Y\} = [\hat{\sigma}^2]^{(k)}P(\hat{\eta}^{(k)})[\Phi\Phi^T P(\hat{\eta}^{(k)}) + [\hat{\sigma}^2]^{(k)}I]^{-1}, \quad (4.13)$$

$$\mathbf{E}\{\theta\theta^T\} = \mathbf{Cov}\{\theta|Y\} + \mathbf{E}\{\theta|Y\} \cdot \mathbf{E}\{\theta|Y\}^T, \quad (4.14)$$

where P and σ^2 are constructed from the hyperparameters of the previous iteration, $\hat{\eta}^{(k)}$.

Returning to (4.9), $\mathbf{E}\{\log p(\theta; \eta)\}$ can also be computed, given that θ is normally distributed by assumption in the Bayesian setting. The term can thus be written as,

$$\begin{aligned} \mathbf{E}\{\log p(\theta; \eta)\} &= \mathbf{E}\{a_2 - \frac{1}{2}\log \det P(\eta) - \frac{1}{2}\theta^T P^{-1}(\eta)\theta\} \\ &= a_2 - \frac{1}{2}\log \det P(\eta) - \frac{1}{2}\text{Tr}[P^{-1}(\eta)\mathbf{E}\{\theta\theta^T\}], \end{aligned} \quad (4.15)$$

where $\mathbf{E}\{\theta\theta^T\}$ can be evaluated using (4.14), and a_2 is a real number independent of η .

The block diagonal structure of P , given in (3.21), can now be exploited to split (4.15) into smaller components. As discussed earlier, we will consider TC-based prior covariances, however similar steps can be applied to any block diagonal structure. In order to partition the hyperparameters further, we require some new notation.

Notation 4.3. *The vector of decay hyperparameters for kernel h_m is denoted*

$$\bar{\eta}_m = [\lambda_m^1, \dots, \lambda_m^m] \quad m = 1, \dots, M. \quad (4.16)$$

Notation 4.4. *The kernel prior covariance P_m can be expressed as*

$$P_m = c_m \cdot \bar{P}_m(\bar{\eta}_m) \quad m = 1, \dots, M. \quad (4.17)$$

For the 0th order kernel, h_0 , we have $\bar{P}_0 = 1$

Notation 4.5. *The expected value, $\mathbf{E}\{\theta\theta^T\}$, is a matrix which can be decomposed as*

$$\mathbf{E}\{\theta\theta^T\} = \begin{bmatrix} W_0 & & * \\ & W_1 & \\ * & & \ddots \\ & & & W_M \end{bmatrix}, \quad (4.18)$$

where $W_m \in \mathbb{R}^{d_m \times d_m}$ is a square matrix with the same dimension as P_m , and $*$ represents arbitrary off diagonal components.

The two terms in (4.15) involving $P(\eta)$ can now be expressed in terms of summations over each nonlinear order. The resulting sums have the desirable property that the m^{th} term depends only on hyperparameters c_m and $\bar{\eta}_m$:

Proposition 4.6. *Under assumptions 4.1 and 4.2, the following two equivalences exist:*

$$\log \det P(\eta) = \sum_{m=0}^M (d_m \log c_m + \log \det \bar{P}_m(\bar{\eta}_m)), \quad (4.19)$$

$$\text{Tr}[P^{-1}(\eta) \mathbf{E}\{\theta\theta^T\}] = \sum_{m=0}^M \frac{1}{c_m} \text{Tr}[\bar{P}_m^{-1}(\bar{\eta}_m) W_m]. \quad (4.20)$$

From here, (4.15) can be rewritten as,

$$\mathbf{E}\{\log p(\theta; \eta)\} = a_2 - \frac{1}{2} \left[\sum_{m=0}^M \left(d_m \log c_m + \log \det \bar{P}_m(\bar{\eta}_m) + \frac{1}{c_m} \text{Tr}[\bar{P}_m^{-1}(\bar{\eta}_m) W_m] \right) \right]. \quad (4.21)$$

4.3.2 The maximization step

In the expectation step, an analytic expression was developed for $Q(\eta, \hat{\eta}^{(k)})$ by splitting the function into the two additive components, (4.10) and (4.15), where the latter term was also subdivided into a summation of terms from each nonlinear order in (4.21). It should be noted that each additive term in $Q(\eta, \hat{\eta}^{(k)})$ has no hyperparameter overlap with the other terms. This allows the maximization step to be performed individually on each term in the lower bound, so that each optimization has limited complexity and search space dimension.

Considering (4.10), we find that it depends only on $\sigma^2 \in \eta$, and is maximized by the choice,

$$[\sigma^2]^{(k+1)} = \frac{Y^T Y - 2Y^T \Phi^T \mathbf{E}\{\theta\} + \text{Tr}[\Phi \Phi^T \mathbf{E}\{\theta\theta^T\}]}{N}. \quad (4.22)$$

Next, considering the summation in (4.21), each term in the sum can be maximized individually, leading to the following set of minimization problems:

$$[\hat{c}_m^{(k+1)} \hat{\eta}_m^{(k+1)}] = \arg \min_{c_m, \bar{\eta}_m} (d_m \log c_m + \log \det \bar{P}_m(\bar{\eta}_m) + \frac{1}{c_m} \text{Tr}[\bar{P}_m^{-1}(\bar{\eta}_m) W_m]). \quad (4.23)$$

Optimizing (4.23) with respect to c_m and back-substituting allows the normalization constant to be separated from the optimization of the remaining hyperparameters,

$\bar{\eta}_m$, such that,

$$\hat{\eta}_m^{(k+1)} = \arg \min_{\bar{\eta}_m} \left(d_m \log \text{Tr}[\bar{P}_m^{-1}(\bar{\eta}_m) W_m] + \log \det \bar{P}_m(\bar{\eta}_m) \right), \quad m = 1, \dots, M, \quad (4.24)$$

and

$$\hat{c}_m^{(k+1)} = \frac{\text{Tr}[\bar{P}_m^{-1}(\hat{\eta}_m^{(k+1)}) W_m]}{d_m}, \quad m = 0, \dots, M. \quad (4.25)$$

The hyperparameter updates in (4.22) and (4.25) are analytic, while the decay hyperparameters still require non-convex optimization in (4.24). Nevertheless, each nonlinear order is now considered separately, and the dimension of the search space for a given order is limited to the dimension of the associated Volterra kernel.

4.3.3 Algorithm for implementation

We have now derived the expectation and maximization steps for Volterra series hyperparameter tuning, thus a practical algorithm can be developed. The full iterative procedure for hyperparameter optimization is given in Algorithm 1, which replaces the single global optimization (2.43) used in [11].

Algorithm 1 EM algorithm for hyperparameter tuning in Volterra series estimation

Require: Y, Φ , initial hyperparameter guess $\hat{\eta}^{(0)}$, and convergence tolerance ϵ

```

1:  $k \leftarrow 0$ 
2: while  $\max(|\hat{\eta}^{(k)} - \hat{\eta}^{(k-1)}|/|\hat{\eta}^{(k-1)}|) > \epsilon$  do
3:   Form  $P, \mathbf{E}\{\theta\}$  and  $\mathbf{E}\{\theta\theta^T\}$  using  $\hat{\eta}^{(k)}$ 
4:   for all  $m \in 0, 1, \dots, M$  do
5:     Extract  $\bar{P}_m$  from  $P$  and  $W_m$  from  $\mathbf{E}\{\theta\theta^T\}$ 
6:     if  $m > 0$  then
7:       Compute  $\hat{\eta}_m^{(k+1)}$  using (4.24)
8:     end if
9:     Compute  $\hat{c}_m^{(k+1)}$  using (4.25)
10:    end for
11:    Compute  $[\sigma^2]^{(k+1)}$  using (4.22)
12:    Form full hyperparameter vector  $\hat{\eta}^{(k+1)}$ 
13:     $k \leftarrow k + 1$ 
14: end while
15: return  $\hat{\eta}^{(k)}$ 

```

4.3.4 Computation time scaling

Hyperparameter tuning is by far the largest contributor to computational burden in regularized Volterra series estimation. In particular, when the maximum series order, M , increases, the dimension of the hyperparameter space grows rapidly. Looking in particular at the TC and DC-based covariance structures from (3.23) and (3.24)

respectively, the total number of hyperparameters in the problem can be expressed as,

$$n_h = \underbrace{\frac{1}{\sigma^2}}_{\sigma^2} + \sum_{m=0}^M \left(\underbrace{n_d m}_{\bar{\eta}_m} + \underbrace{1}_{c_m} \right) \quad (4.26)$$

$$= \frac{n_d}{2} M^2 + \left(\frac{n_d}{2} + 1 \right) M + 2, \quad (4.27)$$

where $n_d = 1$ for TC and $n_d = 2$ for DC-based structure. It can be seen from (4.27) that the number of hyperparameters is growing quadratically, $\mathcal{O}(M^2)$.

If the traditional MLM approach (2.43) is used, there will be a single non-convex optimization over the entire space of dimension n_h . The dimensionality scales quadratically, $\mathcal{O}(M^2)$, and while it is difficult to quantify the computation time for a non-convex optimization routine, it is suffice to say that scaling with dimension is rapid.

Now considering the EM method outlined in Algorithm 1, any given iteration has analytic updates for σ^2 and each c_m , as well as separate non-convex optimizations for each $\bar{\eta}_m$. The analytic updates have negligible computation time, while all the optimization problems are independent and may be computed in parallel, such that we can consider only the largest problem, $\bar{\eta}_M$, involving $n_d M$ decay hyperparameters. The dimension scaling is now linear, $\mathcal{O}(M)$, with maximum series order. Of course, this optimization is performed iteratively until convergence, however the improved computation time scaling will still be evident compared to MLM when M becomes large enough.

4.4 Numerical examples

To evaluate the computation time benefits of the proposed EM-based tuning over a traditional global MLM, both methods are applied to simulated Volterra systems of increasing order in a set of Monte Carlo studies.

4.4.1 Simulation Settings

The test systems have a Wiener-type structure given by,

$$y(t) = \sum_{m=1}^M \left(\frac{N_m(q)}{A(q)} u(t) \right)^m + e(t), \quad (4.28)$$

where q^{-1} is the backwards shift operator, and $e(t) \sim \mathcal{N}(0, \sigma^2)$ is Gaussian output noise added in each realization. Each term in the sum defines an m^{th} order Volterra kernel (see Chapter 3.4), and the constant offset term is neglected.

All kernels in every system are given the same denominator dynamics, i.e.

$$A(q) = 1 - 0.5q^{-1} + 0.3q^{-2}. \quad (4.29)$$

Systems from 1st to 5th order ($M = 1, 2, 3, 4, 5$) are generated such that each kernel contributes equally to the output. The numerator polynomials for each system are taken from,

$$\begin{aligned} N_1(q) &= q^{-1} & N_2(q) &= 0.75q^{-1} \\ N_3(q) &= 0.6q^{-1} & N_4(q) &= 0.52q^{-1} \\ N_5(q) &= 0.46q^{-1} \end{aligned}$$

The input to the systems is Gaussian with unit variance, and the output noise variance, σ^2 , is chosen to give a Signal to Noise Ratio (SNR) of 20dB. The chosen data length is

$$N = 3 \sum_{m=1}^M d_m,$$

i.e. $3 \times$ the number of parameters to be estimated.

Using both MLM and EM tuning, the regularized least squares method is used to estimate system kernels of memory lengths $n_m = 7, 9$ and 12 . The three memory lengths are chosen using three corresponding thresholds, 3%, 1% and 0.1%, on the decay of the system kernels with respect to their maximum value. Figure 4.1 shows the three truncation lengths (blue circles) as applied to the true first order kernel, $h_1(\tau_1)$. Note that the memory lengths must be sufficiently low in order to maintain computational feasibility at high kernel orders.

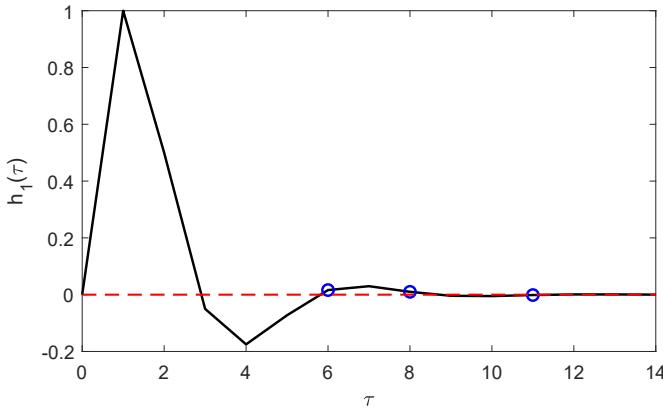


Figure 4.1: The true first order kernel h_1 (black line) in the simulation example, and the three applied truncation lengths (blue circles)

For both tuning methods, hyperparameters are initialized according to,

$$[\hat{\sigma}^2]^{(0)} = 1.5\sigma^2, \quad \hat{c}_m^{(0)} = 1, \quad [\hat{\lambda}_m^j]^{(0)} = 0.8 \quad \forall m, j$$

All non-convex optimizations are performed using `fmincon` in MATLAB, with a step tolerance of 10^{-3} . Convergence of the iterative EM method is defined by a change of less than 1% in all hyperparameters. Computation times are measured from the beginning of hyperparameter tuning to convergence, using a PC with Intel Xeon 3.50 GHz processor. In order to observe mean behaviour, each Monte Carlo study considers 10 realizations of each system.

4.4.2 Results

As well as computation time measurements, a validation error metric is calculated for each estimated system by applying a Gaussian validation input of length 20,000 to the estimated and true systems. Labeling the true output as y_{val} and estimated output as y_{est} , we construct a normalized RMS error metric defined by,

$$E_{NRMS} = \frac{\text{rms}(y_{val} - y_{est})}{\text{rms}(y_{val})}. \quad (4.30)$$

Table 4.1: Mean computation times and validation errors for $n_m = 7$

M	Mean Comp. Time (s)		Mean E _{NRMS}	
	MLM	EM	MLM	EM
1	0.064	0.173	0.249	0.224
2	0.132	0.389	0.189	0.187
3	2.848	2.114	0.176	0.161
4	40.10	23.13	0.133	0.135
5	501.9	217.7	0.117	0.114

Table 4.2: Mean computation times and validation errors for $n_m = 9$

M	Mean Comp. Time (s)		Mean E _{NRMS}	
	MLM	EM	MLM	EM
1	0.071	0.269	0.218	0.203
2	0.229	0.778	0.148	0.148
3	8.704	7.761	0.123	0.120
4	282.6	155.3	0.089	0.101
5	3962	1523	0.071	0.085

The mean computation time results shown in Table 4.1 for $n_m = 7$, Table 4.2 for $n_m = 9$, and Table 4.3 for $n_m = 12$, demonstrate that a dramatic improvement in computation time scaling is achieved using the proposed EM method for hyperparameter optimization. For the given experimental conditions, estimation time using

Table 4.3: Mean computation times and validation errors for $n_m = 12$

M	Mean Comp. Time (s)		Mean ENRMS	
	MLM	EM	MLM	EM
1	0.047	0.135	0.170	0.187
2	0.476	1.387	0.116	0.120
3	30.99	46.75	0.079	0.073
4	2799	1211	0.059	0.060
5	80000	15720	0.040	0.062

EM is comparable to MLM tuning at 3rd order, and significantly faster for 4th and higher order systems. The mean validation errors are also provided in the tables, confirming that there is no degradation in model quality. This is not surprising, since the hyperparameters were observed to converge to the same locations for both methods. Figure 4.2 shows the EM convergence pathway for all hyperparameters in an example 2nd order ($M = 2$) realization.

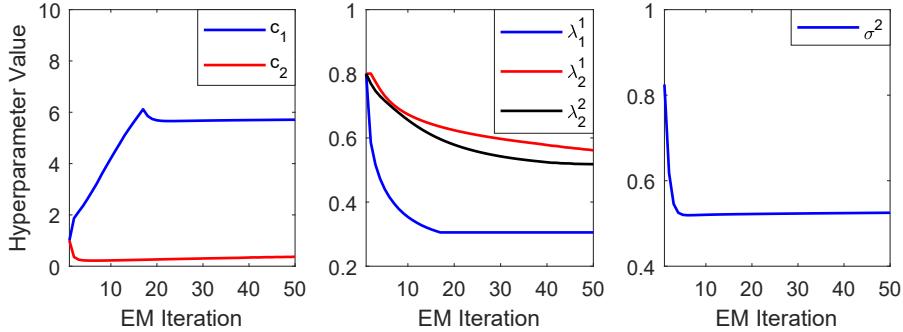


Figure 4.2: EM convergence pathway for all hyperparameters in a 2nd order example

To visualize the computation time trend, we label the M^{th} order mean computation times as $t_{EM}(M)$ and $t_{MLM}(M)$ for the EM and MLM methods respectively, and define the ratio function,

$$R(M) = \frac{t_{EM}(M)}{t_{MLM}(M)}. \quad (4.31)$$

$R(M)$ is plotted for all memory lengths in Figure 4.3. Despite a significant difference in the number of parameters being estimated in each case, the relationship between $R(M)$ and M remains similar, with a crossover point to EM superiority, in terms of computation time, at $M = 3$ or 4, depending on the memory length. Note that as n_m increases, $R(3)$ increases while $R(M \geq 4)$ decreases. The trend predicts increased relative benefit of EM over MLM tuning if M and n_m are increased further.

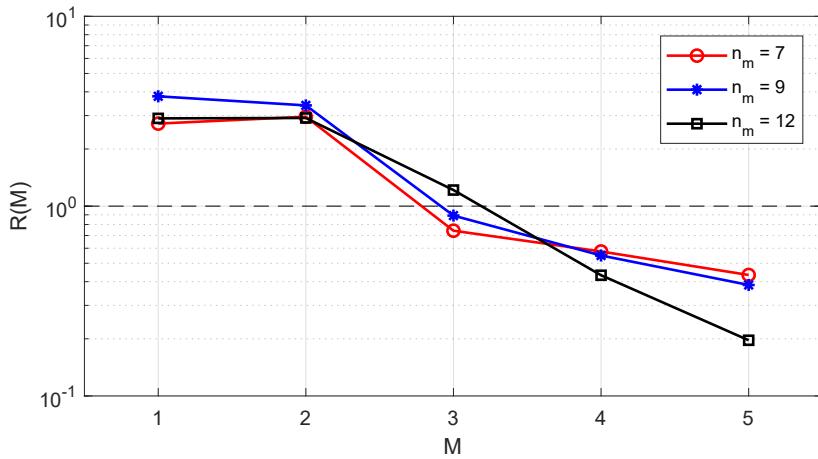


Figure 4.3: Ratios of computation times for varying M and n_m

4.5 Conclusion

Bayesian regularization of Volterra kernel estimates necessitates the optimization of a large number of hyperparameters. When optimizing through a single global marginal likelihood maximization, the computational burden scales rapidly with the order of the Volterra series. In this chapter, the optimization problem was reformulated within an expectation-maximization framework. The new formulation splits the optimization into many small problems, some of which are analytic in nature. The remaining non-convex optimization problems have significantly reduced dimension and do not scale as rapidly with the order of the series.

Evaluating the MLM and proposed EM-based tuning methods in a Monte-Carlo study revealed improved computation time scaling for the proposed method, with computation time for 4th and 5th order system estimates being significantly lower overall. In the studied example, validation errors obtained using each method's hyperparameters were very similar, and the EM hyperparameters were confirmed to be converging to a maximum of the marginal likelihood function.

CHAPTER



5

Volterra Kernel Identification using Regularized Basis Function Expansions

This chapter addresses the feasibility of high order Volterra series estimation from noisy data records, by developing an extension to the Bayesian regularization method which can be applied directly to basis function expansions of the Volterra model. The number of parameters requiring estimation in a basis function expanded model is typically much smaller than for the equivalent time domain series, provided that the basis functions are chosen appropriately for the system. In this chapter, theoretical justification is provided for the proposed approach. The EM algorithm designed in Chapter 4 is then extended to allow the optimization of both the regularization hyperparameters and the basis functions at each nonlinear order of the series. Monte Carlo simulation studies highlight the benefits of a regularized basis function approach over the original time domain method.

5.1 Introduction

The contributions in this part of the thesis are focussed on the challenge of time domain Volterra series identification, and in particular, increasing the feasibility of parameter estimation for high series orders under general experimental conditions. In Chapter 4, this issue was addressed from the perspective of computation time scaling, where the main optimization was reformulated into a method whose computational burden scales more favourably with series order.

There is another important factor affecting the feasibility of high order estimation, which is the sheer number of parameters required to be estimated in a time domain Volterra model. While the problem is particularly severe in multi-dimensional Volterra kernels, the desire to perform parameter reduction exists also in the linear impulse response case, from which inspiration can be drawn. In particular, orthonormal basis function modeling is a technique traditionally applied to linear systems [53] that has also exhibited significant potential in the nonlinear domain [26].

The use of orthonormal basis functions for linear system identification has been rigorously studied for several decades [53], [131], [132]. The basic goal is to formulate a compact representation of a linear time-invariant (LTI) system by describing it as a weighted sum of rational orthonormal basis functions. If the poles of the basis functions are chosen wisely, then only a small number of functions are required to sufficiently model the system behaviour. Estimating the coefficients of the expansion is a linear-in-the-parameters problem, and the compact structure can produce lower variance estimates at the price of a small bias. Basis function representations are also possible in a Volterra series context [106], however, placement of the basis function poles is more difficult to optimize in multi-dimensional kernels than in the linear case [19], [30].

This chapter makes a novel proposal to combine the parameter reduction technique of basis function modeling with the EM-based regularization method developed in Chapter 4. By applying regularization directly on the more compact basis function model structure, the variance of model estimates can be dramatically reduced, along with their associated computation time. Theoretical justification is given for the use of regularization on higher-dimensional basis function expansions, and the EM method for hyperparameter tuning is modified to also iteratively optimize the basis function poles at each series order. A Monte Carlo simulation study is performed on example Volterra systems ranging from 2nd to 5th order, to compare the feasibility of high order parameter estimation using the proposed basis function method against the original regularized method in [11].

5.2 Orthonormal basis functions in identification

The motivation for using orthonormal basis functions in linear identification stems directly from the shortcomings of FIR identification [53]. If we consider the impulse response, $h(\tau)$, generated by the rational transfer function, $G(z)$, we observe that if the poles of $G(z)$ are close to the unit circle, then $h(\tau)$ will tend to 0 very slowly with increasing τ . This implies that a large number of FIR parameters are required to accurately model the system. If we instead model the system $G(z)$ as a weighted sum of transfer functions $F_i(z)$ with impulse responses $f_i(\tau)$,

$$G(z) = \sum_{i=1}^{\mathcal{B}} \alpha_i F_i(z) \quad (5.1)$$

$$\text{and } h(\tau) = \sum_{i=1}^{\mathcal{B}} \alpha_i f_i(\tau), \quad (5.2)$$

with the poles of each $F_i(z)$ chosen carefully, the number of coefficients required to model $G(z)$ can be significantly reduced [53].

When choosing a set of basis functions $F_i(z)$, it is desirable that the functions form a complete basis for the subspace of strictly proper, real, rational transfer functions. One class of basis functions which satisfy this criterion are the Takenaka-Malmquist functions [1]. The functions are constructed from a sequence of all-pass filters, whose poles, a_i , lie within the unit disc and satisfy the completeness condition,

$$\sum_{i=1}^{\infty} (1 - |a_i|) = \infty.$$

Their general form [53] can be written as

$$\begin{aligned} F_i(z) &= \frac{\sqrt{1 - |a_i|^2}}{z - a_i} \prod_{j=1}^{i-1} \frac{1 - \bar{a}_j z}{z - a_j} \\ &= \frac{\sqrt{1 - |a_i|^2}}{z - a_i} \prod_{j=1}^{i-1} G_j(z). \end{aligned} \quad (5.3)$$

The so-called Generalized Orthonormal Basis Functions (GOBFs) are obtained in the case where $G_j(z) = \bar{G}(z) \ \forall j$. The McMillan degree of $\bar{G}(z)$ can also be made greater than 1, by taking a linear combination of the complex functions generated in (5.3) [87]. It follows that the total number of generating poles required to define a GOBF sequence is equal to the McMillan degree of $\bar{G}(z)$, which we will denote d_G . In the following sections we will examine more closely the case of $d_G = 1$ and $d_G = 2$.

5.2.1 Laguerre basis functions

Laguerre Basis Functions (LBFs) are a subset of the GOBFs which are parameterized by a single, real pole, such that $d_G = 1$ for the all-pass filter $\bar{G}(z)$. This results in basis functions of the form

$$F_i(z) = \frac{\sqrt{1 - |a|^2}}{z - a} \left(\frac{1 - az}{z - a} \right)^{i-1}, \quad a \in (-1, 1). \quad (5.4)$$

While the Laguerre functions are easy to construct with a simple structure, one distinct disadvantage is that the functions cannot be constructed using complex poles. As a consequence, a larger number of expansion coefficients are required to accurately model oscillatory systems [131].

5.2.2 Kautz basis functions

In the case where $d_G = 2$, we can obtain the 2-parameter Kautz Basis Functions (KBFs) [132]. For these functions, the opportunity exists to incorporate a pair of complex poles into the generating all-pass filter in order to better model oscillatory responses. In much the same way that the Laguerre functions were parameterized by a single real pole in (5.4), the Kautz functions can be parameterized by two complex conjugate poles, $[x + iy, x - iy]$. The problem with this parameterization is that the allowable ranges for x and y are interdependent, since $(x^2 + y^2) < 1$ is required for stability of $\bar{G}(z)$ [53]. A more practical parameterization defines two new parameters,

$$a = \frac{2x}{1 + x^2 + y^2}, \quad b = -(x^2 + y^2),$$

such that the 2-parameter Kautz functions can be expressed in the form,

$$\begin{aligned} F_{2i-1}(z) &= \frac{\sqrt{1 - b^2}(z - a)}{z^2 + a(b - 1)z - b} \left(\frac{-bz^2 + a(b - 1)z + 1}{z^2 + a(b - 1)z - b} \right)^{i-1}, \\ F_{2i}(z) &= \frac{\sqrt{(1 - b^2)(1 - a^2)}}{z^2 + a(b - 1)z - b} \left(\frac{-bz^2 + a(b - 1)z + 1}{z^2 + a(b - 1)z - b} \right)^{i-1}, \end{aligned} \quad (5.5)$$

where $a, b \in (-1, 1)$.

5.3 Basis function expansions of Volterra models

Using orthonormal basis functions in a Volterra series model is a concept referred to as the Volterra/Wiener approach [106]. In the same way that an impulse response, $h(\tau)$, can be approximated by a sum of basis responses in (5.1), higher-dimensional Volterra kernels can also be approximated using a sum of basis functions. For a

kernel, $h_m(\tau_1, \dots, \tau_m)$, the basis function expansion can be expressed as

$$h_m(\tau_1, \dots, \tau_m) = \sum_{i_1=1}^{\mathcal{B}_m} \dots \sum_{i_m=1}^{\mathcal{B}_m} \alpha_m(i_1, \dots, i_m) \prod_{j=1}^m f_{m,i_j}(\tau_j), \quad m = 1, \dots, M, \quad (5.6)$$

where $f_{m,l}(\tau)$ is the impulse response corresponding to the l^{th} basis function of the kernel h_m , \mathcal{B}_m is the number of basis functions in the expansion, and $\alpha_m(\cdot)$ is the set of expansion coefficients. The time domain Volterra series model (3.2) can be combined with (5.6) to restructure the model, i.e.

$$\begin{aligned} y^0(t) &= \alpha_0 + \sum_{m=1}^M y_m(t), \\ y_m(t) &= \sum_{i_1=1}^{\mathcal{B}_m} \dots \sum_{i_m=1}^{\mathcal{B}_m} \alpha_m(i_1, \dots, i_m) \prod_{j=1}^m u_{m,i_j}^f(t) \end{aligned} \quad (5.7)$$

where $\alpha_0 = h_0$ from (3.1), and $u_{m,l}^f$ is the input, u , filtered by the l^{th} basis function of the m^{th} kernel's basis. Note the similarity in structure between the models in (3.2) and (5.7), which motivates the treatment of the expansion coefficient sets α_m as *basis function kernels* in the domains of their corresponding bases. Taking this view, \mathcal{B}_m is analogous to memory length, and the assumption of symmetry, as discussed in Chapter 3.3, is equally valid with the same consequences for parameter estimation. The basis function kernels can be much more compact than their time-domain counterparts, provided that the bases are carefully designed [19], [30]. Figure 5.1 shows an example 2nd order kernel in the time domain (left) and using a well parameterized LBF expansion (right).

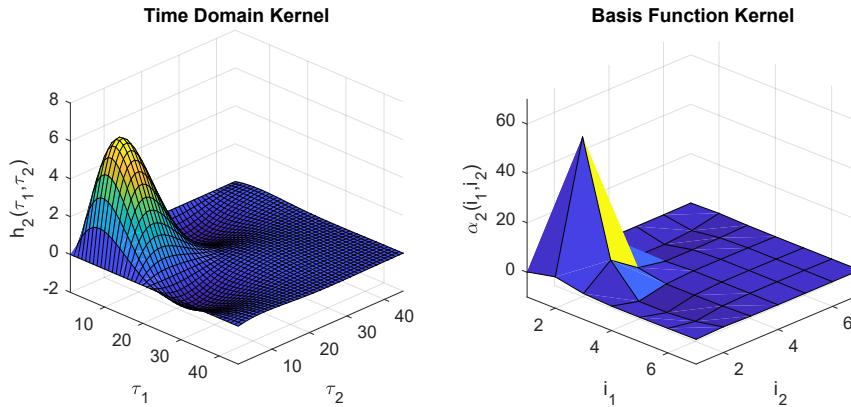


Figure 5.1: A well parameterized LBF kernel (right) is typically far more compact than its equivalent time domain kernel (left)

A linear regression formulation for the estimation of the α_m kernels is possible using the same framework described in Chapter 3.6. Based on Assumption 3.9, a

regression model for the basis function expanded series (5.7) can be given as

$$Y = \Phi_f^T \alpha + E, \quad (5.8)$$

where the regressor, Φ_f , now contains appropriate *filtered* input products. The parameter vector is still constructed as if the expansion coefficients α_m were standard Volterra kernels, i.e. $\alpha = [\alpha_0, \alpha_1^V, \dots, \alpha_M^V]^T$, where α_m^V denotes a vector of *unique* m^{th} order expansion coefficients.

Naturally a least squares estimate of the basis function kernels is possible, provided that there is enough input/output data available, which is analytically computed using

$$\hat{\alpha}_{LS} = (\Phi_f \Phi_f^T)^{-1} \Phi_f Y. \quad (5.9)$$

So far we have assumed that a set of appropriate basis functions exist for each nonlinear order m . In reality, these basis functions must be chosen by the user, and ideally the functions for a kernel h_m should reflect the system dynamics present in that kernel. The problem of selecting optimal basis functions for Volterra kernel expansion was considered in [19] for the case of LBFs, and in [30] for the case of KBFs. Optimality here is defined by the minimization of error introduced from truncation of the LBF/KBF expansions to a finite length, \mathcal{B}_m . That is to say that an optimal basis will allow the most compact basis function kernel.

5.3.1 Optimal selection of Laguerre poles

We will consider a Volterra kernel h_m of order m , which can be expressed as a Laguerre basis function expansion using (5.4) and (5.6), parameterized by the pole $a = a_m$. It can be shown that minimizing the cost function,

$$J_m(a_m) = \sum_{i_1}^{\infty} \dots \sum_{i_m}^{\infty} (i_1 + \dots + i_m) \alpha_m^2(i_1, \dots, i_m), \quad (5.10)$$

with respect to a_m will also minimize the upper bound on the squared norm of the truncation error [19]. Furthermore, there is a strict global solution to (5.10) which can be obtained solely from the time-domain kernel. The optimal pole can thus be calculated from h_m [19], i.e.

$$a_m = \frac{2Q_{1,m} - 1 - Q_{2,m}}{2Q_{1,m} - 1 + \sqrt{4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m}}}, \quad (5.11)$$

where

$$Q_{j,m} = \frac{1}{m} \sum_{l=1}^m S_{j,l},$$

$$S_{1,l} = \frac{1}{\|h_m\|^2} \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_m=0}^{\infty} \tau_l h_m^2(\tau_1, \dots, \tau_m),$$

$$S_{2,l} = \frac{1}{\|h_m\|^2} \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_m=0}^{\infty} \tau_l [\Delta_l h_m(\tau_1, \dots, \tau_m)]^2,$$

and the operators $\|\cdot\|^2$ and Δ_l are defined by

$$\|h_m\|^2 = \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_m=0}^{\infty} h_m^2(\tau_1, \dots, \tau_m),$$

$$\Delta_l h_m = h_m(\tau_1, \dots, \tau_l + 1, \dots, \tau_m) - h_m(\tau_1, \dots, \tau_l, \dots, \tau_m).$$

5.3.2 Optimal selection of Kautz poles

For the 2-parameter Kautz case, we consider again the Volterra kernel h_m , expanded now using the Kautz functions defined in (5.5). Unlike the Laguerre case, no analytic solution exists in the literature to find optimal Kautz parameterizations. However, sub-optimal estimates can be obtained by fixing $a = a_m$, as this provides an analytic solution [30] for optimizing $b = b_m$ at the given a_m value. Combining this analytic, sub-optimal method with an exhaustive search through the space of valid a_m values will then reveal the global optimum.

It can be shown [30] that the upper bound on truncation error resulting from expansion into Kautz functions is minimized by minimizing the cost function,

$$L_m(a_m, b_m) = \frac{q_2 b_m^2 - 2q_1 b_m + q_3}{1 - b_m^2}, \quad (5.12)$$

where the scalars q_1, q_2, q_3 are given by

$$\begin{aligned} q_1 &= \mu_1(g_{even}) + \mu_1(g_{odd}), \\ q_2 &= \mu_2(g_{even}) + \mu_2(g_{odd}), \\ q_3 &= q_2 + m \cdot \mu_3(g_{even}) + m \cdot \mu_3(g_{odd}). \end{aligned}$$

The moments μ_j are defined as,

$$\begin{aligned} \mu_1(g) &= \sum_{j=1}^m \left[\sum_{k_1=0}^{\infty} \dots \sum_{k_j=0}^{\infty} \dots \sum_{k_m=0}^{\infty} k_j \cdot g(k_1, \dots, k_j, \dots, k_m) \cdot g(k_1, \dots, k_j - 1, \dots, k_m) \right], \\ \mu_2(g) &= \sum_{j=1}^m \left[\sum_{k_1=0}^{\infty} \dots \sum_{k_j=0}^{\infty} \dots \sum_{k_m=0}^{\infty} k_j \cdot g^2(k_1, \dots, k_m), \right. \\ \mu_3(g) &= \left. \sum_{k_1=0}^{\infty} \dots \sum_{k_m=0}^{\infty} g^2(k_1, \dots, k_m), \right. \end{aligned}$$

and the functions g_{even} and g_{odd} are functions on the time domain kernel:

$$g_{even}(k_1, \dots, k_m) = \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_m=0}^{\infty} h_m(\tau_1, \dots, \tau_m) \cdot \prod_{j=1}^m \hat{f}_{2k_j+2}(\tau_j),$$

$$g_{odd}(k_1, \dots, k_m) = \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_m=0}^{\infty} h_m(\tau_1, \dots, \tau_m) \cdot \prod_{j=1}^m \hat{f}_{2k_j+1}(\tau_j),$$

where $\hat{f}_{(.)}$ are the Kautz function impulse responses with $b_m = 0$.

For a fixed value of a_m , (5.12) is convex, and minimized by a specific selection of b_m [30], given by

$$b_m^* = \begin{cases} \zeta - \sqrt{\zeta^2 - 1} & \text{if } \zeta > 1, \\ \zeta + \sqrt{\zeta^2 - 1} & \text{if } \zeta < -1, \end{cases}$$

where $\zeta = (q_2 + q_3)/2q_1$

Thus, by searching through a discretized space of $a_m \in (-1, 1)$, we can analytically compute the optimal b_m^* for each a_m , and the corresponding cost, $L_m(a_m, b_m^*)$. Once the discretized a_m -space has been searched, the parameter set which produced a globally minimum L_m is the optimal choice for expansion of h_m .

5.4 Regularized identification of basis function kernels

The parallels between time domain and basis function Volterra models have motivated the treatment of expansion coefficients as *basis function kernels* in the domains of their bases. A natural progression then is to consider applying regularization to these new kernels in the same way that it was applied to time domain kernels, i.e. by imposing smoothness and decay along the hypersurfaces generated by the basis function expansions.

To this end, one can define a typical quadratically regularized estimator,

$$\hat{\alpha}_{ReLS} = \arg \min_{\alpha} \|Y - \Phi_f^T \alpha\|_2^2 + \sigma^2 \alpha^T P^{-1} \alpha \quad (5.13)$$

$$= (P\Phi_f\Phi_f^T + \sigma^2 I)^{-1} P\Phi_f Y. \quad (5.14)$$

As in Chapter 3.6, the penalty matrix P can be constructed block diagonally using Bayesian prior covariances, with the following assumption.

Assumption 5.1. Every vector of expansion coefficients, α_m^V , is a zero mean Gaussian process:

$$\alpha_m^V \sim \mathcal{N}(0, P_m) \quad m = 1, \dots, M. \quad (5.15)$$

Furthermore, all pairs of coefficient vectors, α_i^V and α_j^V , are independent for $i \neq j$.

The constant offset is also Gaussian, i.e. $\alpha_0 \sim \mathcal{N}(0, P_0)$, and independent of the

other processes.

If we simply frame the basis function kernels as Volterra kernels in a different domain, then the prior covariance for each kernel can be formed via an identical procedure to the time domain case in (3.21), (3.23)/(3.24), and (3.25). Furthermore, the hyperparameters may be tuned using the same EM-based optimization derived in Chapter 4 (Algorithm 1). It should be noted, however, that for estimation using basis functions there is an added complication of choosing poles to generate the basis functions at each nonlinear order. These poles can be treated as additional hyperparameters requiring optimization, which requires some modifications to the EM tuning approach.

5.4.1 Theoretical justification

To impose the same covariance structures on expansion coefficients that have been used on standard Volterra kernel parameters, there must be justification that the basis function kernels will in fact exhibit the same smooth, decaying behaviour.

For linear FIR models, the concept of regularized basis function estimation has already been explored in [23], where the TC covariance structure was applied in regularized estimation of Laguerre coefficients. The approach was shown, in [31], to be equivalent to applying a basis-function-based covariance structure directly on the time-domain FIR parameters. In the linear case, the justification for such an approach is trivial when using Laguerre or Kautz functions. In these cases, the set of expansion coefficients for a (rational) linear system are seen to be the impulse response of the system under a given transformation, where the transformation maps stable poles to other stable poles [53], [132]. This implies that the expansion coefficients are also free to be interpreted as a stable impulse response, with all the familiar smoothness and decay properties.

For higher-dimensional Volterra kernels the linear justification is not applicable, because the definition of stability is more complex (see Chapter 3.3), and the notion of poles in some rational linear filter may not exist for an arbitrary nonlinear system. In order to solve this issue, a Bayesian approach is proposed here to show that basis function kernels decay to zero in all causal directions of the hypersurface, so long as their time domain counterparts are well behaved. The approach will be demonstrated for the case of Laguerre basis functions.

Theorem 5.2. *If $h_m(\tau_1, \dots, \tau_m)$ is an m^{th} order, zero mean Gaussian Volterra kernel with finite covariance, then any valid Laguerre basis function expansion of the kernel, $\alpha(i_1, \dots, i_m)$, will decay to 0 in all causal directions.*

Proof. Let $i_1 = x_1 s, \dots, i_m = x_m s$, where $x_1, \dots, x_m \geq 0$ and $s \rightarrow \infty$ is the limit of interest, i.e. we consider the limiting behaviour in some causal direction.

The infinite basis function expansion is given by

$$\alpha(i_1, \dots, i_m) = \alpha(x_1 s, \dots, x_m s) = \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_m=0}^{\infty} h_m(\tau_1, \dots, \tau_m) \prod_{j=1}^m f_{x_j s}(\tau_j) \quad (5.16)$$

where $f_{x_j s}$ is the impulse response for the $x_j s = i_j^{\text{th}}$ Laguerre function.

The mean of the expansion coefficients is trivially zero, since h_m is a zero mean process. The variance of the coefficients is given by

$$\begin{aligned} \text{Var}[\alpha(x_1 s, \dots, x_m s)] &= \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_m=0}^{\infty} \sum_{q_1=0}^{\infty} \dots \sum_{q_m=0}^{\infty} \\ &\quad \text{Cov}[h_m(\tau_1, \dots, \tau_m), h_m(q_1, \dots, q_m)] \cdot \prod_{j=1}^m f_{x_j s}(\tau_j) f_{x_j s}(q_j). \end{aligned} \quad (5.17)$$

The Laguerre filters (5.4) can be rewritten as

$$F_i(z) = \frac{0z^i + \sum_{k=0}^{i-1} p_{k,1}(i)a^k z^k}{z^i + \sum_{k=0}^{i-1} p_{k,2}(i)a^{i-k} z^k}, \quad a \in (-1, 1) \setminus \{0\}, \quad (5.18)$$

where $p_{m,n}$ are arbitrary polynomials in i resulting from the binomial expansion. Performing the division in (5.18) gives the form for the first i values of $f_i(\tau)$ (See Appendix 5.A for the division):

$$\begin{aligned} f_i(\tau) &= p_{l,j}(i) \cdot a^{i-\tau} \quad \text{for } 0 \leq \tau < i, \quad l, j \in \mathbb{Z}^+ \\ &\implies \lim_{i \rightarrow \infty} f_i(\tau) = 0 \quad \forall \tau \in \mathbb{Z}^+ \end{aligned}$$

Hence $s \rightarrow \infty$ causes all $f_{x_j s}(\tau_j), f_{x_j s}(q_j)$ to go to 0 in (5.17), and since all kernel covariances are finite,

$$\lim_{s \rightarrow \infty} \text{Var}[\alpha(x_1 s, \dots, x_m s)] = 0$$

□

Remark 5.3. The smoothness of basis function kernels is not addressed in Theorem 5.2, despite being imposed by most common covariance structures. However, if we use the same Bayesian approach on the first order case as an example, the correlation between two expansion coefficients separated by $\epsilon \in \mathbb{Z}^+$ is given by:

$$\mathbf{E}\{\alpha(i)\alpha(i+\epsilon)\} = \sum_{\tau=0}^{\infty} \sum_{\tau'=0}^{\infty} \mathbf{E}\{h(\tau)h(\tau')\} f_i(\tau) f_{i+\epsilon}(\tau'),$$

which will be nonzero unless $h(\tau)$ was an i.i.d. sequence. This provides the motivation to impose a level of correlation between neighbours, and this level can be tuned by the corresponding hyperparameters.

5.4.2 Modified EM for pole selection and hyperparameter optimization

It has been noted that the poles used to generate each Volterra kernel's basis functions must also be optimized to ensure the basis function kernels are as compact as possible. In [23], which examined regularized basis function estimation for the linear FIR case, the single Laguerre pole, a , was treated as an additional hyperparameter, and placed in the same marginal likelihood optimization routine (2.43) used for all other hyperparameters. In the nonlinear case, with multidimensional kernels, it has been shown that a marginal likelihood approach does not scale well with series order, and the addition of extra basis function hyperparameters to the search space would only compound this issue. Instead, the EM method developed in Chapter 4 will be extended here to incorporate the iterative optimization of basis function poles, thereby preserving the computation time benefits associated with an EM approach.

First, some notational conventions will be clarified. The method will be derived for the case where the TC covariance structure (3.23) is used, however it can be adapted to other structures with minimal effort. Collecting all the hyperparameters for the prior covariance matrix,

$$\eta_{TC} = [c_0 \ c_1 \ \lambda_1^1 \ c_2 \ \lambda_2^1 \ \lambda_2^2 \ \dots \ c_M \ \lambda_M^1 \ \dots \ \lambda_M^M],$$

and all the basis-generating parameters,

$$\eta_{BF} = \begin{cases} [a_1 \ \dots \ a_M], & \text{for LBFs} \\ [a_1 \ b_1 \ \dots \ a_M \ b_M], & \text{for KBFs} \end{cases},$$

we can define the total hyperparameter vector as $\eta = [\eta_{TC} \ \eta_{BF} \ \sigma^2]$. The quantities introduced in Notation 4.3, 4.4 and 4.5 will also remain valid here, where time domain parameters h_m and θ are replaced by the basis function model equivalents, α_m and α respectively. The number of unique kernel coefficients per order, d_m , is still computed using (3.11), where n_m is replaced by \mathcal{B}_m .

As summarized in Sections 5.3.1 and 5.3.2, optimal selection of the LBF/KBF poles requires exact knowledge of the time-domain kernels, h_m . Since the kernels are the quantities we are required to estimate, we will instead embed the pole optimization as an additional step in the recursive EM method, such that poles are computed at each iteration which are 'optimal' for the regularized kernel estimates obtained using the current hyperparameter guess. The specific implementation details for this modified EM approach are outlined in Algorithm 2, where η_{TC} and σ^2 are optimized using the same methods introduced in Chapter 4, and η_{BF} is also iteratively updated inside the loop using the relevant results from either [19] (LBFs) or [30] (KBFs).

One desirable property of traditional EM methods is their guaranteed conver-

Algorithm 2 EM-based hyperparameter tuning for basis function Volterra kernels

Require: N input samples, Y , initial hyperparameter guess $\hat{\eta}^{(0)}$, and tolerance ϵ

- 1: $k \leftarrow 0$
- 2: **while** $\max(|\hat{\eta}^{(k)} - \hat{\eta}^{(k-1)}|/|\hat{\eta}^{(k-1)}|) > \epsilon$ **do**
- 3: Form $\phi_f(\eta_{BF})$ using $\hat{\eta}_{BF}^{(k)}$
- 4: Form P , $\mathbf{E}\{\alpha\}$ and $\mathbf{E}\{\alpha\alpha^T\}$ using $\hat{\eta}^{(k)}$
- 5: **for all** $m \in 0, 1, \dots, M$ **do**
- 6: Extract \bar{P}_m from P and W_m from $\mathbf{E}\{\alpha\alpha^T\}$
- 7: **if** $m > 0$ **then**
- 8: Compute $\hat{\eta}_m^{(k+1)}$ using (4.24)
- 9: **end if**
- 10: Compute $\hat{c}_m^{(k+1)}$ using (4.25)
- 11: **end for**
- 12: Compute $[\sigma^2]^{(k+1)}$ using (4.22)
- 13: Compute $\hat{\alpha}_{ReLS}^{(k+1)} = \mathbf{E}\{\alpha\}$ using $[\hat{\eta}_{TC}^{(k+1)} [\sigma^2]^{(k+1)}]$
- 14: Convert $\hat{\alpha}_{ReLS}^{(k+1)}$ to time domain kernel estimates $\{\hat{h}_m^{(k+1)}\}_{m=0}^M$ using (5.6)
- 15: Compute ‘optimal’ basis generating parameters $\hat{\eta}_{BF}^{(k+1)}$ from $\{\hat{h}_m^{(k+1)}\}_{m=1}^M$ (see Section 5.3.1 (LBFs) or 5.3.2 (KBFs))
- 16: $k \leftarrow k + 1$
- 17: **end while**
- 18: **return** $\hat{\eta}^{(k)}$

gence to a stationary point of the marginal likelihood function, $L(Y; \eta)$. The result in Theorem 5.4 shows that this property is essentially maintained in Algorithm 2.

Theorem 5.4. *The sequence, $\hat{\eta}^{(k)}$, generated by Algorithm 2 will converge to a stationary point of the marginal likelihood function $L(Y; \eta)$, provided that*

$$\exists j \in \mathbb{Z}^+ : \forall k \geq j, \quad \hat{\eta}_{BF}^{(k)} = \hat{\eta}_{BF}^{(k+1)},$$

i.e. the basis function parameters converge before the total sequence converges.

Proof. After iteration j , we can define a new sequence $\hat{\eta}_*^{(l)}$ where $\hat{\eta}_*^{(0)} = \hat{\eta}^{(q)}$ and $l = k - j$. It is clear that the new sequence, $\hat{\eta}_*^{(l)}$, corresponds to a fixed (basis function) model structure,

$$Y = \phi_f^T(\hat{\eta}_{BF}^{(j)})\alpha + E,$$

and is therefore a standard EM sequence as in Chapter 4 (Algorithm 1). Thus, the new sequence converges to a stationary point of $L(Y; \eta)$ [138], and so must the original, $\hat{\eta}^{(k)}$. \square

5.4.3 Computational complexity and parameter reduction

In the context of regularized Volterra series estimation, the main benefit of a basis function formulation over the original time domain approach is the potential for a dramatic reduction in the number of parameters required for an accurate model.

More specifically, one would expect the length, \mathcal{B}_m , of a basis function kernel to be much smaller in general than the equivalent time domain memory length, n_m . Considering a generic m^{th} order kernel with length q , the number of unique parameters requiring estimation was given in (3.11), i.e.

$$d_m = \frac{\prod_{j=0}^{m-1} (q+j)}{m!},$$

and thus d_m grows as $\mathcal{O}(q^m)$. It is clear that even a small difference between \mathcal{B}_m and n_m can imply a large difference in the *total* number of parameters required, especially as the order, m , is increased.

A reduction in the number of parameters being estimated will also impact on the computation time (and memory requirements) for the estimator. Computation time scaling is hard to define with respect to the number of parameters, due to the non-convex optimization problems involved in Algorithm 2, however the largest matrix computations scale as $\mathcal{O}((\sum_{m=0}^M d_m)^3)$, such that smaller parameter numbers imply *much* lower computation times.

5.5 Numerical examples

A number of Monte Carlo simulation studies are conducted to evaluate the performance of the proposed regularized basis function method for Volterra kernel identification. The studies are performed on systems with disparate dynamics, to emphasize the different advantages of Kautz and Laguerre bases. The basis function methods also permit estimation of higher order kernels than their time-domain counterpart, due to the compact nature of basis function representations, and this is demonstrated via third, fourth and fifth order systems.

5.5.1 Simulation Settings

In similar fashion to Chapter 4, the test systems are given by a Wiener-type structure,

$$y(t) = \sum_{m=1}^M \left(\frac{N_m q^{-1}}{A(q)} u(t) \right)^m + e(t), \quad (5.19)$$

where q^{-1} is the backwards shift operator, and $e(t) \sim \mathcal{N}(0, \sigma^2)$ is Gaussian output noise added in each realization. Each term in the sum defines an m^{th} order Volterra kernel (see Chapter 3.4), and each kernel possesses the same denominator dynamics, with $N_m \in \mathbb{R}^+$ scaling the kernel's contribution to the output. The constant offset term is neglected.

The simulation studies differ both in the resonance of the underlying linear filter, as well as the maximum kernel order, M . Two second order ($M = 2$) systems are

tested, with their dynamics given by

$$\text{Sys2a: } A(q) = 1 - 1.8036q^{-1} + 0.8338q^{-2},$$

$$\text{Sys2b: } A(q) = 1 - 1.5q^{-1} + 0.8125q^{-2}.$$

‘Sys2a’ is lightly underdamped, while ‘Sys2b’ has a significantly lower damping ratio. Studies are also performed on three higher order systems, ‘Sys3’ ($M = 3$), ‘Sys4’ ($M = 4$) and ‘Sys5’ ($M = 5$), which use the same underlying filter as Sys2a, with approximately equal contributions to the output from each kernel (scaled using N_m). The impulse response for each of the underlying linear filters can be seen in Figure 5.2. The corresponding optimal basis function expansions are plotted in Figure 5.3

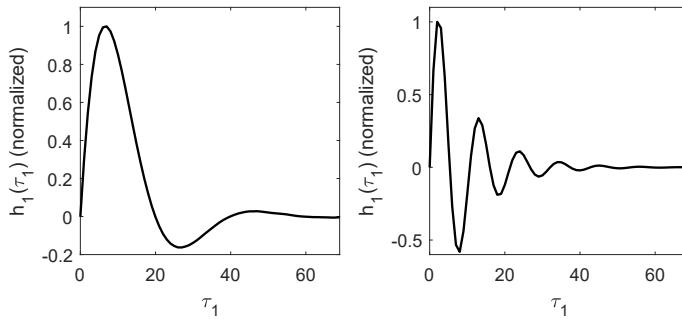


Figure 5.2: Normalized impulse response of the linear filter for Sys2a/Sys3/Sys4/Sys5 (left) and Sys2b (right)

up to $i_1 = 10$. It is clear for Sys2b that a KBF expansion is the more suitable choice, since the resonant behaviour can be captured in a more compact set of basis functions.

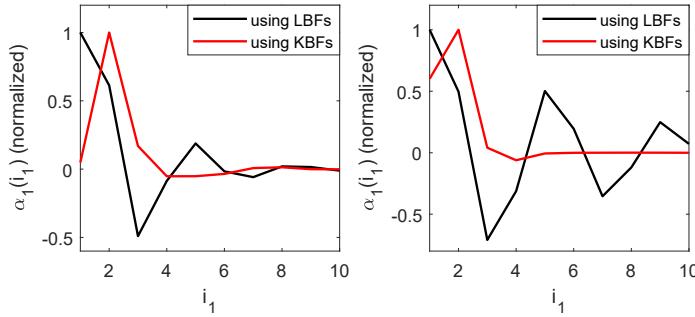


Figure 5.3: Normalized optimal LBF/KBF expansion coefficients of the linear filter for Sys2a/Sys3/Sys4/Sys5 (left) and Sys2b (right)

For each system and method, Monte Carlo studies are performed at Signal to Noise Ratios (SNR) of 20dB and 5dB, with 100 realizations per setting. The input, u , is constructed as a white Gaussian signal, $u(t) \sim \mathcal{N}(0, 1)$, and the data length chosen to be $N = 3412$; only 30% longer than the minimum least squares requirement for Sys2a and Sys2b.

The methods proposed in this chapter are evaluated alongside their unregularized basis function counterparts, as well as a direct time-domain approach. Note that for high nonlinear orders ($M > 2$), it is not possible to evaluate the time domain approach due to its immense computational requirements. The details of each method are:

1. **ReLS**: The Bayesian regularized method from [11] in the time domain, using the TC covariance structure in (3.21), (3.23) and (3.25), and tuned using MLM.
2. **LBF/KBF**: Least squares estimation of LBF/KBF coefficients using (5.9). Optimal basis-generating hyperparameters are assumed to be known a priori (an oracle assumption).
3. **ReLBF/ReKBF**: Regularized estimation of LBF/KBF kernels using the proposal (5.14) and the modified EM approach in Algorithm 2 for hyperparameter tuning and pole selection.

The time domain method is required to estimate each kernel up to a memory length $n_m = 70$, while the basis function methods used a length of $\mathcal{B}_m = 10$. For ReLBF/ReKBF, the optimization tasks in Algorithm 2 are performed using MATLAB's `GlobalSearch` and `fmincon` functions, and the iterative procedure is terminated when within a 0.5% convergence tolerance on all hyperparameters.

For each method and order, Table 5.1 shows the calculated number of unique kernel parameters to be estimated, n_p , and the total number of hyperparameters, n_h . From the table, it is clear that for Sys3, Sys4 and Sys5, ReLS requires an unreasonably high number of parameters, and will become too computationally intensive to be feasible. Hence, we consider only LBF and ReLBF methods for these cases.

Table 5.1: Number of parameters (n_p) and hyperparameters (n_h) for each method and order

Method:		ReLS	LBF	KBF	ReLBF	ReKBF
2nd order	n_p	2555	65	65	65	65
	n_h	6	0	0	8	10
3rd order	n_p	62,195	285	285	285	285
	n_h	10	0	0	13	16
4th order	n_p	1,150,625	1000	1000	1000	1000
	n_h	15	0	0	19	23
5th order	n_p	17,259,389	3002	3002	3002	3002
	n_h	21	0	0	26	31

5.5.2 Results

For each Monte Carlo study, the estimation errors are quantified with the same validation error metric used in [11], calculated by applying a validation input of length 50,000 to both the true system and the estimated system and defining ‘normalized RMS error’ as,

$$E_{NRMS} = \frac{\text{rms}(y_{val} - y_{mod})}{\text{rms}(y_{val})}, \quad (5.20)$$

where y_{val} and y_{mod} are the noise-less outputs of the true and estimated system respectively.

The second order results, presented as boxplots in Figures 5.4 and 5.5, show significant improvements in output prediction using ReLBF and ReKBF. The advantage of Kautz functions for resonant systems is clear in the Sys2b (low damping) results, with ReKBF outperforming all other methods. The unregularized estimates do not perform as well, despite being provided with optimal basis functions prior to identification.

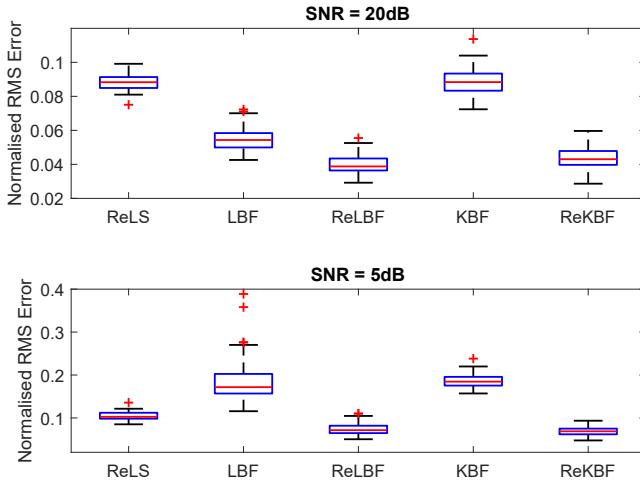


Figure 5.4: Validation errors for Sys2a with SNR = 20dB (top) and 5dB (bottom)

The validation errors for Sys3, Sys4 and Sys5 are given in Figures 5.6, 5.7 and 5.8 respectively, which highlight the benefit of the regularized basis function method in lowering model error to tolerable levels. The time domain method, ReLS, requires unreasonably high numbers of parameters and hence is computationally unfeasible. The least squares LBF model estimates, while computationally tractable, produce extremely inaccurate predictions. The newly proposed regularized method, however, obtains estimates with reasonable accuracy and computation time using a standard computer architecture, which would not previously have been possible with existing estimation methods under these experimental conditions. For comparison, mean computation times are provided, in Table 5.2, for the attempted methods at each order.

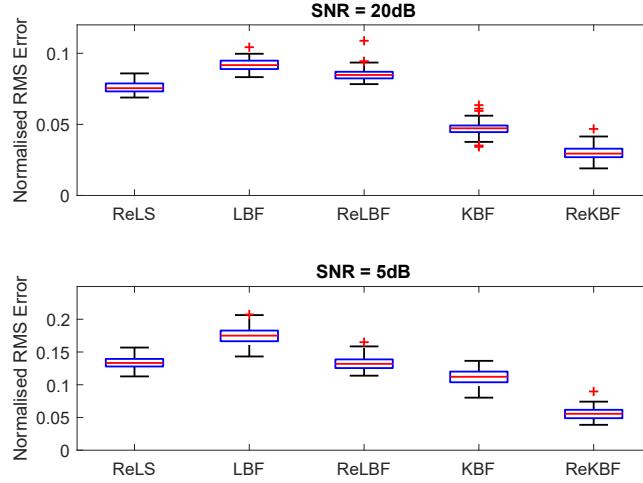


Figure 5.5: Validation errors for Sys2b with $\text{SNR} = 20\text{dB}$ (top) and 5dB (bottom)

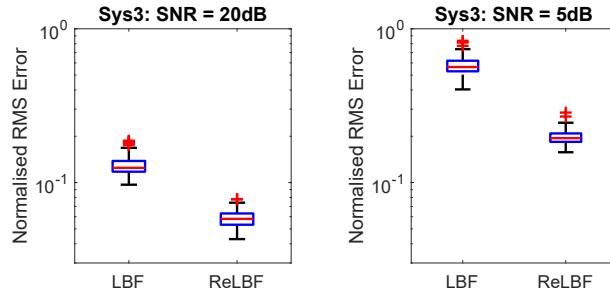


Figure 5.6: Validation errors for Sys3 with $\text{SNR} = 20\text{dB}$ (left) and 5dB (right)

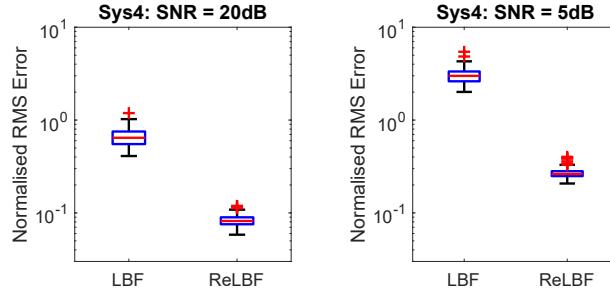


Figure 5.7: Validation errors for Sys4 with $\text{SNR} = 20\text{dB}$ (left) and 5dB (right)

Table 5.2: Mean computation time (s) for each method and order

Method:	ReLS	LBF	KBF	ReLBF	ReKBF
2nd order	376	0.4	0.9	22	32
3rd order	-	0.7	-	48	-
4th order	-	1.3	-	228	-
5th order	-	4.3	-	3687	-

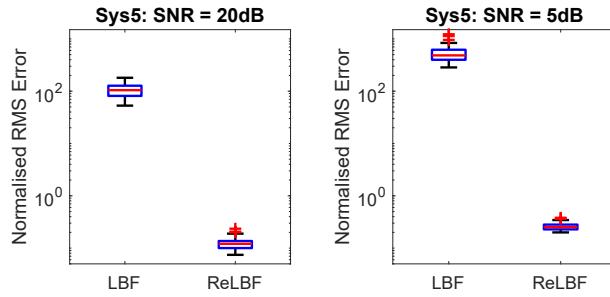


Figure 5.8: Validation errors for Sys5 with SNR = 20dB (left) and 5dB (right)

5.6 Conclusion

This chapter makes a novel proposal to combine the techniques of Bayesian regularization and basis function modeling for the purpose of Volterra series estimation. Some theoretical motivation is provided for imposing standard covariance structures on the resultant basis function kernels, and a modified EM algorithm is constructed as an extension of the results in Chapter 4. The extended algorithm is capable of optimizing both the covariance hyperparameters and the basis-generating hyperparameters, and the iterative procedure will scale more favourably with series order than a direct marginal likelihood approach. The performance of the proposed estimation method has been compared against unregularized least squares estimates, and estimates directly in the time domain, with improved output prediction in all test cases. Furthermore, under the experimental conditions considered in the simulation examples, the proposed method allowed access to computationally feasible and accurate estimates at higher series orders than was previously possible with existing methods in the literature.

Regularized basis function estimation is clearly a promising tool in addressing the major challenge of time domain Volterra series identification: accuracy and computational feasibility for high order models under arbitrary experimental conditions.

5.A Long division of Laguerre basis function filters

For a rational transfer function in the z -domain, performing a polynomial long division between the numerator and denominator will produce the corresponding impulse response. For example, long division of the transfer function $\frac{1}{z-p}$ for some pole $p \in \mathbb{R}$ will produce

$$0 + 1z^{-1} + pz^{-2} + p^2z^{-3} + \dots + p^{i-1}z^{-i} + \dots, \quad (5.21)$$

where the sequence of coefficients, $[0, 1, p, p^2, \dots, p^{i-1}, \dots]$, is the impulse response.

The same process is now applied to the arbitrary order Laguerre filter in (5.18), to reveal the structure of the filter's impulse response:

$$\begin{array}{r}
 0 + [\mathbf{p}_{i-1,1}(\mathbf{i})\mathbf{a}^{i-1}]z^{-1} + [\mathbf{p}_{i-1,4}(\mathbf{i})\mathbf{a}^{i-2}]z^{-2} \\
 + [\mathbf{p}_{i-1,8}(\mathbf{i})\mathbf{a}^{i-3}]z^{-3} \\
 + \dots
 \end{array}$$

$$\begin{array}{r}
 z^i + \sum_{k=0}^{i-1} p_{k,2}(i)a^{i-k}z^k \left(0z^i + \sum_{k=0}^{i-1} p_{k,1}(i)a^kz^k \right) \\
 0 + 0
 \end{array}$$

(end of step 0)

$$\begin{array}{r}
 \sum_{k=0}^{i-1} p_{k,1}(i)a^kz^k \\
 = p_{i-1,1}(i)a^{i-1}z^{i-1} + \sum_{k=1}^{i-1} p_{k-1,1}(i)a^{k-1}z^{k-1}
 \end{array}$$

(remove $i-1$ from sum)
(extract $f_i(1)$ and multiply out)

$$\begin{array}{r}
 p_{i-1,1}(i)a^{i-1}z^{i-1} + \sum_{k=1}^{i-1} p_{k-1,3}(i)a^{2i-k-1}z^{k-1} \\
 + \dots
 \end{array}$$

(end of step 1)

$$\begin{array}{r}
 \sum_{k=1}^{i-1} (p_{k-1,1}(i) - p_{k-1,3}(i)a^{2i-2k})a^{k-1}z^{k-1} + \dots
 \end{array}$$

(remove $i-1$ from sum)

$$\begin{array}{r}
 = p_{i-1,4}(i)a^{i-2}z^{i-2} \\
 + \sum_{k=2}^{i-1} (p_{k-2,1}(i) - p_{k-2,5}(i)a^{2i-2k})a^{k-2}z^{k-2} + \dots \\
 + \sum_{k=2}^{i-1} p_{k-2,6}(i)a^{2i-k-2}z^{k-2} + \dots
 \end{array}$$

(end of step 2)

$$\begin{array}{r}
 \sum_{k=2}^{i-1} (p_{k-2,1}(i) - p_{k-2,7}(i)a^{2i-2k})a^{k-2}z^{k-2} + \dots \\
 = p_{i-1,8}(i)a^{i-3}z^{i-3} \\
 + \sum_{k=3}^{i-1} \dots \\
 + \sum_{k=3}^{i-1} \dots
 \end{array}$$

and so on up to step $i-1$

CHAPTER

6

Practical Case Studies in Time Domain Identification

In this chapter, the theoretical contributions of Chapters 4 and 5 are evaluated on experimental data obtained from nonlinear benchmark systems in the identification community. Firstly, Volterra series model estimates are obtained for a cascaded water tanks system using the regularized basis function approach, with accuracy and computation time compared against the equivalent time domain method. Following this, the same basis function method is applied to a coupled electric drives system to estimate a second order Volterra model. The validation performance of the model is compared against other popular linear and nonlinear model classes estimated using the MATLAB System Identification Toolbox.

6.1 Introduction

Monte Carlo simulations are a useful tool for the analysis of estimation methods, however it is difficult to replicate all of the complexities which feature in a real world identification problem. The numerical examples in Chapters 4 and 5 revealed many favourable properties of the techniques proposed in this thesis, but the simulations relied on an artificially constructed true system which was in the correct model class, had a known maximum nonlinear order, and was disturbed only by measurement noise of the correct Gaussian distribution. Such ideal conditions are never present in practice, and therefore it is important to evaluate the estimation methods in a practical setting as well.

In this chapter, we will consider the estimation of Volterra series models for two real system examples. Both systems are part of a set of established nonlinear benchmark problems used within the system identification community. The first system is known as the cascaded tanks benchmark [113], while the second is referred to as the coupled electric drives [136]. Note that the full set of benchmark problems can be found at www.nonlinearbenchmark.org.

In keeping with the motivation for this part of the thesis, the experimental conditions in both benchmark problems pose a significant challenge for identification of Volterra series models due to the extremely short data records available for estimation. The regularized basis function approach with EM tuning, proposed in Chapter 5, will be examined for each benchmark and compared against existing methods in the literature.

6.2 Cascaded tanks

The cascaded tanks benchmark [113] is a system concerned with fluid level control, which is a common control problem in process industries. Model estimation for the total system is a challenging task due to a number of factors, including combined weak and hard nonlinear dynamics, process noise, unknown initial conditions and a short estimation data record. While there have been several successful grey-box modeling attempts (see e.g. [90]) which consider the fluid dynamics and process noise directly, in this chapter the identification problem will be undertaken without the use of any prior knowledge, since this is the strength of a nonparametric Volterra series approach. More specifically, the computational burden and prediction accuracy of regularized basis function estimation will be assessed and compared against equivalent estimates obtained using the regularized time domain method in [11], which was applied to the cascaded tanks benchmark in [9].

6.2.1 System description

The system consists of two vertically cascaded tanks, each with small valve openings in the bottom. The top tank is fed with water via a pump which is controlled by a voltage. Water flows through a valve in the top tank to the bottom tank, and through a valve in the bottom tank to a reservoir underneath. The height of the water in the lower tank is the output of the system, measured in volts by a noisy, uncalibrated capacitive sensor. The process is shown in Figure 6.1.

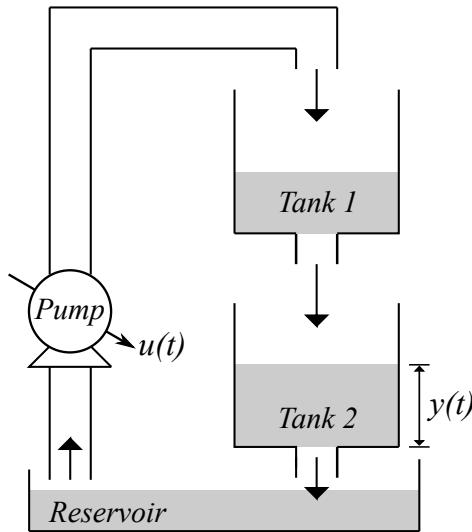


Figure 6.1: Diagram of cascaded tanks benchmark system with typical water flow

Ignoring saturation, the system is weakly nonlinear based on the flow equations through the valves in each tank, which can be derived from Bernoulli's principle. The system also contains a hard nonlinearity in the form of water level saturation. Once the maximum level of either tank is reached, an overflow occurs. Furthermore, saturation in the top tank introduces input-dependent process noise, since some of the overflowing water can fall into the lower tank and affect the output.

6.2.2 Measurement process

Two sets of data have been recorded for the benchmark: an estimation set and a validation set. For both datasets, the voltage input to the tank system is excited using one period of a multisine which excites frequencies from 0 to 0.0144 Hz. Using a sampling period of $T_s = 4\text{s}$, this gives input vectors of length $N = 1024$. The corresponding output measurements are obtained from the capacitive level sensor of the lower tank, which may introduce measurement noise. The system is not in steady state during the measurement window, and the initial states of the system are unknown but approximately equal for both datasets. The recorded signals are plotted in Figure 6.2 in the time domain (top) and frequency domain (bottom).

While the hard saturation nonlinearity is clearly visible in the time domain output signals, there is also evidence of nonlinearity in the frequency domain, where some output power exists at unexcited frequencies.

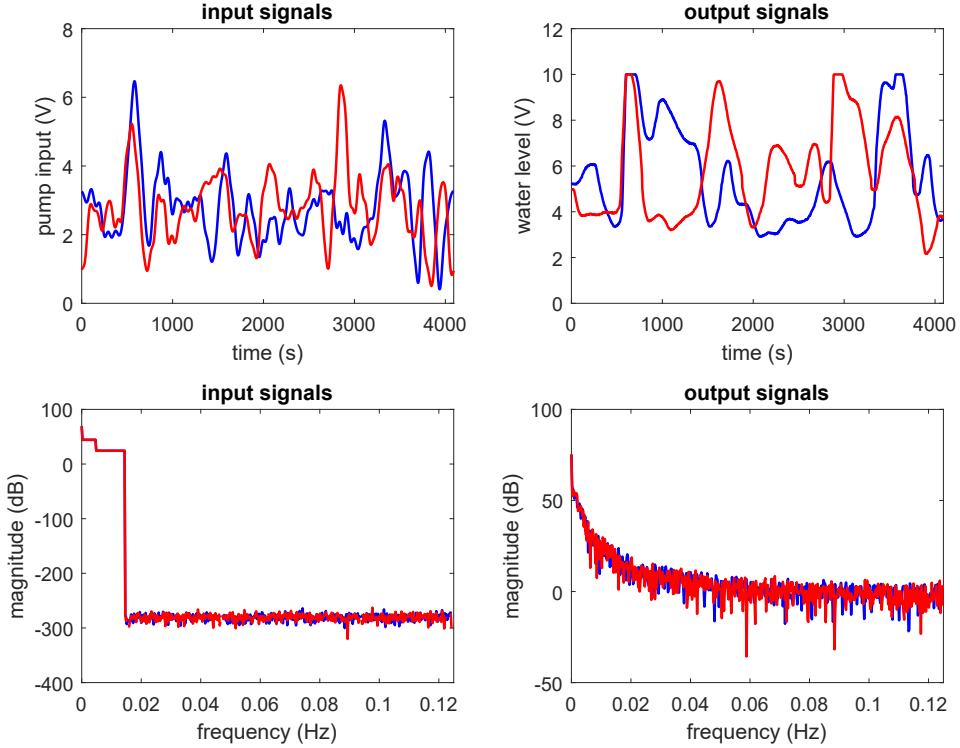


Figure 6.2: Estimation (blue) and validation (red) datasets for the cascaded tanks in the time (top) and frequency (bottom) domain

6.2.3 Problem statement

To maintain consistency with the estimates obtained in [9], the identification problem is considered exactly as stated in the benchmark document [113]. The required outcome is to identify a model of the system using the estimation dataset, and report the accuracy of the estimated model in predicting the output of the *entire* 1024-sample validation dataset using the following metric,

$$e_{RMSt} = \sqrt{\frac{1}{N_v} \sum_{t=1}^{N_v} (y_{mod}(t) - y_v(t))^2}, \quad (6.1)$$

where $N_v = 1024$ is the length of y_v , the validation output, and y_{mod} is the modeled output for the same validation input. Note that prediction of the entire output vector requires estimation of the unknown initial states, since the start of the record will be affected by inputs applied prior to the measurement window.

6.2.4 Estimating unknown initial states

In the nonparametric Volterra series context, the initial states problem becomes an issue of ‘past inputs’, since Volterra kernel outputs are functions of lagged inputs only. Hence, in order to estimate/validate using the entire output vector, $\{y(t)\}_{t=1}^{1024}$, we require knowledge of $u(t)$ for $n < t < 0$ and some reasonable memory length n . Of course, no knowledge of these past inputs can be gained from the provided datasets, so two strategies are discussed in this section in order to continue estimating/validating with the complete output vector. Both strategies are applied to the benchmark data in Section 6.2.5.

Bayesian transient estimation

For consistency, one approach we consider is the method used in [9], whose results will be the subject of comparison for this benchmark. In the paper, which relies on previous results established in [29], the input is assumed to be periodic, i.e. $u(t - N) = u(t) \forall t$, where $N = 1024$ for the cascaded tanks. However, since the system is not in steady state, this assumption is clearly in error, and we expect to see a ‘transient’ function which describes the difference between the measured output and the ideal steady-state output. This difference will be denoted θ_{tr} , which has length $n_{tr} < N$.

In this case, the regression structure for basis function kernels in (5.8) can be updated to include the transient,

$$Y = \Phi_f^T \alpha + \Phi_\delta \theta_{tr} + E, \quad (6.2)$$

where the Y vector now contains all N samples of $y(t)$, Φ_f contains measured *and* unmeasured input values, and Φ_δ is a $N \times n_{tr}$ rectangular identity. Staying within the Bayesian framework, a Gaussian assumption is placed on θ_{tr} , such that the regularization problem can be augmented as follows,

$$\begin{aligned} \alpha' &= [\alpha^T \ \theta_{tr}^T]^T, \quad \Phi' = [\Phi_f^T \ \Phi_\delta], \\ \hat{\alpha}'_{ReLS} &= \arg \min_{\alpha'} \|Y - \Phi' \alpha'\|_2^2 + \sigma^2 \alpha'^T [P']^{-1} \alpha', \end{aligned} \quad (6.3)$$

where P' is an augmented prior covariance matrix given by

$$P' = \begin{bmatrix} P & \mathbf{0} \\ \mathbf{0} & P_{tr} \end{bmatrix},$$

and P_{tr} is the covariance of θ_{tr} , which is chosen to have the linear DC structure given in (2.38). The prior covariance, P , for the original parameter vector, α , is also chosen to be DC-based and is constructed as in previous chapters using (3.21), (3.24), and (3.25).

The hyperparameter set, η (which now includes hyperparameters for P_{tr}) is still tuned with EM using Algorithm 2, however some minor modifications are made to include the augmented transient components. The P_{tr} hyperparameters are then re-used in the validation dataset, to re-estimate and remove the transient function from the output.

Remark 6.1. *It is important to note that while the Bayesian transient method is included to be consistent with [9], its requirement to re-estimate the validation transient is not in the spirit of the benchmark problem. Instead, it would be fairer to keep the validation data only for validation purposes, and not for the estimation of any parameters.*

Constant past input assumption

Observing Remark 6.1, another approach can be formulated which adds less complexity to the EM optimization, and requires no re-estimation on the validation data. In this approach, we restrict the unmeasured input, $\{u(t)\}_{t=-\infty}^{-1}$ to the class of constant sequences, i.e.

$$u(t) = c \quad \forall t < 0; \quad c \in \mathbb{R}^+.$$

The constant, c , is tuned during the optimization in Algorithm 2 such that the following condition is satisfied in each iteration,

$$y_P^{(k)}(0) = y(0),$$

where $y_P^{(k)}(0)$ is the output value predicted using $\hat{\alpha}^{(k)}$ and past inputs $u(t) = c$ for $t < 0$ in the regressor, Φ_f .

6.2.5 Identification results

The regularized estimation method from Chapter 5 (ReLBF) is applied for several orders of the model structure in (5.7) expanded using Laguerre basis functions:

- A linear LBF expansion, i.e. excluding α_0 and setting $M = 1$
- A 2nd order Volterra-Laguerre series ($M=2$)
- A 3rd order Volterra-Laguerre series ($M=3$)

The truncation lengths at each order are chosen as $\mathcal{B}_1 = 15$, $\mathcal{B}_2 = 10$ and $\mathcal{B}_3 = 8$, having been obtained through a typical iterative identification process. For the Bayesian transient removal method [9], the length of the transient is defined as $n_{tr} = 100$.

Hyperparameter optimization is achieved using the MATLAB `GlobalSearch` and `fmincon` functions to perform all non-convex minimizations in Algorithm 2.

Computation times for the estimation are measured on an Intel Xeon 3.50 GHz processor.

The two methods for dealing with unknown initial states are denoted ‘Bayesian transient’ for the first method in Section 6.2.4 and ‘Constant input’ for the second method.

Estimation and validation

Modeling the system first with a linear LBF expansion, a reasonable result is achieved, but the shortcomings of a linear model are clear. The estimated LBF coefficients are plotted in Figure 6.3, along with the corresponding time-domain impulse response. The choice of $\mathcal{B}_1 = 15$ is supported by the estimate, and the decay time of the impulse response suggests that $n_{tr} = 100$ is a reasonable choice, since the transient function typically decays at the same rate as the system response, at least for linear systems [67].

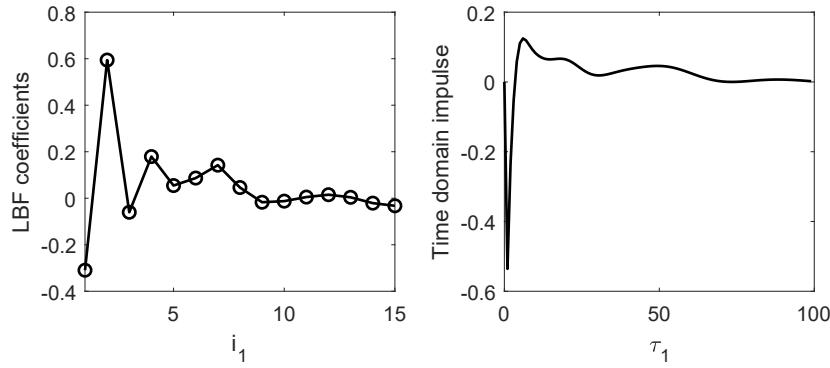


Figure 6.3: Linear model for the cascaded tanks: estimated LBF coefficients (left) and equivalent time domain FIR (right)

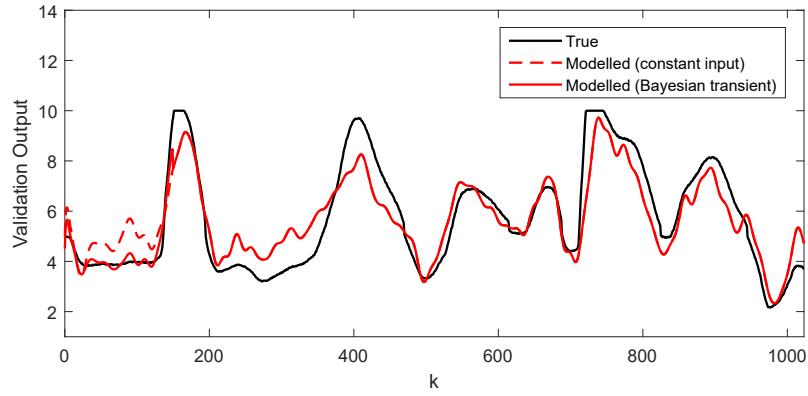


Figure 6.4: Linear model for the cascaded tanks: true (black) and modeled (red) validation outputs using both transient removal methods

The true and modeled validation outputs for the linear estimate are given in

Figure 6.4, where we see that the modeled output is quite poor in the first half of the record. The Bayesian transient approach is also seen to produce a better result than the constant input assumption, although it relies on some estimation using the validation data. Overall, there is clearly room for improvement, which can likely be achieved by estimating higher order terms of the Volterra-Laguerre series.

Estimating a second order Volterra-Laguerre model, the validation performance is significantly improved. The constant kernel is estimated as $\hat{\alpha}_0 = -4.55$, and the first and second order basis function kernels are plotted in Figure 6.5. The true and modeled validation outputs are given in Figure 6.6, where it can be observed that the output predicted by the model is visibly more accurate than in the linear case. Furthermore, the Bayesian transient and constant input methods both obtain a good prediction of the initial 100 output samples.

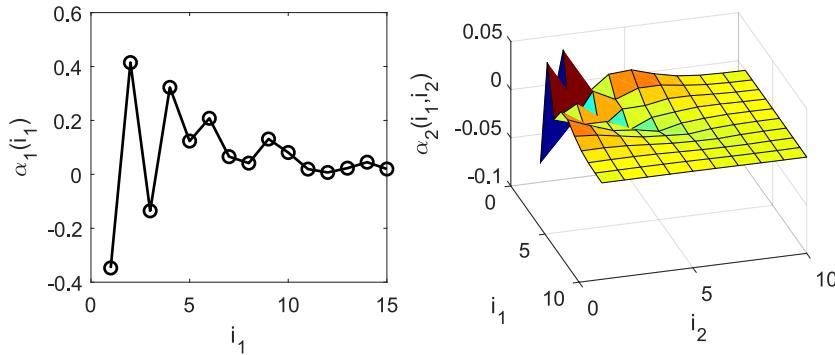


Figure 6.5: Volterra model ($M=2$) for the cascaded tanks: estimated 1st order (left) and 2nd order (right) LBF kernels

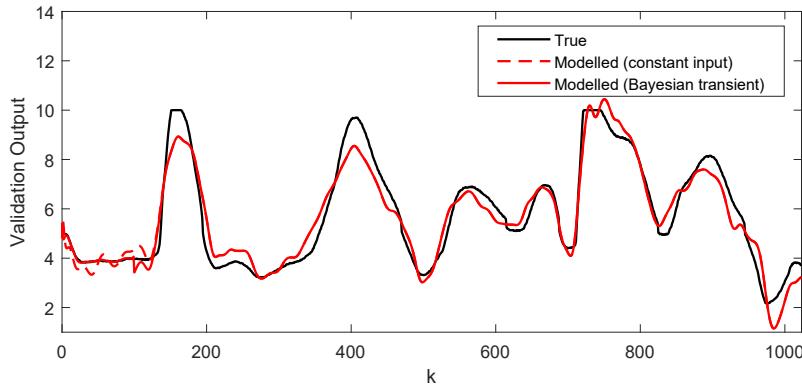


Figure 6.6: Volterra model ($M=2$) for the cascaded tanks: true (black) and modeled (red) validation outputs using both transient removal methods

The third order Volterra-Laguerre model has also been estimated, however the improvement in validation performance is negligible with respect to the second order model. The error metrics and computation times are still reported.

The RMS error in the validation is computed for each model and method using

(6.1), and these errors are compared, in Table 6.1, with the time domain results reported in [9]. The results show very similar validation performance between the time domain and LBF approach. The constant input method of transient removal is seen to provide essentially the same accuracy as the Bayesian method (with the exception of the linear case), despite its more simple nature.

Table 6.1: Comparison of validation errors between methods

Model	e_{RMST} (from (6.1))		
	Birpoutsoukis et. al. (2017) [9]	ReLBF (Bayesian)	ReLBF (Constant)
Linear	0.84	0.85	0.91
Volterra (M=2)	0.55	0.56	0.57
Volterra (M=3)	0.54	0.56	0.56

While the validation performance of the time domain and basis function Volterra models is extremely similar for the cascaded tanks benchmark, significant advantages of the basis function formulation can be seen when comparing the numbers of unique parameters required, and the computation time required for model estimation. In Table 6.2, the total parameter numbers and computation times are reported for the LBF method, alongside the corresponding figures for the time domain approach as reported in [10]. It is clear that a LBF formulation of the regularized problem produces far more compact Volterra models, with an order of magnitude difference in the parameter numbers. As a direct consequence, the difference in computation time is even greater, where the Volterra-Laguerre estimates are computed two or three orders of magnitude faster.

Table 6.2: Comparison of computation time and parameter numbers between methods

Model	Parameters		Comp. Time (mins)	
	Time domain	ReLBF	Time domain	ReLBF
Linear	92	15	3	0.1
Volterra (M=2)	721	71	210	0.6
Volterra (M=3)	860	191	390	2

6.3 Coupled electric drives

The coupled electric drives is a benchmark system inspired by an example which appeared in [133]. The original example was designed to represent machines which transfer a belt of material from one spool to another using two electric drives and

a pulley. In the benchmark problem, a data set of only 500 samples is provided for estimation purposes. The system has previously been modeled using block-oriented models, where the structure is derived from knowledge of the relevant physical interactions [136]. As was done for the cascaded tanks benchmark, here the identification problem will be approached from a position of no prior knowledge, to highlight the versatility of Volterra series modeling even when using an extremely short estimation dataset. The comparison for the drives system will be between regularized basis function estimates (using ReLBF) and estimates obtained using several other popular nonlinear model classes.

6.3.1 System description

The benchmark system is configured as shown in Figure 6.7, where the input is simultaneously applied to two motors driving a flexible belt, which in turn rotates a pulley being held by a spring. The angular speed of the pulley is measured by a pulse transducer, whose measurement is sent through an analogue low-pass and anti-aliasing filter to produce the final speed output which can be used for control purposes.

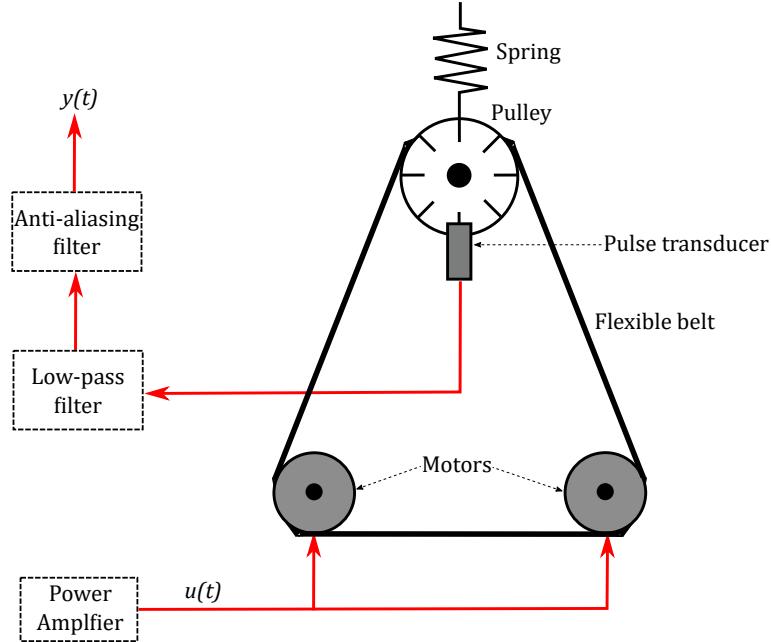


Figure 6.7: Schematic of the coupled electric drives benchmark system

The spring and flexible belt add lightly damped dynamic modes to the system, and there will also be time constants associated with the motors, however these are all linear effects. The modeling complication is brought about by the pulse transducer, which is insensitive to the *direction* of rotation of the pulley, since it simply counts pulses to approximate angular velocity. As the motors can be driven in either

direction, the transducer can be considered as having an ‘absolute value’ nonlinearity, such that the total system is well approximated by a Wiener-Hammerstein block structure, as depicted in Figure 6.8.

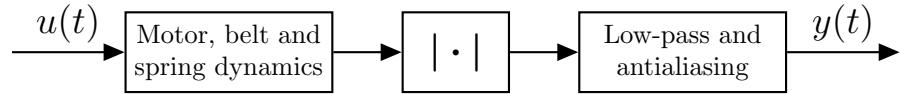


Figure 6.8: Wiener-Hammerstein block structure for the coupled electric drives

6.3.2 Measurement process

While some physical modeling insights can be obtained by analyzing the system set-up, it is the experimental input-output data which will be used to identify and assess candidate models. An estimation and a validation dataset have been recorded, both containing 500 input-output samples recorded at 50 Hz. The inputs are constructed from a Pseudo-Random Binary Sequence (PRBS) which is multiplied by a noise sequence with uniform distribution. The input and output signals for each set are shown in Figure 6.9. The validation input is seen to be the estimation input shifted by an offset of +0.5V, and the absolute value nonlinearity is clearly visible in the output signals.

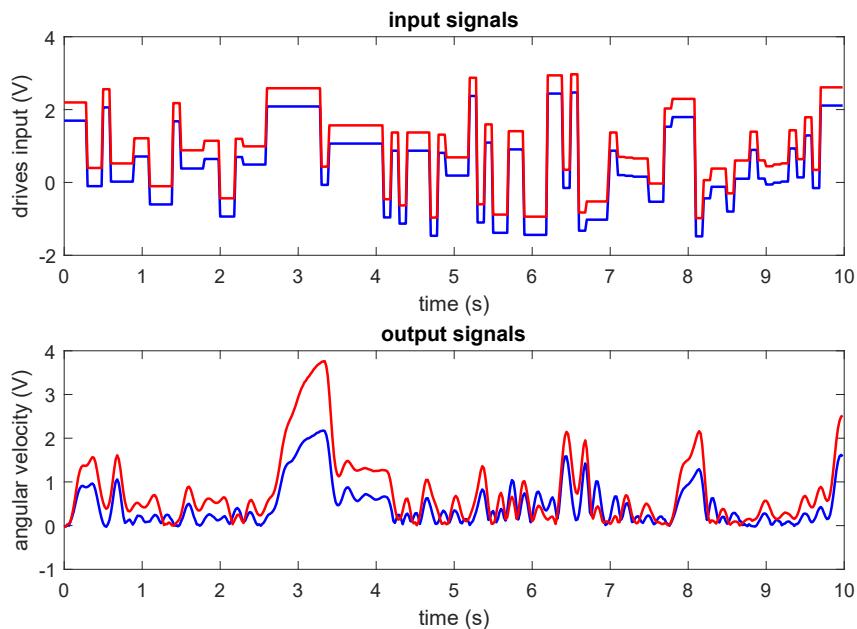


Figure 6.9: Input (top) and output (bottom) signals for the coupled electric drives estimation (blue) and validation (red) datasets

6.3.3 Problem statement

The identification problem for the coupled electric drives is to estimate a nonlinear model using the estimation data, and use this model to predict the output for the validation dataset. Unlike the cascaded tanks benchmark, the issue of estimating initial states will not be considered. Instead, the model predicted output begins at the 60th sample, when enough measured input information is available.

The performance of each nonlinear model on the validation data is assessed numerically, using a normalized root mean square error (NRMSE) metric defined by

$$\text{NRMSE} = 1 - \frac{\|Y_{val} - Y_{mod}\|_2}{\|Y_{val} - \text{mean}(Y_{val})\|_2}, \quad (6.4)$$

where Y_{val} is the true validation output vector and Y_{mod} is the model-predicted output vector.

6.3.4 Identification results

LBF-expanded Volterra kernels are estimated for the benchmark system using the ReLBF method proposed in Chapter 5. Due to the limited number of estimation samples available, the order of the Volterra series is fixed at $M = 2$, and using no prior knowledge of the system, the number of basis functions for each order is chosen to be $\mathcal{B}_1 = \mathcal{B}_2 = 20$, such that the total number of unique parameters requiring estimation (231) is roughly half the number of data samples available. All non-convex optimization problems in Algorithm 2 are solved using the MATLAB `GlobalSearch` and `fmincon` functions to locate a global minimum.

The first order kernel, $\hat{\alpha}_1$, is estimated to be negligible. This is to be expected, since the absolute value nonlinearity is an *even* function and as such generates only even order Volterra kernels in the series expansion. The second order LBF kernel

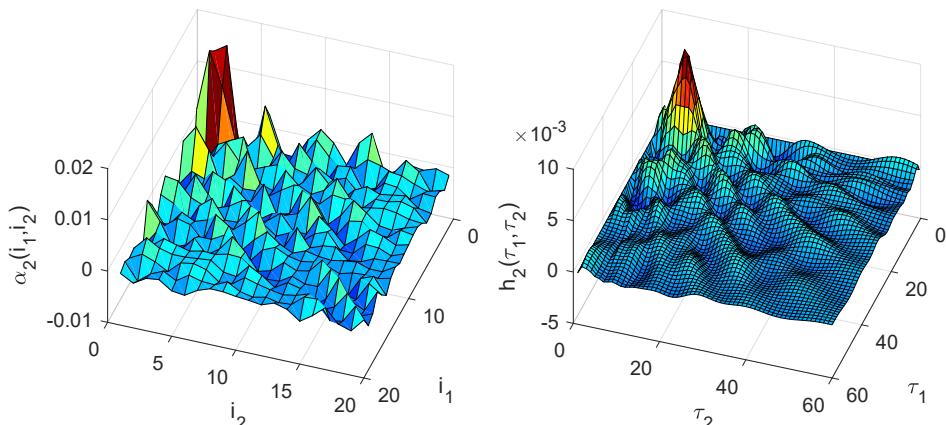


Figure 6.10: Estimated 2nd order LBF kernel (left) and its time domain equivalent (right) for the coupled electric drives

estimate, $\hat{\alpha}_2$, is shown in Figure 6.10 as well as its time domain equivalent, \hat{h}_2 . The zeroth order kernel is estimated as $\hat{\alpha}_0 = 0.102$.

Applying the validation input to the estimated Volterra-Laguerre model produces the output prediction shown in Figure 6.11, which is plotted with the given validation output. Despite using only 500 samples for estimation, the regularized basis function approach provides a very accurate model which is suitable for prediction or control purposes.

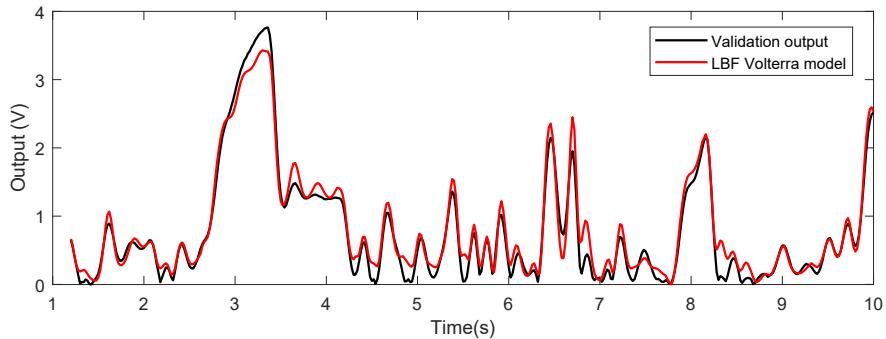


Figure 6.11: True and model-predicted validation outputs for the Volterra-Laguerre model applied to the coupled electric drives benchmark

Comparison with Alternate Methods

In order to provide some context when assessing the performance of the proposed method, several alternate model classes and methods are also applied to the coupled electric drives benchmark. The other models considered are:

1. Linear state space
2. Linear transfer function
3. Nonlinear ARX (NARX)
4. Wiener block structure

For each model class, the estimation dataset is used to estimate a model using the appropriate method provided by MATLAB's System Identification toolbox. As in the proposed ReLBF method, no prior knowledge is used in the estimation, and the model orders for each case are chosen based on a cross-validation approach. High order polynomials are used for nonlinear function estimates. Note that Wiener-Hammerstein models cannot be estimated using the toolbox, however this case has already been treated in [136].

The predicted validation output from each model is plotted in Figure 6.12, along with the true experimental output (in black). It is clear that for such a short estimation dataset, and without the use of prior knowledge, none of the estimated

models provide accurate predictions over the entire validation set for the coupled electric drives system.

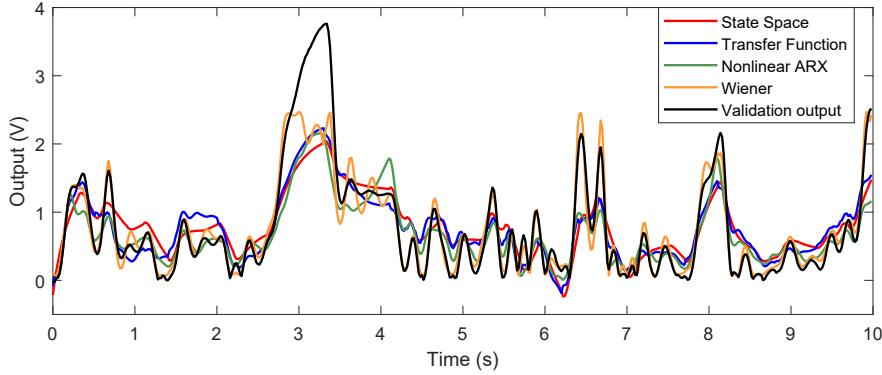


Figure 6.12: True and model-predicted validation outputs for toolbox methods in MATLAB applied to the coupled electric drives benchmark

The NRMSE error metric (6.4) for each toolbox method is compared in Table 11.1 against the method proposed in this thesis (ReLBF), highlighting the superior performance of the regularized nonparametric approach. From the four toolbox models, the Wiener structure produces the next best validation results, which can be attributed to the Wiener-Hammerstein nature of the true system (see Figure 6.8), where the final linear filter block contains dynamics which could be considered negligible. This is further supported by the fact that similar prediction performance was achieved using a Wiener-Hammerstein model in [136].

Table 6.3: NRMSE in validation for proposed and toolbox methods

Model	State Space	Transfer Func.	NARX	Wiener	ReLBF
NRMSE	0.402	0.457	0.446	0.639	0.811

6.4 Conclusion

The Volterra series has historically been regarded as an impractical tool for data-driven modeling. The benchmark examples in this chapter, however, have shown that by using a more compact basis function formulation, and efficiently regularizing the estimation to impose known properties of smoothness and decay, the Volterra series *can* be a broad and powerful tool for nonlinear system identification.

The cascaded tanks benchmark revealed that a regularized basis function method can be adapted to deal with unknown initial states in a system. The resulting 2nd and 3rd order series estimates gave reasonably accurate prediction performance, despite the short data record used for estimation. This performance was compared against the time domain estimation method in [9], showing equivalent validation metrics

but a stark decrease in parameter numbers and computation times for the proposed basis function approach.

The coupled electric drives benchmark provided further evidence for the proposed method. Despite having an even shorter estimation dataset of 500 samples, and without incorporating prior knowledge of the drives system into the modeling process, the estimated Volterra-Laguerre kernels provided accurate predictions in the validation step. Furthermore, the proposed method significantly outperformed linear and nonlinear models provided by the established System Identification toolbox in MATLAB in terms of the considered error metric.

PART

III

Frequency Domain Identification

CHAPTER



7

Gaussian Process Regression in the Frequency Domain - the Parallel Hammerstein Case

Frequency domain estimation of parallel Hammerstein systems is considered in this chapter, using nonlinear output frequency response functions (NOFRFs). The NOFRFs are a series of one-dimensional frequency functions which are input dependent in general, however, they can be shown to be input independent linear filters in the parallel Hammerstein case. A Gaussian process regression (GPR) framework is proposed for the NOFRF estimation, using concepts adapted from the linear identification literature. When compared to the existing multi-level excitation method, the proposed GPR approach is less sensitive to noisy measurements and requires fewer experimental restrictions.

7.1 Introduction

In Chapter 3.2, several competing representations were discussed for Volterra series modeling in the frequency domain, where each representation offers different benefits for interpretation and estimation [103], [26]. The GFRFs give perhaps the most natural representation, being multidimensional Fourier transforms of their corresponding time domain Volterra kernels [46]. However, due to the complex multidimensional nature of the GFRFs, the NOFRF model was proposed [64] to increase the ease of system analysis. The model (see (3.6)) contains only one-dimensional frequency functions which have a more intuitive interpretation from the perspective of linear systems theory, however, the functions are input dependent in general. This input dependence has restricted the use of NOFRFs to cases where input behaviour remains uniform over time, such as fault detection applications [93], [20], [139] and nonlinearity detection [66].

The NOFRF model proposal in [64] was accompanied by a data-driven identification algorithm, which uses multi-level excitation to estimate NOFRFs in a regression framework. This traditional method requires multiple experiments on the system under conditions that may not be possible in practice if the system excitation cannot be precisely controlled. The method also employs a least squares estimator that is sensitive to noisy measurements. In the special case of parallel Hammerstein systems it is possible to perform more sophisticated estimation, since the NOFRFs become independent of the applied input and the form of the functions closely resemble that of the underlying linear filters in the system's block structure. This property allows us to adopt well-developed concepts from linear identification theory.

The Bayesian regularization method for FIR estimation [94], which provided the framework for contributions in Part II of this thesis, also has a linear frequency domain interpretation. Developed in [67], the frequency domain method considers estimation of FRFs via Gaussian process regression (GPR). However, some additional challenges arise in the frequency domain case, due to the emergence of a transient function in the output spectrum when using non-steady-state data and the mixture of real and complex parameters contained in the FRF parameter vector. The results in [67] will provide a foundation for many of the contributions contained in Part III of the thesis.

In this chapter, a GPR method is developed for estimating NOFRFs in the special case of a parallel Hammerstein system. The method is an extension to the linear frequency domain method in [67]. By framing the NOFRFs as normally distributed quantities with standard prior covariance structures, we can estimate all NOFRFs in a regularized fashion using only one experiment and with less restrictions than the traditional multi-level excitation method. Numerical examples reveal that the method proposed here is also significantly less sensitive to measurement noise than the traditional method in [64]. The general case of non-steady-state data is

also considered, where the GPR approach can be adapted to minimize the effect of transients on NOFRF estimation.

The proposed method is distinct from other parallel Hammerstein identification algorithms in that it is a fully nonparametric method applied directly in the frequency domain. Traditionally, the linear dynamic and nonlinear blocks are estimated separately, and parametrically, in an iterative scheme [45], [114], however the resulting models do not provide direct intuition on frequency domain behaviour. More recently, GPR has been applied to Hammerstein identification in the time domain [104], but the approach was limited to a single Hammerstein branch.

7.2 Real/complex normal distributions

The parameter vectors considered in the current chapter, as well as Chapter 8, are free to contain both real and complex parameters. In order to express the distribution of Gaussian random vectors that contain both real and complex entries, the complex normal distribution [115] is not sufficient, since the associated augmented covariance matrix will be singular. Consequently, a hybrid Real/Complex Gaussian (RCG) distribution framework was developed in [67], which will also be used in the sequel for parameter estimation. In this section we define some relevant notation and properties to enhance the readability of Chapters 7 and 8.

Definition 7.1 (Real/Complex Gaussian vector). *A random vector*

$$X = \begin{bmatrix} X^{\mathbb{R}} \\ X^{\mathbb{C}} \end{bmatrix}, \quad X^{\mathbb{R}} \in \mathbb{R}^{n_r}, \quad X^{\mathbb{C}} \in \mathbb{C}^{n_c} \quad (7.1)$$

is said to be a RCG vector if the vector

$$\begin{bmatrix} X^{\mathbb{R}} \\ \Re X^{\mathbb{C}} \\ \Im X^{\mathbb{C}} \end{bmatrix}$$

is Gaussian distributed, where \Re and \Im give the real and imaginary parts respectively.

Notation 7.2 (Augmented vector). *For a RCG vector, X , the augmented vector is defined as*

$$\tilde{X} = \begin{bmatrix} X^{\mathbb{R}} \\ X^{\mathbb{C}} \\ \bar{X}^{\mathbb{C}} \end{bmatrix}, \quad (7.2)$$

where \bar{X} denotes the complex conjugate of a vector X .

Notation 7.3 (Augmented mean and covariance). A RCG vector X has a real/complex normal distribution denoted by

$$X \sim \mathcal{RCN}(\mu, \Sigma), \quad (7.3)$$

where $\mu = \mathbf{E}\{\tilde{X}\}$ is the augmented mean, $\Sigma = \mathbf{E}\{(\tilde{X} - \mu)(\tilde{X} - \mu)^H\}$ is the augmented covariance, and H is the Hermitian transpose operator. The augmented covariance, Σ , can be decomposed using covariance functions R , Q , K , and relation function C , as

$$\Sigma = \begin{bmatrix} R & Q & \bar{Q} \\ Q^H & K & C \\ \bar{Q}^H & C^H & \bar{K} \end{bmatrix} \quad (7.4)$$

$$\begin{aligned} \text{where } R &= \mathbf{E}\{(X^{\mathbb{R}} - \mathbf{E}\{X^{\mathbb{R}}\})(X^{\mathbb{R}} - \mathbf{E}\{X^{\mathbb{R}}\})^T\}, \\ Q &= \mathbf{E}\{(X^{\mathbb{R}} - \mathbf{E}\{X^{\mathbb{R}}\})(X^{\mathbb{C}} - \mathbf{E}\{X^{\mathbb{C}}\})^H\}, \\ K &= \mathbf{E}\{(X^{\mathbb{C}} - \mathbf{E}\{X^{\mathbb{C}}\})(X^{\mathbb{C}} - \mathbf{E}\{X^{\mathbb{C}}\})^H\}, \\ C &= \mathbf{E}\{(X^{\mathbb{C}} - \mathbf{E}\{X^{\mathbb{C}}\})(X^{\mathbb{C}} - \mathbf{E}\{X^{\mathbb{C}}\})^T\}. \end{aligned}$$

For the following properties, consider $A \sim \mathcal{RCN}(\mu_A, \Sigma_A)$ and $B \sim \mathcal{RCN}(\mu_B, \Sigma_B)$

Property 7.4 (Independent Sum). For A and B independent and of equal dimension, their sum is distributed as

$$A + B \sim \mathcal{RCN}(\mu_A + \mu_B, \Sigma_A + \Sigma_B). \quad (7.5)$$

Property 7.5 (Hadamard Product). For a complex vector U with equal dimension to A , the Hadamard product $U \circ A$ is distributed as

$$U \circ A \sim \mathcal{RCN}(\tilde{U} \circ \mu_A, (\tilde{U} \tilde{U}^H) \circ \Sigma_A). \quad (7.6)$$

Property 7.6 (Conditional Distributions). If A and B are jointly (Gaussian) distributed, the conditional distribution of A given B is given by

$$A|B \sim \mathcal{RCN}(\mu_A + \Sigma_{AB}\Sigma_B^{-1}(\tilde{B} - \mu_B), \Sigma_A - \Sigma_{AB}\Sigma_B^{-1}\Sigma_{BA}), \quad (7.7)$$

where $\Sigma_{AB} = \mathbf{E}\{(\tilde{A} - \mu_A)(\tilde{B} - \mu_B)\}$.

7.3 The NOFRF model

Some relevant concepts relating to the NOFRF model structure will be reviewed and clarified here. First, let $u(t)$ and $y_0(t)$ be discrete, noiseless input and output

measurements respectively from a nonlinear system, where $t = 0, 1, \dots, N - 1$. The N -point discrete Fourier transforms (DFTs) of u and y_0 will be labelled $U(k)$ and $Y_0(k)$ respectively, where $k \in (0, 1, \dots, N - 1)$ indicates the frequency bin.

The original NOFRF model definition in [64] assumes that the frequency domain output, $Y_0(k)$, is transient-free. The assumption implies that the input to the system is periodic in N and has been applied for an infinitely long period before measurements were taken, such that the output measurement is one period of a steady-state N -periodic signal. In this case, the noise-less output spectrum is given by (3.6), repeated here for convenience:

$$Y_0(k) = H_0(k) + \sum_{m=1}^M Y_m(k), \quad (7.8)$$

$$Y_m(k) = G_m(k)U_m(k).$$

Recall that $G_m(k)$ is the m^{th} order NOFRF, $H_0(k)$ is a zero frequency contribution, and $U_m(k)$ is the DFT of $u^m(t)$. Recall also the equivalent NOFRF block structure, which is repeated here in Figure 7.1, showing the intuitive interpretation of NOFRFs as linear filters on the nonlinear input quantities.

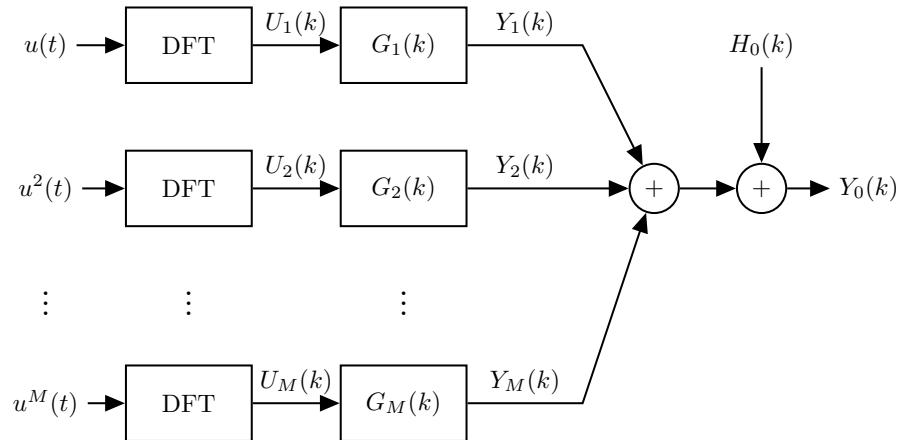


Figure 7.1: Equivalent block structure of the NOFRF model

In this chapter, for estimation, the model will also include additive white Gaussian measurement noise at the output. Thus, the *observed* output spectrum is given by

$$Y(k) = Y_0(k) + E(k), \quad (7.9)$$

where $E(k)$ is assumed i.i.d. with augmented covariance $\sigma^2 I$.

7.3.1 The parallel Hammerstein case

The NOFRFs are input-dependent in general, since they act on exponents of the input rather than the input directly. There exist some special cases in which the frequency functions become independent of the applied input, notably when the system has a parallel Hammerstein structure with polynomial nonlinearities. This structure is depicted in Figure 7.2, where there are L distinct Hammerstein branches, and each branch has a nonlinearity with maximum degree less than or equal to M .

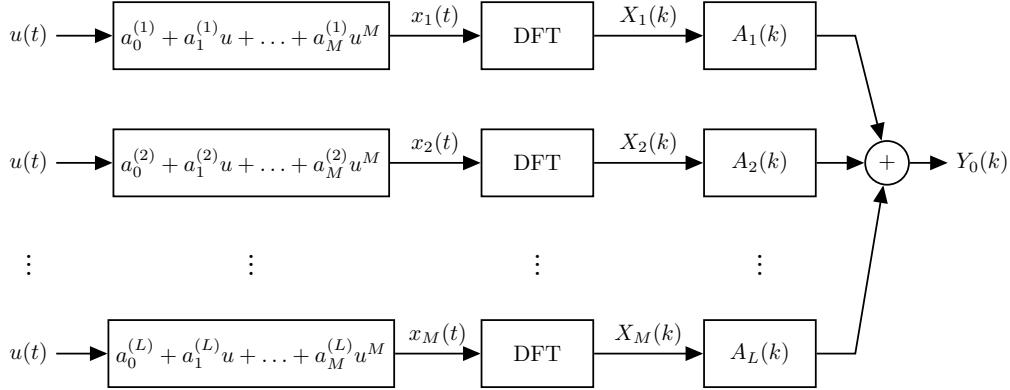


Figure 7.2: Frequency domain block structure of a parallel Hammerstein system with polynomial nonlinearities

To understand why the NOFRFs of such a system are input-independent, the structure can be rearranged by exploiting the linearity of the DFT and the filters, A_1, A_2, \dots, A_L . An alternate parallel Hammerstein block structure is given in Figure 7.3, where each exponent of the input is collected in a separate branch, and $H_0(k)$ is a zero frequency contribution produced by the polynomial constants, $a_0^{(i)}$, $i = 1, \dots, L$. Clearly, this alternate structure matches the NOFRF model structure in Figure 7.1, where the NOFRFs are replaced by

$$G_m = a_m^{(1)}A_1 + \dots + a_m^{(L)}A_L \quad m = 1, 2, \dots, M. \quad (7.10)$$

Thus, for a parallel Hammerstein system, the NOFRFs are linear combinations of the filters from each branch, and since the original filters are all linear and input-independent, the resulting NOFRFs must be also. This is the case which will be considered in the current chapter, where the NOFRFs can be treated as linear filters for identification, and the resulting model will be valid regardless of the system input.

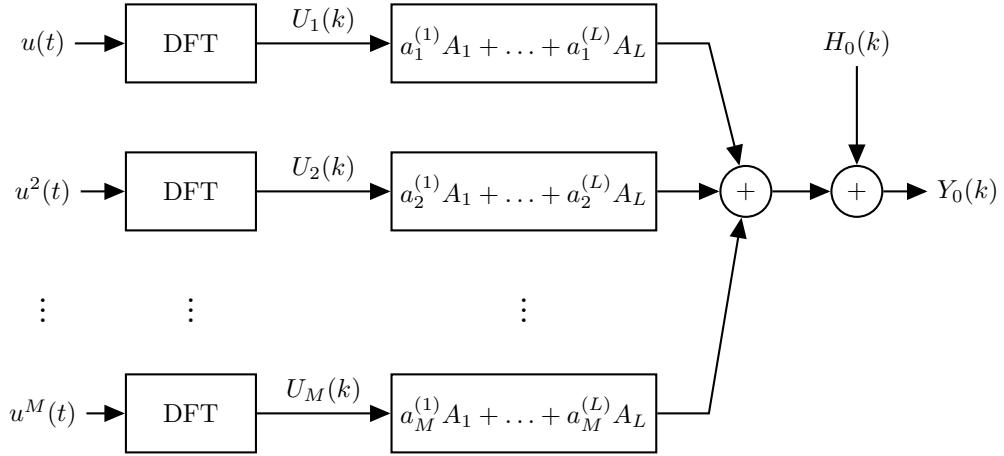


Figure 7.3: Alternate block structure for the parallel Hammerstein system

7.4 Identification using multi-level excitation

The introduction of the NOFRF model in [64] was accompanied by a data-driven identification scheme. The proposed method requires the system to be excited multiple times by scaled versions of a prototype input signal. Labelling the unscaled prototype input as $u^*(t)$, the set of applied input signals is given by,

$$\beta_i u^*(t), \quad i = 1, \dots, q, \quad \beta_i \in \mathbb{R}^+,$$

where $q \geq M$ is the total number of excitation levels used. The excitations will produce q corresponding output frequency responses which are denoted as $Y^{(q)}(k)$.

Given the model structure in (7.9), a regression model can be formed at each frequency bin, k , using the q excitations applied to the system. These models have the form,

$$\mathbf{Y}(k) = \mathbf{U}^*(k)\mathbf{G}(k) + E, \quad (7.11)$$

where

$$\mathbf{Y}(k) = \begin{bmatrix} Y^{(1)}(k) \\ \vdots \\ Y^{(q)}(k) \end{bmatrix}, \quad \mathbf{U}^*(k) = \begin{bmatrix} \beta_1 U_1^*(k) & \dots & \beta_1^M U_M^*(k) \\ \vdots & \ddots & \vdots \\ \beta_q U_1^*(k) & \dots & \beta_q^M U_M^*(k) \end{bmatrix}, \quad \mathbf{G}(k) = \begin{bmatrix} G_1(k) \\ \vdots \\ G_M(k) \end{bmatrix}, \quad (7.12)$$

and U_1^*, \dots, U_M^* are the nonlinear spectral quantities in (7.8) corresponding to $u^*(t)$.

Least squares estimates for the NOFRFs can be obtained analytically from (7.11)

at any given frequency, i.e.

$$\hat{\mathbf{G}}(k) = \left([\mathbf{U}^*(k)]^T [\mathbf{U}^*(k)] \right)^{-1} [\mathbf{U}^*(k)]^T \mathbf{Y}(k) \quad \forall k \in \mathcal{K} \quad (7.13)$$

where \mathcal{K} is the set of excited frequency bins in the input, $U_1^*(k)$.

This multi-level excitation method for NOFRF estimation has several disadvantages:

- Data collection may take a long time to perform, requiring at least M experiments which must reach steady-state before measuring.
- The method cannot be used in cases where the input is not precisely controlled.
- Least squares solutions are sensitive to measurement noise.
- The quantity and value of scaling parameters, β_i , are arbitrary choices which are made by the user.

These shortcomings highlight the need for a more sophisticated algorithm, capable of accurate parameter estimation in a generic and noisy experimental setting. In the case of parallel Hammerstein systems, such an algorithm will be developed in the sequel.

7.5 Gaussian process regression for NOFRFs

Taking inspiration from the linear case in [67], an estimation method can be developed from the Bayesian perspective. The NOFRF quantities will be modeled as real/complex parameter vectors with Gaussian prior covariances and estimated using empirical Bayes techniques. To achieve this, some assumptions and notation must first be clarified.

Assumption 7.7 (Parallel Hammerstein structure). *The system of interest can be represented by a parallel Hammerstein structure with polynomial nonlinearities of known maximum degree, M . Hence, there exist nonzero NOFRFs up to the M^{th} nonlinear order.*

Remark 7.8. *In the case where M is unknown, or the nonlinearities are not polynomial in nature, there are many established order selection methods, e.g. cross-validation, which can be applied to select an appropriate M .*

Notation 7.9. *Let \mathbf{k} be the vector of DFT bins, k , for which the input spectrum $U(k)$ is nonzero due to excitation. The corresponding parameter vector to be estimated for the m^{th} order NOFRF will be denoted $G_m(\mathbf{k})$.*

Remark 7.10. In the parallel Hammerstein case, all NOFRFs $G_m(k)$ will only be strictly real for $k = 0$, since they behave like linear filters. The same is true for the measured output, $Y(k)$.

Assumption 7.11 (Independent Gaussian NOFRFs). Assume that each NOFRF parameter vector is a RCG process with zero mean, i.e.

$$\begin{aligned} G_m(\mathbf{k}) &\sim \mathcal{RCN}(0, c_{G,m}\Sigma_m), \quad m = 1, \dots, M, \\ H_0(0) &\sim \mathcal{RCN}(0, c_{G,0}), \end{aligned} \tag{7.14}$$

where $c_{G,m} \in \mathbb{R}^+$ are normalization hyperparameters, and Σ_m are augmented covariances constructed from the covariance and relation matrices, R_m , Q_m , K_m and C_m as in (7.3). Furthermore, assume that all NOFRFs are independent of each other.

Recalling that parallel Hammerstein NOFRFs can be interpreted as linear filters, we are free to use the same covariance structures for Σ_m as were used in [67] for FRF estimation. In this chapter we consider the frequency domain equivalent of the DC structure in (2.38), whose form is provided in [67]¹. More specifically, the covariance and relation matrices for the RCG process $G_m(\mathbf{k})$ can be constructed using,

$$\mathbf{E}\{G_m(k_x)\overline{G_m(k_y)}\} = \frac{1}{\lambda_m + j(k_x - k_y)} \left(\frac{1}{\rho_m + \lambda_m/2 + jk_x} + \frac{1}{\rho_m + \lambda_m/2 - jk_y} \right), \tag{7.15}$$

$$\mathbf{E}\{G_m(k_x)G_m(k_y)\} = \frac{1}{\lambda_m + j(k_x + k_y)} \left(\frac{1}{\rho_m + \lambda_m/2 + jk_x} + \frac{1}{\rho_m + \lambda_m/2 + jk_y} \right), \tag{7.16}$$

where λ_m and ρ_m are the DC hyperparameters which require tuning.

7.5.1 Deriving the MAP estimate

Assumption 7.12 (No strictly real components). For simplicity and mathematical convenience, the remainder of this chapter will assume that the zero frequency bin is not excited, i.e. $0 \notin \mathbf{k}$. As noted in Remark 7.10, this implies that there are no strictly real components in the vectors $G_m(\mathbf{k})$ and $Y(\mathbf{k})$. Consequently, the real/complex normal framework in Section 7.2 will reduce to the complex normal framework, i.e. $X^\mathbb{R}$ is an empty vector in (7.2).

In order to derive the Bayesian posterior estimates for each NOFRF, an expression for the output spectrum distribution is first developed.

¹The transformation of time domain prior covariance structures to the frequency domain will be discussed in more detail in Chapter 8

Theorem 7.13. *For the NOFRF model defined in (7.8) and (7.9) with G_m given by (7.14), the vectorized output spectrum, $Y(\mathbf{k})$, is distributed as,*

$$Y(\mathbf{k}) \sim \mathcal{RCN}(0, \Sigma_Y),$$

where $\Sigma_Y = \sum_{m=1}^M \Sigma_{Y,m} + \sigma^2 I$,

and $\Sigma_{Y,m} = c_{G,m} (\widetilde{U_m(\mathbf{k})} \widetilde{U_m(\mathbf{k})}^H) \circ \Sigma_m$.

(7.17)

Proof. Noting that (7.8) implies $Y_m(\mathbf{k}) = G_m(\mathbf{k}) \circ U_m(\mathbf{k})$, the result follows directly from Assumption 7.11 and Properties 7.4 and 7.5 \square

The NOFRF parameter vectors are now jointly distributed with $Y(\mathbf{k})$, and the Bayesian framework allows computation of maximum a posteriori (MAP) estimates for each G_m .

Theorem 7.14. *Under Assumptions 7.7 and 7.11, the MAP estimate of $G_m(\mathbf{k})$ is given by,*

$$\widehat{G_m(\mathbf{k})} = c_{G,m} \Sigma_m \text{diag}(\widetilde{U_m(\mathbf{k})}^H) \Sigma_Y^{-1} \widetilde{Y(\mathbf{k})}, \quad (7.18)$$

where $\text{diag}(X)$ denotes a diagonal matrix with diagonal X .

Proof. Acknowledging the independence of each NOFRF, the m^{th} order NOFRF and output spectrum are jointly distributed as,

$$\begin{bmatrix} G_m(\mathbf{k}) \\ Y(\mathbf{k}) \end{bmatrix} \sim \mathcal{RCN}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_m & \Sigma_{G_m Y} \\ \Sigma_{Y G_m} & \Sigma_Y \end{bmatrix}\right), \quad (7.19)$$

where the joint covariance is computed as,

$$\begin{aligned} \Sigma_{G_m Y} &= \mathbf{E}\left\{\widetilde{G_m(\mathbf{k})} \widetilde{Y(\mathbf{k})}^H\right\} \\ &= \mathbf{E}\left\{\widetilde{G_m(\mathbf{k})} \left[\widetilde{G_m(\mathbf{k})}^H \circ \widetilde{U_m(\mathbf{k})}^H\right]\right\} \\ &= \mathbf{E}\left\{\widetilde{G_m(\mathbf{k})} \widetilde{G_m(\mathbf{k})}^H\right\} \cdot \text{diag}(\widetilde{U_m(\mathbf{k})}^H) \\ &= c_{G,m} \Sigma_m \text{diag}(\widetilde{U_m(\mathbf{k})}^H), \end{aligned} \quad (7.20)$$

using (7.8), (7.9) and Assumption 7.11. Now the MAP estimate will be the mean of the conditional distribution, $G_m(\mathbf{k})|Y(\mathbf{k})$, which is given by Property 7.6 as

$$\mathbf{E}\left\{\widetilde{G_m(\mathbf{k})}| \widetilde{Y(\mathbf{k})}\right\} = \Sigma_{G_m Y} \Sigma_Y^{-1} \widetilde{Y(\mathbf{k})}. \quad (7.21)$$

Combining (7.21) with (7.20) yields the result in (7.18). \square

7.5.2 Hyperparameter tuning

As in all Bayesian estimation methods discussed in this thesis, the hyperparameters describing the covariance of each NOFRF must be empirically tuned prior to estimation. For an M^{th} order model, the DC hyperparameter vector will be

$$\eta = [\sigma^2 \ \eta_1 \ \dots \ \eta_M],$$

where $\eta_m = [c_{G,m} \ \lambda_m \ \rho_m]$.

In this chapter, as in [67], the tuning process is achieved by maximizing the marginal log likelihood of the hyperparameters with respect to the observed output, i.e.

$$\hat{\eta} = \arg \min_{\eta} \widetilde{Y(\mathbf{k})}^H \Sigma_Y^{-1}(\eta) \widetilde{Y(\mathbf{k})} + \log \det \Sigma_Y(\eta). \quad (7.22)$$

7.6 GPR estimation in the presence of transients

Unlike the traditional multi-level excitation method, the Gaussian process regression approach can be modified to remove the effect of transient functions in estimation, allowing identification of NOFRFs in the case of non-periodic or non-steady-state measurement data.

It was shown in [67] that in the linear case, transient functions result from the difference $f(t) = u(t) - u(t + N)$ for $-\infty < t < 0$. Considering now the NOFRF case and observing Figure 7.1, it is clear that for a non-periodic $u(t)$, $u^m(t)$ will also be non-periodic, and there will be a transient function at each nonlinear order. We can use this information to update the model equation in (7.8) as follows:

$$Y_0(k) = H_0(k) + \sum_{m=1}^M G_m(k)U_m(k) + T_m(k), \quad (7.23)$$

where $T_m(k)$ is the m^{th} order transient function in the frequency domain.

In general, it is not obvious what kind of Gaussian prior is appropriate for a transient function. However, for the case of a zero mean white noise input it has been shown in [67] that for linear systems, the prior covariance of the transient function is approximately proportional to the system covariance. This result also extends to the NOFRF case for parallel Hammerstein systems, as can be confirmed by observing Figure 7.1. Whenever $u(t)$ is white noise we will have, in each branch, a transient produced by non-periodic white noise $u^m(t)$ passing through a linear filter G_m . Consequently, the following Gaussian assumptions will be placed on the transient vectors.

Assumption 7.15. *Given a white noise input, the distribution of the m^{th} order*

transient vector is given by,

$$T_m(\mathbf{k}) \sim \mathcal{RCN}(0, c_{T,m} \Sigma_m), \quad (7.24)$$

where $c_{T,m} \in \mathbb{R}^+$ is a normalization hyperparameter, and Σ_m is the augmented covariance of NOFRF vector $G_m(\mathbf{k})$.

Assumption 7.16. *The transient vectors at each nonlinear order, m , are independent of each other and all NOFRF vectors.*

The distribution of the output spectrum can now be updated accordingly, giving

$$\begin{aligned} \Sigma_Y &= \sum_{m=1}^M \Sigma_{Y,m} + \sigma^2 I, \\ \Sigma_{Y,m} &= c_{G,m} (\widetilde{U_m(\mathbf{k})} \widetilde{U_m(\mathbf{k})}^H) \circ \Sigma_m + c_{T,m} \Sigma_m. \end{aligned} \quad (7.25)$$

Hyperparameter tuning can still be performed using (7.22), with additional hyperparameters in the optimization, i.e.

$$\eta_m = [c_{G,m} \ c_{T,m} \ \lambda_m \ \rho_m].$$

Finally, the MAP estimates for each $G_m(\mathbf{k})$ will remain as in Theorem 7.14, using the updated definition of Σ_Y in (7.25). Furthermore, MAP estimates of the transient functions can also be obtained using the same logic, yielding

$$\widehat{T_m(\mathbf{k})} = c_{T,m} \Sigma_m \Sigma_Y^{-1} \widetilde{Y(\mathbf{k})}. \quad (7.26)$$

7.7 GPR estimation using multiple input realizations

If a set of NOFRFs are estimated using only *one* input realization, as in (7.18), then the estimation problem is inevitably underdetermined. This is because for every measurement, $Y(k)$, in the output spectrum, we must estimate the value of multiple NOFRFs at that same frequency. The GPR framework is flexible enough to handle this case and produce reasonable estimates, however the accuracy of the estimates will improve if the ratio of measurement data to model parameters is increased.

In frequency domain estimation problems, the typical method for increasing the amount of measurement data is to perform multiple experiments, each with a different input realization. Indeed, this concept was required for the multi-level excitation method in Section 7.4, where each input realization was a scaled version of some prototype signal. In this section we will present an extension to the GPR method developed in Section 7.5, which considers multiple input/output realizations in the estimation.

Assumption 7.17. *There are q separate realizations of transient-free input/output measurement data available, with a shared vector, \mathbf{k} , of excited DFT frequencies.*

Notation 7.18. *The i^{th} realization has input and output spectrum measurements given by $Y^{(i)}(\mathbf{k})$ and $U^{(i)}(\mathbf{k})$ respectively. The associated vectors containing all realizations are denoted by*

$$\mathbf{Y}(\mathbf{k}) = \begin{bmatrix} Y^{(1)}(\mathbf{k}) \\ \vdots \\ Y^{(q)}(\mathbf{k}) \end{bmatrix} \text{ and } \mathbf{U}_m(\mathbf{k}) = \begin{bmatrix} U_m^{(1)}(\mathbf{k}) \\ \vdots \\ U_m^{(q)}(\mathbf{k}) \end{bmatrix} \text{ for } m = 1, \dots, M. \quad (7.27)$$

Noting Assumptions 7.12 and 7.17 as well as Notation 7.18, and using the NOFRF model (7.8), the model equation for multiple realizations can be written as

$$\mathbf{Y}(\mathbf{k}) = \sum_{m=1}^M \underbrace{\begin{bmatrix} G_m(\mathbf{k}) & \dots & G_m(\mathbf{k}) \end{bmatrix}}_{q \text{ copies}}^T \circ \mathbf{U}_m(\mathbf{k}). \quad (7.28)$$

Now with the new model structure (7.28) and recalling Assumption 7.11, the GPR method in Section 7.5 can be applied with the following modifications.

The distribution of $\mathbf{Y}(\mathbf{k})$ is given by

$$\begin{aligned} \mathbf{Y}(\mathbf{k}) &\sim \mathcal{RCN}(0, \Sigma_Y), \\ \text{where } \Sigma_Y &= \sum_{m=1}^M \Sigma_{Y,m} + \sigma^2 I, \\ \Sigma_{Y,m} &= c_{G,m} (\widetilde{\mathbf{U}_m(\mathbf{k})} \widetilde{\mathbf{U}_m(\mathbf{k})}^H) \circ \Sigma_m. \end{aligned} \quad (7.29)$$

Recall the covariance and relation matrices, K_m and C_m , of Σ_m , which can be constructed using the transformed DC structure (7.15) and (7.16). The matrix Σ_m is constructed using its own covariance and relation matrices, \mathbf{K}_m and \mathbf{C}_m , whose form is easily found as

$$\mathbf{K}_m = \underbrace{\begin{bmatrix} K_m & \dots & K_m \\ \vdots & \ddots & \vdots \\ K_m & \dots & K_m \end{bmatrix}}_{q \times q \text{ copies}}, \quad \mathbf{C}_m = \underbrace{\begin{bmatrix} C_m & \dots & C_m \\ \vdots & \ddots & \vdots \\ C_m & \dots & C_m \end{bmatrix}}_{q \times q \text{ copies}}. \quad (7.30)$$

The MAP estimate of $G_m(\mathbf{k})$ now becomes

$$\widehat{G_m}(\mathbf{k}) = c_{G,m} \underbrace{\begin{bmatrix} K_m & \dots & K_m & C_m & \dots & C_m \\ C_m^H & \dots & C_m^H & \overline{K_m} & \dots & \overline{K_m} \end{bmatrix}}_{q \text{ copies of each}} \text{diag} (\widetilde{\mathbf{U}_m(\mathbf{k})}^H) \Sigma_Y^{-1} \widetilde{\mathbf{Y}(\mathbf{k})}, \quad (7.31)$$

and hyperparameter tuning is achieved using

$$\hat{\eta} = \arg \min_{\eta} \widetilde{\mathbf{Y}(\mathbf{k})}^H \Sigma_Y^{-1}(\eta) \widetilde{\mathbf{Y}(\mathbf{k})} + \log \det \Sigma_Y(\eta). \quad (7.32)$$

7.8 Simulation examples

Numerical simulations are performed in this section to compare the performance of the proposed GPR method, which we will refer to as *regularized*, against the *traditional* method of multi-level excitation. Using a Monte Carlo approach, the methods are compared for varying noise levels and number of NOFRFs.

The GPR method is also applied to simulated experimental data which is not in steady-state. In these examples, output transient functions are estimated via the proposed Bayesian framework and compared against the true observed transient.

7.8.1 Simulation settings

All simulations use the parallel Hammerstein structure shown in Figure 7.4, which is intentionally constructed such that each parallel branch contains only one exponent of the input in its nonlinear block. With such a structure, the NOFRFs are directly equal to the linear filter in their corresponding branch. Note that each branch can be switched in and out of the system depending on the experiment settings.

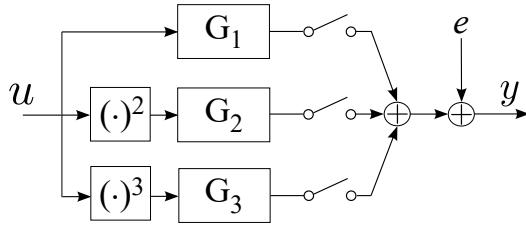


Figure 7.4: Parallel Hammerstein structure used for numerical examples

In the simulation, the linear filters (and therefore the NOFRFs), G_1 , G_2 and G_3 , are constructed as Chebyshev filters with resonant modes at normalized frequencies 0.072, 0.111 and 0.091 respectively. The inputs in any given experiment are random-phase multisines which are periodic in $N = 128$ samples and have no zero frequency component. The band of excitation, \mathbf{k} , for the multisines is constructed so as to avoid aliasing in the output spectrum. The output error, e , is a Gaussian white

noise vector added in each experiment to provide the required signal-to-noise ratio (SNR).

7.8.2 Comparison of traditional and GPR methods

Four 1000-run Monte Carlo studies are performed on the traditional and regularized estimation methods to determine their relative accuracy. The studies are:

1. $M = 2$ (G_1 and G_2 branches) with SNR = 40dB. 1 input/output realization for the *regularized* method. 2 input scalings for the *traditional* method.
2. $M = 2$ (G_1 and G_2 branches) with SNR = 20dB. 1 input/output realization for the *regularized* method. 2 input scalings for the *traditional* method.
3. $M = 3$ (all branches) with SNR = 40dB. 1 input/output realization for the *regularized* method. 3 input scalings for the *traditional* method.
4. $M = 3$ (all branches) with SNR = 40dB. 10 input/output realizations for the *regularized* method. 10 input scalings for the *traditional* method.

Note that for studies 1, 2 and 3, the traditional multi-level excitation method uses its minimum requirement of M differently scaled inputs, and hence M experiments. In contrast, the regularized method is disadvantaged in these studies by using only one input and experiment, such that data collection is faster by a factor of M . Furthermore, the estimation is performed on a dataset which is smaller by a factor of M for the GPR method. The final study (4) compares the two methods for an equal amount of measurement data, using the results in Section 7.7 for the regularized method. For a fair comparison of the two methods, all studies use measurement data taken from the steady-state portion of the experiment, i.e. there are no transient components in the output response.

To visualize performance in each Monte Carlo study, Figures 7.5, 7.6, 7.7 and 7.8 plot the true NOFRF magnitudes as well as the central 95% intervals (shaded) from each estimation method. For studies 1, 2 and 3, the regularized method produces tighter intervals which are more centered around the true NOFRF, despite having a shorter experiment time and using significantly less data in the estimation. The contrast in performance is even clearer in the results for study 4, where the regularized method is allowed to use an equal amount of measurement data. All of the Monte Carlo results show a clear accuracy benefit for the proposed method when output measurement noise is present in the system, which can be attributed to the ability of GPR methods to significantly reduce estimation variance at the price of a small bias [95]. The bias here is small but evident, particularly at the resonance peaks of each NOFRF.

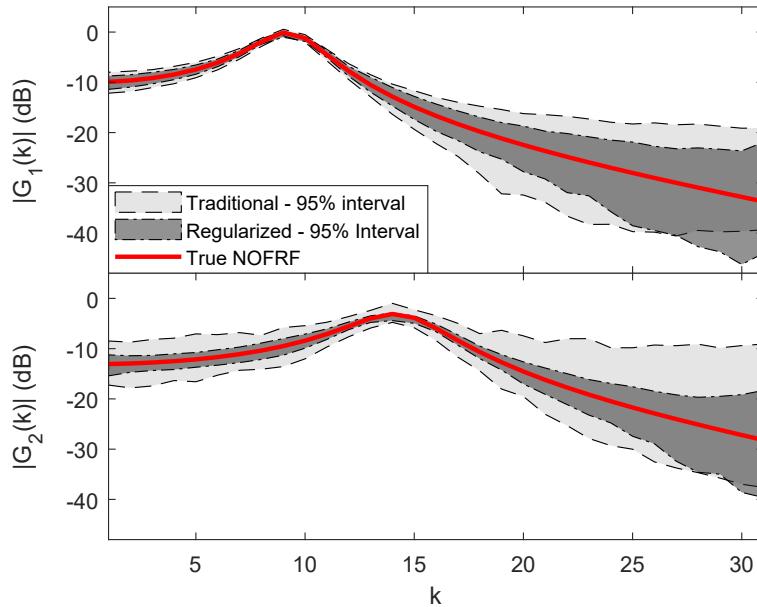


Figure 7.5: Magnitude intervals for study 1, $M = 2$ and SNR=40dB

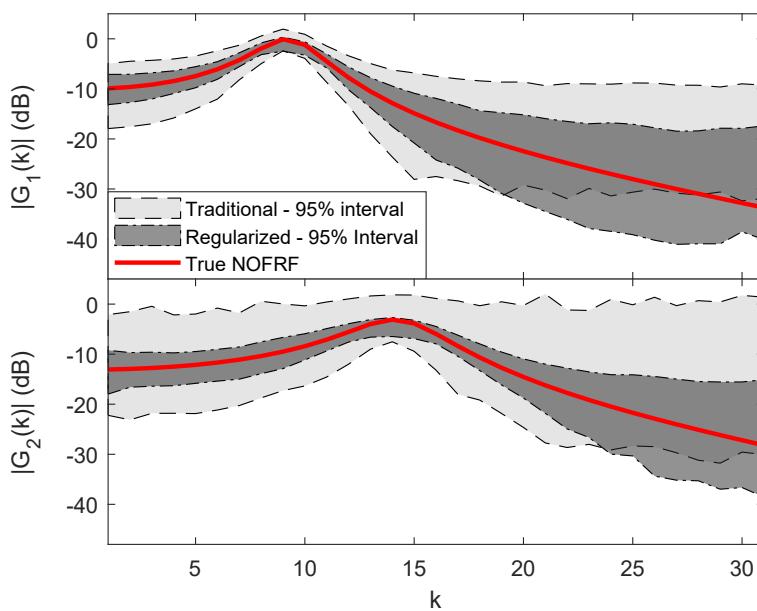


Figure 7.6: Magnitude intervals for study 2, $M = 2$ and SNR=20dB

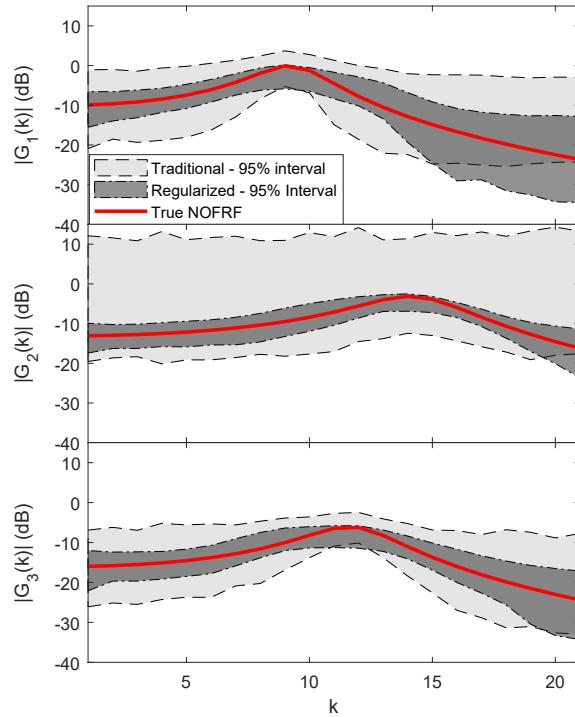


Figure 7.7: Magnitude intervals for study 3, $M = 3$ and SNR=40dB

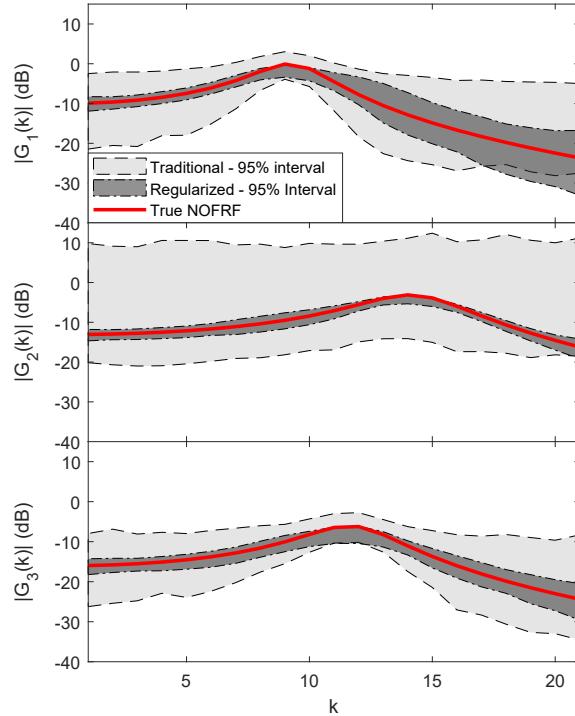


Figure 7.8: Magnitude intervals for study 4, $M = 3$ and SNR=40dB (equal data)

7.8.3 Transient estimation test

While the traditional multi-level excitation method cannot estimate or remove the effect of transients, it was shown in Section 7.6 that the proposed approach can be modified to consider transient functions in the Gaussian process regression. This allows us to simultaneously estimate the NOFRFs and their corresponding transients, thereby increasing the accuracy of estimation for non-periodic or non steady-state data.

To demonstrate this capability, the modified method in Section 7.6 was applied to the simulated system of Figure 7.4, using the G_1 and G_2 branches and no output noise. The input multisine was applied for a single period and preceded by zero initial conditions, producing a transient in the output response. Using this input/output data, the total transient could be estimated by summing the transient estimates from each nonlinear order, i.e.

$$T(k) = \sum_{m=1}^M T_m(k). \quad (7.33)$$

The true and estimated transients are plotted in Figure 7.9 for three input realizations, showing reasonable estimation performance which significantly reduces the effect of the transients on NOFRF estimation accuracy.

7.9 Conclusion

This chapter adapts the Bayesian techniques from linear identification theory to the problem of estimating parallel Hammerstein systems in the frequency domain. First, the NOFRF model structure was shown to be an intuitive and useful tool in estimating such systems, since the frequency function at each nonlinear order will be input-independent and pseudo-linear for this case. In developing a GPR method, the Gaussian prior covariances for each frequency function could then be chosen and tuned according to results obtained for linear FRF estimation in [67]. The proposed method was extended to consider concurrent transient estimation in the case of non-steady-state data. When compared to the traditional NOFRF estimation method of multi-level excitation, the proposed method has fewer experimental constraints and can achieve higher accuracy using less data when output noise is present.

In order to estimate frequency domain models for a broader class of nonlinear systems, the NOFRF method proposed in this chapter would not be sufficient. This is because the NOFRFs are input-dependent in general, and thus we cannot always impose prior covariance structures borrowed from the linear theory. An alternate option is to move to the multidimensional GFRF model structure, and design an analogous method to the regularized Volterra kernel estimation outlined in Chapter 3.6. Such an approach will be considered in Chapter 8.

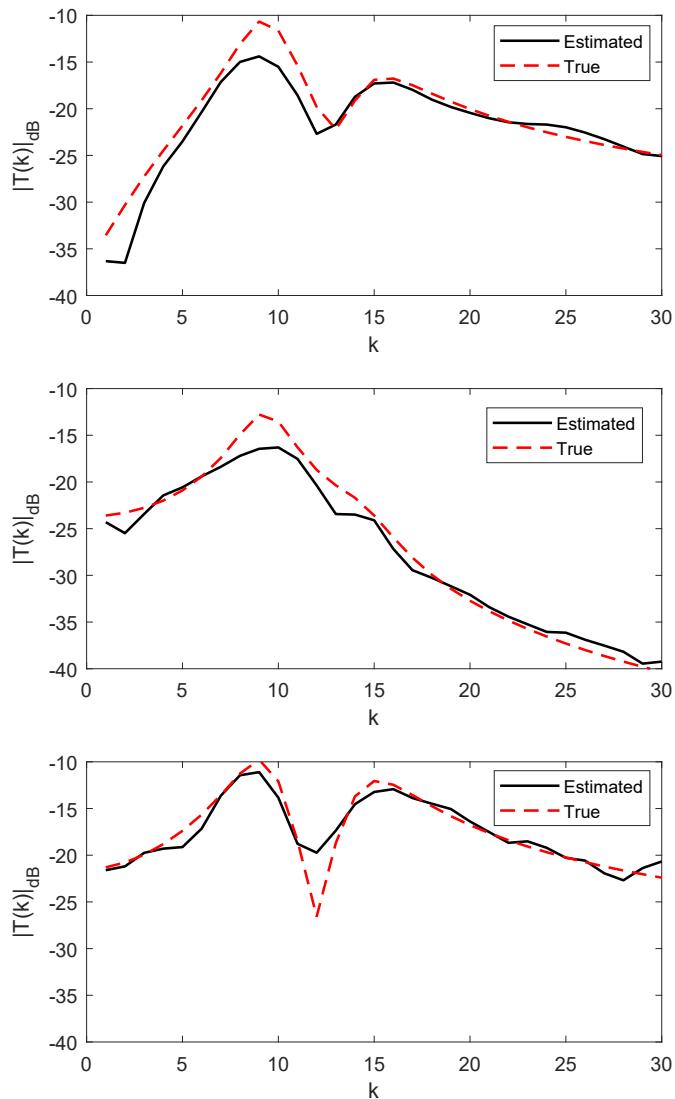


Figure 7.9: True and estimated transient functions for three input realizations of the test system ($M = 2$)

CHAPTER

8

Gaussian Process Regression for the Estimation of Generalized Frequency Response Functions

The generalized frequency response function (GFRF) model is the most direct frequency domain equivalent to the time domain Volterra series, where each GFRF is a multidimensional Fourier transform of the corresponding Volterra kernel. This chapter develops a Gaussian process regression (GPR) method for data-driven identification of GFRF models. Each GFRF is modeled as a real/complex Gaussian process with prior covariance related to the time domain characteristics of its corresponding Volterra kernel, such that the method is a frequency domain equivalent of the Bayesian regularization method considered in Part II. Numerical examples highlight the benefit, when only a limited frequency band is of interest, of using the proposed direct approach over indirect estimation from a time domain model.

8.1 Introduction

In Part III of this thesis, we are considering Bayesian estimation of Volterra series models in the frequency domain. While Chapter 7 revealed that Gaussian process regression (GPR) can be applied to the NOFRF model, the results were limited to the case of a parallel Hammerstein nonlinear system, as this is the case where the NOFRFs become input independent and thus easy to model in a Bayesian sense. In order to make Bayesian frequency domain estimation more broadly applicable, it is necessary to move to the generalized frequency response function (GFRF) model, where each complex-valued GFRF in the series is related to its corresponding time domain kernel through a multidimensional Fourier transform.

The estimation of GFRFs has been extensively studied in the literature. In one case, an interpolation method for the second order term was developed [85], which requires multiple input realizations to give an overdetermined problem. Other studies utilized excitations with specifically selected harmonics, however such methods become prohibitively complex for high nonlinear orders, and there is a limit to the number of GFRF elements which can be estimated [17], [39]. Alternatively, full GFRF estimates can be obtained from arbitrary excitation by first identifying the time domain Volterra kernels and then transforming to the frequency domain [130]. In the case where a limited frequency band is of interest, then identification directly in the frequency domain is more beneficial with respect to the number of estimated parameters.

The contribution of this chapter is to develop a frequency domain GFRF identification method using GPR, and thus further expand the capabilities of nonparametric Bayesian estimation. As in previous chapters, concepts from [11] and [67] will serve as inspiration in developing the method. In particular, the real/complex Gaussian framework described in Chapter 7.2 is utilized again, and the GFRFs are modeled using prior covariances translated from the time domain functions designed in [11].

The proposed approach can obtain GFRF estimates up to an arbitrary nonlinear order using a single multisine input excitation, even though the estimation problem is severely rank deficient. For cases where only a limited frequency band is of interest, the proposed method presents a clear advantage over indirect time domain GPR, which is a property that was also observed for the linear case in [67].

8.2 Problem formulation

For convenience, the relevant time domain and frequency domain Volterra models will be recalled here for reference throughout the chapter. The original time domain

series is given by

$$\begin{aligned} y^0(t) &= h_0 + \sum_{m=1}^M y_m(t), \\ y_m(t) &= \sum_{\tau_1=0}^{n_m-1} \dots \sum_{\tau_m=0}^{n_m-1} h_m(\tau_1, \dots, \tau_m) \prod_{\tau=\tau_1}^{\tau_m} u(t-\tau). \end{aligned} \quad (8.1)$$

The GFRF model then relates the (steady-state) input and output spectrum in the frequency domain, i.e.

$$\begin{aligned} Y^0(k) &= H_0(k) + \sum_{m=1}^M Y_m(k), \\ Y_m(k) &= \frac{1}{N^{m-1}} \sum_{k_1+\dots+k_m=k} H_m(k_1, \dots, k_m) \prod_{i=1}^m U(k_i), \end{aligned} \quad (8.2)$$

where the GFRFs, H_m , are obtained via multidimensional Fourier transforms on the corresponding Volterra kernels, h_m , using

$$H_m(k_1, \dots, k_m) = \sum_{\tau_1=0}^{n_m-1} \dots \sum_{\tau_m=0}^{n_m-1} h_m(\tau_1, \dots, \tau_m) e^{-j2\pi k_1 \tau_1} \dots e^{-j2\pi k_m \tau_m}. \quad (8.3)$$

The nonlinear system identification problem is now formulated in the frequency domain under steady-state conditions, using the following assumptions and notation:

Assumption 8.1. *The measured output, $\{y(t)\}_{t=0}^{N-1}$, is a Volterra output corrupted by white noise, i.e.*

$$y(t) = y^0(t) + e(t); \quad e(t) \sim \mathcal{N}(0, \sigma^2), \quad (8.4)$$

where $y^0(t)$ is the noise-free Volterra output from (8.1).

Assumption 8.2 (Steady-state). *The measured input $\{u(t)\}_{t=0}^{N-1}$ is one period of an N -periodic sequence which has been applied to the system for a sufficiently long time such that the system is in steady-state with a corresponding N -periodic output response, $\{y^0(t)\}_{t=0}^{N-1}$, whose spectrum can be given by the steady-state GFRF model (8.2).*

Notation 8.3. *The set of DFT indices, k , for which the input spectrum, $\{U(k) : k \in \mathbb{Z}, 0 \leq k \leq N/2\}$, is non-zero is denoted by \mathbf{k}_u . Likewise, the set of indices for which the output spectrum, $\{Y(k) : k \in \mathbb{Z}, 0 \leq k \leq N/2\}$, is non-zero (due to input excitation) is denoted by \mathbf{k}_y (see [63] for the precise definition of this set for a given $U(k)$ and M).*

Notation 8.4. *The frequencies of interest for estimation are given by the set*

$$\mathbf{k}_e = -\mathbf{k}_u \cup \mathbf{k}_u. \quad (8.5)$$

Given Assumptions 8.1 and 8.2, the goal is to estimate the GFRF elements, H_0 and $H_m(k_1, \dots, k_m) \forall k_1, \dots, k_m \in \mathbf{k}_e$ and $m = 1, \dots, M$.

8.3 Gaussian process regression for GFRFs

Given the assumptions and problem formulation described in Section 8.2, a GPR method is developed in this section. The hybrid real/complex normal distribution will be required for this purpose, since every GFRF contains both complex *and* strictly real components. Thus, the notation and properties introduced in Chapter 7.2 will also be used in this chapter.

8.3.1 GFRFs as real/complex Gaussian vectors

Any set of GFRF parameters can be arranged into a real/complex vector, as outlined in the following definition.

Definition 8.5. *The m -dimensional array containing $H_m(k_1, \dots, k_m), \forall k_1, \dots, k_m \in \mathbf{k}_e$ has a real/complex vectorized form (shown in (7.1)) which will be denoted by*

$$H_m^{\mathcal{V}} = \begin{bmatrix} H_m^{\mathbb{R}} \\ H_m^{\mathbb{C}} \end{bmatrix}, \quad (8.6)$$

where $H_m^{\mathbb{R}}$ contains the strictly real components.

Unlike in [67] and Chapter 7, where the one-dimensional frequency functions are strictly real only at zero frequency, higher order GFRFs originating from symmetric Volterra kernels present a larger set of real components. Since we are required to express the vectorized GFRFs in the form dictated by (8.6), it is important to identify the location of all strictly real elements, as addressed in the following theorem.

Theorem 8.6. *Consider a symmetric m^{th} order Volterra kernel, $h_m(\tau_1, \dots, \tau_m)$, as defined in (8.1), and its corresponding GFRF, $H_m(k_1, \dots, k_m)$, as defined in (8.3). The strictly real components of H_m are defined by the set of DFT indices,*

$$\mathbf{k}_m^{\mathbb{R}} = \{k_1, \dots, k_m \in \mathbb{Z} : \sum_{\mathcal{I} \in S_m(X)} \sin\left(\frac{2\pi}{N}(k_1\tau_{i_1} + \dots + k_m\tau_{i_m})\right) = 0 \quad \forall \tau_i \in \mathbb{N}\},$$

where $\mathcal{I} = [i_1, \dots, i_m]$, and $S_m(X)$ denotes the set of all permutations of the vector $X = [1, 2, \dots, m]$.

Proof. Taking the imaginary part of (8.3), we obtain

$$\Im\{H_m(k_1, \dots, k_m)\} = \sum_{\tau_1=0}^{n_m-1} \dots \sum_{\tau_m=0}^{n_m-1} h_m(\tau_1, \dots, \tau_m) \sin\left(\frac{2\pi}{N}(k_1\tau_1 + \dots + k_m\tau_m)\right). \quad (8.7)$$

For a symmetric kernel h_m , the Fourier transformation will be independent of the ordering of the lags, τ_1, \dots, τ_m , so it is equivalent to take an average over all lag permutations,

$$\begin{aligned} & \Im\{H_m(k_1, \dots, k_m)\} \\ &= \frac{1}{m!} \sum_{\mathcal{I} \in S_m(X)} \left[\sum_{\tau_1=0}^{n_m-1} \dots \sum_{\tau_m=0}^{n_m-1} h_m(\tau_{i_1}, \dots, \tau_{i_m}) \sin\left(\frac{2\pi}{N}(k_1\tau_{i_1} + \dots + k_m\tau_{i_m})\right) \right] \end{aligned} \quad (8.8)$$

$$= \frac{1}{m!} \sum_{\tau_1=0}^{n_m-1} \dots \sum_{\tau_m=0}^{n_m-1} h_m(\tau_1, \dots, \tau_m) \sum_{\mathcal{I} \in S_m(X)} \sin\left(\frac{2\pi}{N}(k_1\tau_{i_1} + \dots + k_m\tau_{i_m})\right). \quad (8.9)$$

Thus, for an arbitrary symmetric h_m , strictly real elements of H_m are guaranteed only when

$$\sum_{\mathcal{I} \in S_m(X)} \sin\left(\frac{2\pi}{N}(k_1\tau_{i_1} + \dots + k_m\tau_{i_m})\right) = 0 \quad \forall \tau_i \in \mathbb{N}$$

as required. \square

To explicitly locate the strictly real elements within the estimation set, \mathbf{k}_e , the result in Theorem 8.6 can be elaborated using the antisymmetry of $\sin()$, as shown below for the first three nonlinear orders.

$$\mathbf{k}_1^{\mathbb{R}} = \{k_1 \in \mathbf{k}_e : k_1 = 0\}, \quad (8.10)$$

$$\mathbf{k}_2^{\mathbb{R}} = \{k_1, k_2 \in \mathbf{k}_e : k_1 = -k_2\}. \quad (8.11)$$

$$\begin{aligned} \mathbf{k}_3^{\mathbb{R}} = & \{k_1, k_2, k_3 \in \mathbf{k}_e : k_1 = -k_2, k_3 = 0\} \cup \\ & \{k_1, k_2, k_3 \in \mathbf{k}_e : k_1 = -k_3, k_2 = 0\} \cup \\ & \{k_1, k_2, k_3 \in \mathbf{k}_e : k_2 = -k_3, k_1 = 0\}. \end{aligned} \quad (8.12)$$

Note that the set for the first order GFRF, $\mathbf{k}_1^{\mathbb{R}}$, can only contain the zero frequency entry, as is the case for a linear FRF. The multidimensional GFRFs, however, possess larger sets in general.

8.3.2 The Gaussian assumption

The following Gaussian assumptions are placed on the vectorized GFRF quantities, $H_m^{\mathcal{V}}$.

Assumption 8.7. $H_m^{\mathcal{V}}$ is real/complex normally distributed with zero mean and augmented covariance Σ_m , i.e.

$$H_m^{\mathcal{V}} \sim \mathcal{RCN}(0, \Sigma_m), \quad (8.13)$$

where Σ_m is constructed from the covariance and relation functions R_m , Q_m , K_m and C_m as in (7.4).

Assumption 8.8. Two (real/complex vectorized) Gaussian GFRFs of different orders are independent, i.e. $H_i^{\mathcal{V}}$ and $H_j^{\mathcal{V}}$ are independent for $i \neq j$.

The assumptions 8.7 and 8.8 are consistent with those made in [11] and [67], as well as in previous chapters of this thesis. They will be used in the sequel to derive the output spectrum distribution and define the GPR procedure for GFRF estimation.

8.3.3 Prior covariance design

For multidimensional Volterra kernels, prior covariance functions have already been constructed in the time domain [11], [10], by applying a TC or DC structure along multiple perpendicular regularizing directions. The same approach was adopted for the time domain methods in this thesis, in Chapters 4, 5 and 6. The resulting covariance matrices are guaranteed to be valid and produce stable kernel realizations.

In order to construct a frequency domain analog to the Volterra kernel covariance function, we note that there exists a linear transformation between the vectorized kernel $h_m^{\mathcal{V}}$ and its frequency domain counterpart, $H_m^{\mathcal{V}}$, which will be denoted F_m , i.e.

$$H_m^{\mathcal{V}} = \begin{bmatrix} H_m^{\mathbb{R}} \\ H_m^{\mathbb{C}} \end{bmatrix} = \begin{bmatrix} F_m^{\mathbb{R}} \\ F_m^{\mathbb{C}} \end{bmatrix} h_m^{\mathcal{V}} = F_m h_m^{\mathcal{V}}, \quad (8.14)$$

where $F_m^{\mathbb{R}}$ produces $H_m^{\mathbb{R}}$ and $F_m^{\mathbb{C}}$ produces $H_m^{\mathbb{C}}$. While it is clear from (8.3) that F_m should contain appropriate products of exponentials, a more rigorous approach for obtaining the explicit form of F_m and its submatrices requires the following theorem.

Theorem 8.9. Consider the m -dimensional array $w_m \in \mathbb{R}^{N \times \dots \times N}$, and its m -dimensional DFT, $W_m \in \mathbb{C}^{N \times \dots \times N}$. Consider also their vectorized forms, $w_m^{\mathcal{V}} \in \mathbb{R}^{N^m}$ and $W_m^{\mathcal{V}} \in \mathbb{C}^{N^m}$, where vectorization is performed such that

$$\begin{aligned} w_m^{\mathcal{V}} = & [w_m(1, 1, \dots, 1), w_m(2, 1, \dots, 1), \dots, \\ & w_m(N, 1, \dots, 1), w_m(1, 2, \dots, 1), \dots, w_m(N, N, \dots, N)], \end{aligned}$$

and likewise for $W_m^{\mathcal{V}}$. The vectorized arrays are related by,

$$W_m^{\mathcal{V}} = \Psi_m w_m^{\mathcal{V}}, \quad (8.15)$$

where Ψ_m can be obtained from the recursive definition,

$$\Psi_p = \Psi_{p-1} \otimes \Psi_1, \quad p = 2, 3, \dots, m \quad (8.16)$$

and where Ψ_1 is the $N \times N$ DFT matrix.

Proof. For $m = 1$, by definition we have:

$$W_1^{\mathcal{V}} = \Psi_1 w_1^{\mathcal{V}} \quad (8.17)$$

For $m = 2$, the DFT is applied in each dimension separately:

$$W_2 = \Psi_1 w_2 \Psi_1^T \quad (8.18)$$

$$= \underbrace{[\Psi_1 \otimes \Psi_1]}_{\Psi_2} w_2^{\mathcal{V}}, \quad (8.19)$$

by the well known property of Kronecker products.

For tensor $W_m = \{W_{i_1, \dots, i_m}\}_{i_1, \dots, i_m=0}^{N-1}$, let the $m - 1$ dimension tensor $W_m^{(k)}$ be given by $\{W_{i_1, \dots, i_{m-1}, k}\}_{i_1, \dots, i_{m-1}=0}^{N-1}$.

Assuming $W_{p-1}^{\mathcal{V}} = \Psi_{p-1} w_{p-1}^{\mathcal{V}}$, then by induction:

$$[W_p^{(1)\mathcal{V}} \dots W_p^{(N)\mathcal{V}}] = [\Psi_{p-1} w_p^{(1)\mathcal{V}} \dots \Psi_{p-1} w_p^{(N)\mathcal{V}}] \Psi_1^T \quad (8.20)$$

$$= \Psi_{p-1} [w_p^{(1)\mathcal{V}} \dots w_p^{(N)\mathcal{V}}] \Psi_1^T \quad (8.21)$$

$$\implies W_p^{\mathcal{V}} = \underbrace{[\Psi_{p-1} \otimes \Psi_1]}_{\Psi_p} w_p^{\mathcal{V}} \quad (8.22)$$

again using the Kronecker product property. Thus, $\Psi_p = \Psi_{p-1} \otimes \Psi_1$ as required \square

Assuming $h_m^{\mathcal{V}}$ is vectorized as described in Theorem 8.9, we can form F_m by first computing the matrix Ψ_m , then rearranging and removing rows to correspond to a desired vectorization of $H_m^{\mathcal{V}}$ which satisfies the required form in (8.6).

The transformation can now be used to convert time domain covariance functions to the frequency domain,

$$\begin{aligned} R_m &= \mathbf{E}\{H_m^{\mathbb{R}} H_m^{\mathbb{R}T}\} = F_m^{\mathbb{R}} \mathbf{E}\{h_m^{\mathcal{V}} h_m^{\mathcal{V}T}\} F_m^{\mathbb{R}T} = F_m^{\mathbb{R}} P_m F_m^{\mathbb{R}T} \\ Q_m &= \mathbf{E}\{H_m^{\mathbb{R}} H_m^{\mathbb{C}H}\} = F_m^{\mathbb{R}} \mathbf{E}\{h_m^{\mathcal{V}} h_m^{\mathcal{V}T}\} F_m^{\mathbb{C}H} = F_m^{\mathbb{R}} P_m F_m^{\mathbb{C}H} \\ K_m &= \mathbf{E}\{H_m^{\mathbb{C}} H_m^{\mathbb{C}H}\} = F_m^{\mathbb{C}} \mathbf{E}\{h_m^{\mathcal{V}} h_m^{\mathcal{V}T}\} F_m^{\mathbb{C}H} = F_m^{\mathbb{C}} P_m F_m^{\mathbb{C}H} \\ C_m &= \mathbf{E}\{H_m^{\mathbb{C}} H_m^{\mathbb{C}T}\} = F_m^{\mathbb{C}} \mathbf{E}\{h_m^{\mathcal{V}} h_m^{\mathcal{V}T}\} F_m^{\mathbb{C}T} = F_m^{\mathbb{C}} P_m F_m^{\mathbb{C}T} \end{aligned} \quad (8.23)$$

where $P_m = \mathbf{E}\{h_m^{\mathcal{V}} h_m^{\mathcal{V}T}\}$ denotes the tunable TC/DC-based covariance structure designed in [11] and discussed in Chapter 3.6 of this thesis.

The full augmented covariance, Σ_m , for the m^{th} GFRF is constructed as in (7.4), i.e.

$$\Sigma_m = \begin{bmatrix} R_m & Q_m & \overline{Q_m} \\ Q_m^H & K_m & C_m \\ \overline{Q_m^H} & C_m^H & \overline{K_m} \end{bmatrix}. \quad (8.24)$$

Remark 8.10. In practice, symmetry must be enforced in both the time and frequency domain kernels to guarantee a unique Volterra series representation [109]. Thus, we are required to modify the transformation matrices, F_m , by removing the columns which correspond to redundant time domain parameters, and removing rows that correspond to redundant frequency domain parameters. As an example, see Figure 8.1 in Section 8.3.6 for the symmetry axes of a second order kernel and GFRF, which are used to dictate the location of unique and redundant parameters in those series terms.

8.3.4 Deriving the MAP estimate

First we must derive the output spectrum distribution for $Y(\mathbf{k}_y)$, which requires the restructuring of the model equation (8.2) into the regression form,

$$\begin{aligned} Y(\mathbf{k}_y) &= [\Phi_1 \dots \Phi_M] [H_1^{\mathcal{V}^T} \dots H_M^{\mathcal{V}^T}]^T + E(\mathbf{k}_y) \\ &= \Phi H + E, \end{aligned} \quad (8.25)$$

where E is the (assumed) i.i.d. error vector in the frequency domain, and Φ_m is an appropriate regressor containing the input spectrum products corresponding to $H_m^{\mathcal{V}}$. Note that symmetry should also be enforced in the GFRFs here, which should be reflected in the design of the regressors.

Now $Y(\mathbf{k}_y)$, being formed from the DFT of the real signal $y(t)$, will also be a real/complex vector in the case where $0 \in \mathbf{k}_y$. Thus, we extend (8.25) to the augmented output case, resulting in

$$\begin{aligned} \tilde{Y}(\mathbf{k}_y) &= [\widetilde{\Phi}_1 \dots \widetilde{\Phi}_M] [\widetilde{H}_1^{\mathcal{V}^T} \dots \widetilde{H}_M^{\mathcal{V}^T}]^T + \tilde{E}(\mathbf{k}_y) \\ &= \widetilde{\Phi} \widetilde{H} + \tilde{E}. \end{aligned} \quad (8.26)$$

The augmented regressors can be constructed as

$$\widetilde{\Phi}_m = \begin{bmatrix} \Phi_m^{\mathbb{R}} & \Phi_m^{\mathbb{RC}} & 0 & 0 \\ 0 & 0 & \Phi_m^{\mathbb{C}} & 0 \\ 0 & 0 & 0 & \overline{\Phi_m^{\mathbb{C}}} \end{bmatrix}, \quad (8.27)$$

where $\Phi_m^{\mathbb{R}}$ maps the contribution of $H_m^{\mathbb{R}}$ to $Y(0)$, and $\Phi_m^{\mathbb{RC}}$ maps the contribution to $Y(0)$ from complex GFRF components. Note that this formulation dictates (to some extent) the required ordering of the vector $H_m^{\mathcal{V}}$.

Remark 8.11. When Φ is constructed from a single input/output realization as in (8.25), the associated linear regression problem will be severely rank deficient for $M > 1$. While it is possible to make the problem overdetermined by ‘stacking’ a sufficient number of unique input/output realizations (similarly to Chapter 7.7), we

will focus on the rank deficient case in this chapter to highlight the flexibility of GPR.

From (8.26), the output spectrum distribution is derived as follows.

Theorem 8.12. *For a system whose output is given by (8.4), with Gaussian $H_m^V \sim \mathcal{RCN}(0, \Sigma_m)$ as described in (8.13), the output spectrum $Y(\mathbf{k}_y)$ is real/complex normally distributed as follows,*

$$Y(\mathbf{k}_y) \sim \mathcal{RCN}(0, \Sigma_Y),$$

where $\Sigma_Y = \tilde{\Phi} \Sigma_{tot} \tilde{\Phi}^H + \sigma^2 I$

$$\text{and } \Sigma_{tot} = \begin{bmatrix} \Sigma_1 & & 0 \\ & \ddots & \\ 0 & & \Sigma_M \end{bmatrix}. \quad (8.28)$$

Proof. Follows directly from (8.26), Assumptions 8.1, 8.7 and 8.8, and the properties of real/complex normal distributions. \square

Finally, maximum a posteriori (MAP) estimates for the GFRFs can be obtained from the joint distribution of Y and H , by computing the mean of the conditional distribution $H|Y$. The result is provided in the following theorem.

Theorem 8.13. *The MAP estimate of \tilde{H} in (8.26) is*

$$\hat{H}_{MAP} = \Sigma_{tot} \tilde{\Phi}^H \Sigma_Y^{-1} \tilde{Y} \quad (8.29)$$

Proof. Follows from the properties of real/complex normal distributions [67]. \square

8.3.5 Hyperparameter tuning

We must optimize the hyperparameters, η , describing each covariance function $P_m(\eta)$, which are in turn used to construct Σ_m in (8.23). As in [11] and [67], this chapter will use a marginal likelihood maximization (MLM), i.e.

$$\hat{\eta} = \arg \min_{\eta} \tilde{Y}(\mathbf{k}_y)^H \Sigma_Y^{-1}(\eta) \tilde{Y}(\mathbf{k}_y) + \log \det \Sigma_Y(\eta). \quad (8.30)$$

Remark 8.14. *The disadvantages of using a MLM approach were highlighted in Chapter 4 for the time domain case. The same situation exists in the frequency domain for GFRFs, i.e. the dimension of the search space in (8.30) grows quadratically with the maximum nonlinear order, M , and hence the computational burden for the optimization grows even more rapidly. An equivalent EM approach to that of Algorithm 1 in Chapter 4 can be derived, in an almost identical fashion, for the GFRF estimation problem. The EM tuning method is recommended when M is large.*

8.3.6 Comparison with the indirect time domain approach

There is another viable approach for estimating GFRFs under rank deficient conditions, which is to first compute regularized Volterra kernel estimates in the time domain with memory length $n_m = N$, and subsequently transform them using (8.3). One advantage of this indirect approach is the freedom to use input/output measurements of arbitrary length which are not necessarily at steady-state, whereas the method proposed here requires such constraints.

If we are only interested in GFRF estimation for a *specific frequency band*, then the time domain approach has a severe disadvantage when compared to the frequency domain method proposed here. With an indirect time domain approach, we are required to estimate the full set of kernel parameters regardless of frequency band, since each frequency domain parameter depends on the *entire* time domain set (see (8.3)). In direct frequency domain estimation, the limited frequency band directly reduces the set of parameters to be estimated, and the magnitude of reduction grows rapidly with increasing GFRF order, m , since the band is limited along every dimension of the GFRFs.

An example of this parameter reduction concept is visualized in Figure 8.1 for a second order example with $N = n_2 = 21$. We excite a discrete set of frequencies, $\mathbf{k}_u = \{3, 4, 5, 6, 7\}$, and compare the number of parameters requiring estimation in each domain. It is clear that limiting the excited frequency band in two dimensions significantly reduces the parameter set for H_2 to 55 parameters (5 real, 25 complex), while for h_2 the full set of unique Volterra coefficients (231 real) is still required.

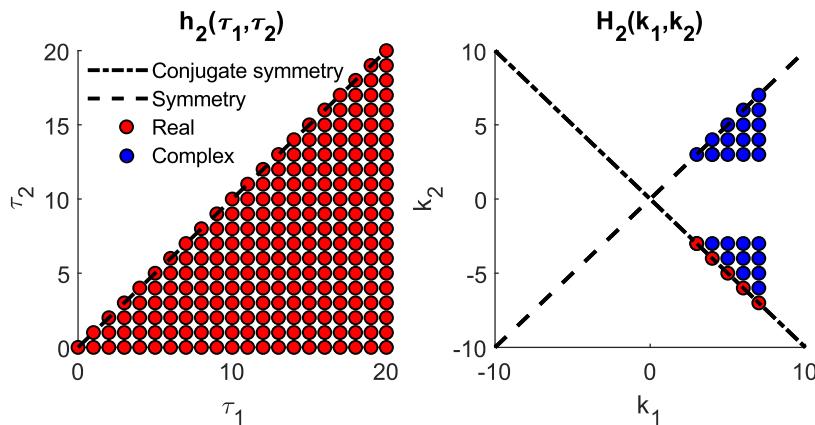


Figure 8.1: Parameters requiring estimation in second order band-limited example where $\mathbf{k}_u = \{3, 4, 5, 6, 7\}$, using a time domain approach (left) and using a direct frequency domain approach (right)

8.4 Simulation examples

Two simulation studies are presented here in order to demonstrate the performance of the proposed method. The first study considers GFRF estimation in a noise-free setting, but using only period of a multisine excitation such that the problem is severely rank-deficient. In the second study, measurement noise is included and the frequencies of interest are limited to a narrow band. For this case a Monte Carlo study is used to compare the proposed GPR method against an indirect time domain approach.

8.4.1 Study 1: rank deficient noise-free case

In the first study, Volterra system outputs are generated using the block structures in Figure 8.2, where the left structure (a) is in a parallel Wiener configuration, and the right structure (b) has a parallel Hammerstein configuration. The linear filters in each structure are defined in the z -domain as

$$G_1(z) = \frac{3z}{z^2 - 1.8z + 0.84}, \quad G_2(z) = 0.25G_1(z).$$

The input, $u(t)$, is given by a periodic multisine with period $N = 80$ and $\mathbf{k}_u = \{0, 1, 2, \dots, 19\}$, which is applied for 10 periods before taking the final period of measurements for estimation purposes, such that all transient effects are essentially negligible.

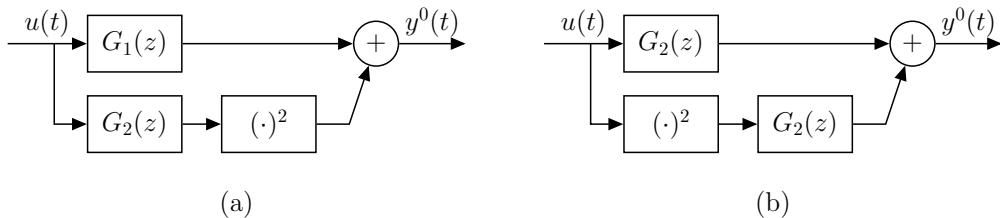


Figure 8.2: System structures used for data generation in the rank deficient noise-free case

We consider identification of each system's nonzero GFRFs, H_1 and H_2 , assuming memory lengths $n_1 = n_2 = N$. The resulting estimation problems are severely rank deficient, with 780 unique parameters requiring estimation, and only 77 corresponding output spectrum measurements. Applying the proposed GPR method, the GFRF estimates are given in Figures 8.3 (for structure (a)) and 8.4 (for structure (b)) alongside the true GFRFs, which can be obtained analytically from the system structure (see Chapter 3.4). Despite severe rank deficiency, it is clearly seen that a good result is achieved for each system at both GFRF orders.

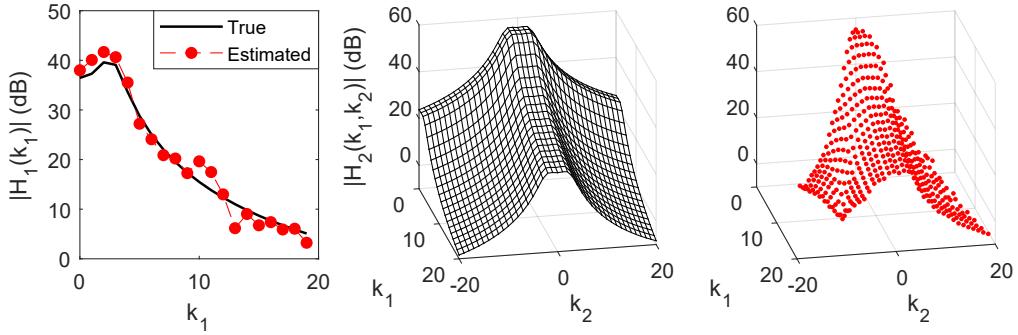


Figure 8.3: Structure (a) - magnitude plots for the first order GFRF estimate (left), true 2nd order GFRF (middle) and 2nd order estimate for unique parameters (right)

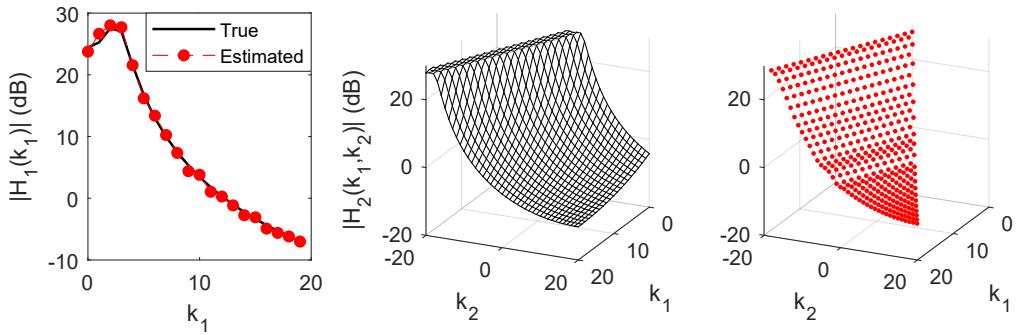


Figure 8.4: Structure (b) - magnitude plots for the first order GFRF estimate (left), true 2nd order GFRF (middle) and 2nd order estimate for unique parameters (right)

8.4.2 Study 2: limited frequency band with measurement noise

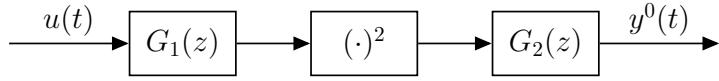


Figure 8.5: Wiener-Hammerstein structure used in the limited frequency band MC study

To highlight the advantage of the proposed method over an indirect time domain approach, a Monte Carlo (MC) study is also performed. The noise-free outputs are generated using the Wiener-Hammerstein structure in Figure 8.5, with linear filters,

$$G_1(z) = \frac{z-1}{z}, \quad G_2(z) = \frac{1.07z}{z^2 - 1.72z + 0.77}.$$

The input is a multisine with period $N = 55$ and a limited band of excited frequencies, $\mathbf{k}_u = \{4, 5, \dots, 12\}$. The output, y^0 , is disturbed by additive white measurement noise. Note that the system structure can be represented using a single GFRF,

H_2 , as was discussed in Chapter 3.4. We will assume for estimation that $n_2 = N$.

For the MC study we consider three different noise levels, with each noise level having 100 input/output realizations. For each realization, transient-free estimation data is obtained in the same manner as in Section 8.4.1. Estimation of the H_2 parameters within the excited band is performed using two separate methods:

1. **GPR-FD** - GPR in the frequency domain as developed in this chapter (i.e. using (8.29) and (8.30)), and using the DC structure (3.24) for each P_m .
2. **GPR-TD** - GPR in the time domain as presented in [11] using the DC structure for P_m , and using periodicity to define the initial conditions $u(t)$ for $t < 0$. GFRF parameters are obtained by the DFT transformation (8.3) on each time domain kernel, followed by the elimination of parameters not relevant to the frequency band of interest.

The noise, $e(t)$, is added in each realization to achieve a Signal-to-Noise Ratio (SNR) of 5, 10 or 20dB.

Estimation errors for each method and realization are quantified using a normalized mean square error (NMSE) metric, given by

$$\text{NMSE} = \frac{\frac{1}{n} \sum_{i=1}^n |\hat{H}_2^V(i) - H_2^V(i)|^2}{\frac{1}{n} \sum_{i=1}^n |H_2^V(i)|^2},$$

where n is the number of GFRF parameters requiring estimation in the excited frequency band, $\hat{H}_2^V(i)$ is the i^{th} element of the GFRF estimate, and $H_2^V(i)$ contains the true GFRF elements in the frequency band of interest.

The errors resulting from each noise level and method are presented as boxplots in Figure 8.6. The GPR-FD accuracy is seen to improve with increasing SNR as expected, however it is also clear that for the case of a limited excitation frequency band, the proposed frequency domain regression method performs better than an indirect time domain approach on the same dataset, for the reasons discussed in Section 8.3.6. More specifically, the number of unique parameters requiring estimation for GPR-TD is 1540 (all real), while for GPR-FD it is only 378 (14 real, 182 complex).

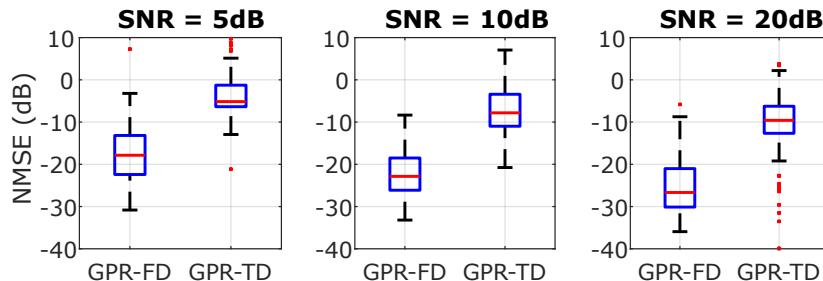


Figure 8.6: NMSEs for the limited frequency band MC study

8.5 Conclusions

In this chapter, a Gaussian process regression method was presented for the estimation of generalized frequency response functions. The method employs the hybrid real/complex Gaussian framework originally developed for the linear case in [67]. The success of the GPR method is dependent on the ability to construct a suitable prior covariance for each GFRF, which is achieved by translating the time domain Volterra kernel covariances designed in [11], and hence their properties, to the frequency domain via vectorized Fourier transformations. Numerical examples demonstrated the ability of the method to produce reasonable estimates under severely rank deficient conditions. It was clearly obvious that the proposed method outperforms indirect time domain estimation of GFRFs in the case of a limited frequency band.

When compared to the equivalent linear method in [67], there is an obvious drawback of the method proposed in this thesis, which is the requirement for steady-state conditions during data collection. When the experiment is not performed in steady-state, the measured output will contain an additional transient component with respect to the periodic spectrum predicted by the GFRF model (8.2). This transient component will negatively affect the estimation accuracy of a method which assumes steady-state whereas the linear case in [67] considered the explicit inclusion of a transient function in the model structure. Furthermore, the prior covariance of the transient function (in a Bayesian setting) was shown to be approximately proportional to the FRF covariance under white noise excitation.

The nature of the output transient in a *nonlinear* system is not well understood for the general case, and further investigation is required in order to model both the GFRFs and transient functions in a GPR estimation method. This investigation is performed in Chapter 9, where the mathematical structure of nonlinear Volterra system transients is derived and examined.

CHAPTER

9

Frequency Domain Response of Nonlinear Systems under Arbitrary Excitation

The steady-state system response to periodic excitation is well understood for both linear and certain nonlinear model classes. When the excitation is not periodic, the measured response will contain both steady-state and transient contributions. For linear systems, these transient contributions have been thoroughly explored, while no equivalent analysis exists for the nonlinear case. In this chapter, an expression is derived in the frequency domain for the system response of all nonlinear systems with a Volterra series representation. The expression contains both steady-state and transient contributions at each nonlinear order, revealing a highly structured view of nonlinear system response. In the context of identification under arbitrary excitation, the output transients must be properly considered to obtain an accurate model, and the implications of the derived expressions on the methods of Chapters 7 and 8 are discussed.

9.1 Introduction

Commonly, in frequency domain studies of dynamical systems, steady-state conditions are assumed for convenience. This assumption implies that the input in the measurement window has been applied periodically for an infinitely long time, such that the system has reached steady-state with a corresponding periodic output response. There are many scenarios, however, where this assumption will be significantly violated, therefore it is important to understand the nature of the system response in the arbitrary excitation case.

For linear systems, the relationship between the measured input and output spectrum in steady-state under periodic excitation is well known, where the ratio of output to input at any given frequency is equal to the system's Frequency Response Function (FRF) at that frequency. When the system is not at steady-state, the measured output spectrum will contain an additional transient term [96], which is dependent on the initial conditions prior to the measurement. Modeling of a linear system's transient term was first explored in [97] from a parametric perspective. For linear systems described nonparametrically (i.e. using an impulse response/FRF), an expression for the transient was derived in [67].

In order to perform a similar transient analysis for nonlinear systems, the Volterra series can be exploited as a nonparametric model structure capable of representing all time-invariant and fading memory (TIFM) nonlinear systems [16]. For Volterra systems, steady-state frequency domain behaviour has been thoroughly explored, producing representations such as the NOFRF and GFRF models considered in this thesis. For Volterra systems under *arbitrary* excitation, however, the analytic form of the transient term is yet to be determined.

In this chapter, we obtain a general expression for the measured output response of a discrete TIFM nonlinear system under arbitrary excitation. The expression is formulated in the frequency domain as a summation of steady-state and transient contributions at each nonlinear order, where the transient terms are defined analogously to the linear case [67], providing insight into their structure and input dependencies. Moreover, the expressions can also be taken to the time domain via appropriate transformations. The validity of the derived expressions is demonstrated through numerical examples, and the implications for system identification are discussed. In particular, the possibility of extending the GPR estimation method in Chapter 8 to non-steady-state data is considered, where the output transients may be modeled using insights obtained in the current chapter.

9.2 Preliminaries and problem formulation

In this section, we provide rigorous definitions for the tools and transformations used in frequency domain analysis. The Volterra series and GFRF models are reiterated

again, with a slightly altered formulation in the frequency domain to allow for a more convenient analysis in the sequel. For completeness, we also provide a brief summary of the linear system transient expression from [67].

9.2.1 Frequency domain analysis

Considering the analysis of N -sample signals, the following definitions are required.

Definition 9.1 (ω_k). *The k^{th} discrete frequency is denoted ω_k and defined as*

$$\omega_k = \frac{2\pi k}{N} \quad (9.1)$$

Definition 9.2 (DFT). *For an N -sample time domain signal $\{x(t)\}_{t=0}^{N-1} \in \mathbb{R}^N$, the Discrete Fourier Transform (DFT) of x at frequency bin k is given by*

$$X(k) = \sum_{t=0}^{N-1} x(t)e^{-j\omega_k t}, \quad (9.2)$$

where $X(k) \in \mathbb{C}$ and ω_k is defined in (9.1).

Definition 9.3 (IDFT). *For an N -sample frequency domain signal $\{X(k)\}_{k=0}^{N-1} \in \mathbb{C}^N$, the inverse DFT (IDFT) of X is given by*

$$x(t) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j\omega_k t}, \quad (9.3)$$

where $x(t) \in \mathbb{R}$.

Definition 9.4 (m -DTFT). *For a discrete m -dimensional real function $x(\tau_1, \dots, \tau_m)$, $\tau_1, \dots, \tau_m \in \mathbb{N}$, and in the context of N -sample signals, the N -periodic m -dimensional Discrete-Time Fourier Transform (m -DTFT) is given by*

$$X(k_1, \dots, k_m) = \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_m=0}^{\infty} x(\tau_1, \dots, \tau_m)e^{-j\omega_{k_1}\tau_1} \dots e^{-j\omega_{k_m}\tau_m}. \quad (9.4)$$

Notation 9.5 (System input/output). *For a discrete-time dynamical system, we denote the input and output of the system as $u(t)$ and $y(t)$ respectively, where $t \in \mathbb{Z}$. If an N -sample portion of the input/output sequence is measured, the measured sequences are denoted $\{u(t)\}_{t=0}^{N-1}$ and $\{y(t)\}_{t=0}^{N-1}$, with corresponding N -point DFTs given by $\{U(k)\}_{k=0}^{N-1}$ and $\{Y(k)\}_{k=0}^{N-1}$.*

9.2.2 Volterra series formulation

Recall the discrete time Volterra series model structure,

$$\begin{aligned} y(t) &= h_0 + \sum_{m=1}^M y_m(t), \\ y_m(t) &= \sum_{\tau_1=0}^{\infty} \dots \sum_{\tau_m=0}^{\infty} h_m(\tau_1, \dots, \tau_m) \prod_{\tau=\tau_1}^{\tau_m} u(t-\tau), \end{aligned} \quad (9.5)$$

which can be used to represent any time-invariant and fading memory nonlinear system [16]. Note that in this chapter, we leave each Volterra kernel untruncated with infinite memory length, as this is the most general case. The maximum nonlinear order, M , can also be arbitrarily large.

We have also seen that the series can be expressed in the frequency domain, i.e.

$$Y(k) = H_0(k) + \sum_{m=1}^M Y_m(k), \quad (9.6)$$

where $\{Y(k)\}_{k=0}^{N-1}$ is the DFT of $\{y(t)\}_{t=0}^{N-1}$, $Y_m(k)$ is the m^{th} order output spectrum contribution, and $H_0(k) = h_0\delta(k)$ with $\delta(k)$ the unit impulse function. When the system is in steady-state, i.e. when $u(t)$ and $y(t)$ are N -periodic and N samples are measured, an expression for the m^{th} order output spectrum contribution can be derived using the GFRF model structure (3.3), which for $m \geq 2$ can be written as

$$Y_m^{ss}(k) = \frac{1}{N^{m-1}} \sum_{k_2=0}^{N-1} \dots \sum_{k_m=0}^{N-1} H_m(k - k_+, k_2, \dots, k_m) U(k - k_+) \prod_{i=2}^m U(k_i), \quad (9.7)$$

where

$$k_+ = \sum_{i=2}^m k_i, \quad (9.8)$$

and H_m is the m -DTFT of Volterra kernel h_m , typically labelled as the m^{th} order GFRF. For $m = 1$, (9.7) reduces to the widely known linear relation,

$$Y_1^{ss}(k) = H_1(k) \cdot U(k). \quad (9.9)$$

9.2.3 Problem statement

The problem statement is introduced by first clarifying the system assumption.

Assumption 9.6 (Volterra system). *The system of interest is a causal, single-input single-output TIFM nonlinear system which satisfies the Volterra series representation (9.5).*

Given N -sample input and output measurements, $\{u(t)\}_{t=0}^{N-1}$ and $\{y(t)\}_{t=0}^{N-1}$, from the system, the aim is to find an expression for the measured frequency domain

system response, $\{Y(k)\}_{k=0}^{N-1}$, as the sum of steady-state contributions, $Y_m^{ss}(k)$, and transient contributions from each nonlinear order.

9.2.4 Linear System Response

The problem described in Section 9.2.3 has been extensively studied for the linear case, i.e. where only the $h_1(\cdot)$ kernel is non-zero in (9.5). In this case, $h_1(\tau_1)$ corresponds to the system impulse response, and its DTFT, H_1 , is the system FRF. While the steady-state contribution has been established in (9.9), the transient contribution is less trivial. In this chapter, we consider a nonparametric approach, hence we refer to [67] for a transient expression obtained using the nonparametric impulse response model:

Consider the system input, $u(t)$, output, $y(t)$, and their N -sample measurements $\{u(t)\}_{t=0}^{N-1}$ and $\{y(t)\}_{t=0}^{N-1}$. We will also introduce the function f as the input difference,

$$f(t) = \begin{cases} u(t) - u(t+N) & \text{for } t < 0, \\ 0 & \text{for } t \geq 0, \end{cases} \quad (9.10)$$

which is defined $\forall t$ and thus can depend on inputs $u(t)$ outside the measured window. The DFT spectrum of the measured output is now given by,

$$\begin{aligned} Y_1(k) &= Y_1^{ss}(k) + T_1^1(k), \quad k = 0, \dots, N-1, \\ T_1^1(k) &= \sum_{t_1=0}^{\infty} g_1^1(t_1) e^{-j\omega_k t_1}, \\ g_1^1(t_1) &= \sum_{\tau_1=t_1+1}^{\infty} h_1(\tau_1) f(t_1 - \tau_1). \end{aligned} \quad (9.11)$$

The linear transient contribution, $T_1^1(k)$, is formed as the DTFT of $g_1^1(t_1)$, which is a free response of the system dependent on the unmeasured initial conditions, $u(t)$, $\forall t < 0$. The subscript denotes the (1st) nonlinear order of the associated system Volterra kernel, and the superscript is an index that will be required in the following sections. Note that in the case where $u(t)$ is an N -periodic signal applied from $t = -\infty$, then $f(t) = 0 \ \forall t$, and the response reduces to the steady-state contribution as expected.

9.3 The second order Volterra kernel case

We now extend the linear result from [67], given in (9.11), to the second order Volterra kernel case, where only the kernel $h_2(\tau_1, \tau_2)$ is non-zero in (9.5). First we establish some relationships between shifted time domain and DFT signals.

Proposition 9.7. *For an arbitrary input, $u(t)$, and corresponding DFT signal $U(k)$*

obtained via the DFT of $\{u(t)\}_{t=0}^{N-1}$, it can be easily shown that,

$$\sum_{t=0}^{N-1} u(t - \tau) e^{-j\omega_k(t-\tau)} = U(k) + \sum_{t'=-\tau}^{-1} f(t') e^{-j\omega_k t'}, \quad (9.12)$$

where $f(t)$ is defined as in (9.10). Applying an inverse DFT, we also have that

$$u(t - \tau) = \frac{1}{N} \sum_{k=0}^{N-1} (U(k) + \sum_{t'=-\tau}^{-1} f(t') e^{-j\omega_k t'}) e^{j\omega_k(t-\tau)}. \quad (9.13)$$

The spectrum of the measured output response can now be derived for a second order Volterra kernel.

Theorem 9.8. For a system described by (9.5) with only $h_2(\tau_1, \tau_2)$ non-zero, the N -sample measured system response can be expressed in the frequency domain as,

$$\begin{aligned} Y_2(k) &= Y_2^{ss}(k) + 2 T_2^1(k) + T_2^2(k), \quad k = 0, \dots, N-1, \\ T_2^i(k) &= \frac{1}{N} \sum_{k_2=0}^{N-1} \sum_{t_1=0}^{\infty} \sum_{t_2=0}^{\infty} g_2^i(t_1, t_2) e^{-j\omega_{k-k_2} t_1} e^{-j\omega_{k_2} t_2}, \\ g_2^1(t_1, t_2) &= \sum_{\tau_1=t_1+1}^{\infty} \sum_{\tau_2=t_2+1-N}^{t_2} h_2(\tau_1, \tau_2) f(t_1 - \tau_1) u(t_2 - \tau_2), \\ g_2^2(t_1, t_2) &= \sum_{\tau_1=t_1+1}^{\infty} \sum_{\tau_2=t_2+1}^{\infty} h_2(\tau_1, \tau_2) f(t_1 - \tau_1) f(t_2 - \tau_2), \end{aligned} \quad (9.14)$$

where $Y_2(k)$ is the measured output DFT spectrum, $Y_2^{ss}(k)$ is the steady-state contribution from (9.7) and $T_2^i(k)$ is the i^{th} transient contribution found from the i^{th} 2-dimensional time domain response, g_2^i .

Proof. From (9.5), we have that

$$y_2(t) = \sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\infty} h_2(\tau_1, \tau_2) u(t - \tau_1) u(t - \tau_2).$$

Applying the DFT to each side, we obtain,

$$Y_2(k) = \sum_{t=0}^{N-1} \sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\infty} h_2(\tau_1, \tau_2) u(t - \tau_1) u(t - \tau_2) e^{-j\omega_k t}. \quad (9.15)$$

The term $e^{-j\omega_k t}$ can be expressed as,

$$e^{-j\omega_k t} = e^{-j\omega_{k-k_2}\tau_1} e^{-j\omega_{k_2}\tau_2} e^{-j\omega_{k-k_2}(t-\tau_1)} e^{-j\omega_{k_2}(t-\tau_2)}. \quad (9.16)$$

Substituting (9.16) into (9.15), we can now use the results (9.12) and (9.13) from

Proposition 9.7 to obtain,

$$Y_2(k) = \frac{1}{N} \sum_{k_2=0}^{N-1} \sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\infty} h_2(\tau_1, \tau_2) e^{-j\omega_{k-k_2}\tau_1} e^{-j\omega_{k_2}\tau_2} \\ \times \left(U(k - k_2) + \sum_{t_1=-\tau_1}^{-1} f(t_1) e^{-j\omega_{k-k_2}t_1} \right) \left(U(k_2) + \sum_{t_2=-\tau_2}^{-1} f(t_2) e^{-j\omega_{k_2}t_2} \right). \quad (9.17)$$

Expanding the 2nd line in (9.17), transforming the U terms, applying the changes of variable, $t_i = t_i - \tau_i$, and rearranging the order of sums, it can be shown that,

$$Y_2(k) = \underbrace{\frac{1}{N} \sum_{k_2=0}^{N-1} \left(\sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\infty} h_2(\tau_1, \tau_2) e^{-j\omega_{k-k_2}\tau_1} e^{-j\omega_{k_2}\tau_2} \right) U(k - k_2) U(k_2)}_{H_2(k-k_2, k_2)} \\ + \frac{1}{N} \sum_{k_2=0}^{N-1} \sum_{t_1=0}^{\infty} \sum_{t_2=0}^{\infty} \left(g_2^{uf}(t_1, t_2) + g_2^{fu}(t_1, t_2) + g_2^{ff}(t_1, t_2) \right) e^{-j\omega_{k-k_2}t_1} e^{-j\omega_{k_2}t_2}, \quad (9.18)$$

where

$$g_2^{uf}(t_1, t_2) = \sum_{\tau_1=t_1+1-N}^{t_1} \sum_{\tau_2=t_2+1}^{\infty} h_2(\tau_1, \tau_2) u(t_1 - \tau_1) f(t_2 - \tau_2), \\ g_2^{fu}(t_1, t_2) = \sum_{\tau_1=t_1+1}^{\infty} \sum_{\tau_2=t_2+1-N}^{t_2} h_2(\tau_1, \tau_2) f(t_1 - \tau_1) u(t_2 - \tau_2) = g_2^1(t_1, t_2), \quad (9.19) \\ g_2^{ff}(t_1, t_2) = \sum_{\tau_1=t_1+1}^{\infty} \sum_{\tau_2=t_2+1}^{\infty} h_2(\tau_1, \tau_2) f(t_1 - \tau_1) f(t_2 - \tau_2) = g_2^2(t_1, t_2).$$

Due to the symmetry of h_2 and the N -periodicity of $e^{j\omega_k t}$, it can be shown that g_2^{uf} and g_2^{fu} provide equal contributions to $Y_2(k)$, hence we can replace $g_2^{uf} + g_2^{fu}$ with $2g_2^1 = 2g_2^{fu}$ in (9.18), thus reducing it to

$$Y_2(k) = Y_2^{ss}(k) + 2T_2^1(k) + T_2^2(k), \quad (9.20)$$

which completes the proof. \square

From Theorem 9.8, we see that the transient contributions from a second order Volterra kernel exhibit a considerably more complex behaviour than the linear case in (9.11). The quantity $g_2^2(t_1, t_2)$ in (9.14) can be seen as the 2-dimensional analog to $g_1^1(t_1)$ in (9.11), as it is formed from a 2D convolution of the system kernel and the difference function, f . The term $g_2^1(t_1, t_2)$ is also present in (9.14), however, which depends directly on the measured input sequence, $\{u(t)\}_{t=0}^{N-1}$.

9.4 The general Volterra series case

For a general TIFM nonlinear system, the intuition gained in Section 9.3 can be used to formulate the contribution to the measured output spectrum from any m^{th} order kernel, h_m , in the corresponding Volterra series representation.

Theorem 9.9. *For a system described by (9.5), the N -sample measured system response can be expressed in the frequency domain as,*

$$\begin{aligned} Y(k) &= Y_0^{ss}(k) + \sum_{i=1}^M Y_m(k), \quad k = 0, \dots, N-1, \\ Y_m(k) &= Y_m^{ss}(k) + \sum_{i=1}^m \binom{m}{i} T_m^i(k), \end{aligned} \quad (9.21)$$

where $Y_m^{ss}(k)$ is the steady-state contribution from (9.7) (and $Y_0^{ss} = H_0$), and T_m^i are transient terms which emerge in the arbitrary excitation case, given by

$$T_m^i(k) = \frac{1}{N^{m-1}} \sum_{k_2=0}^{N-1} \dots \sum_{k_m=0}^{N-1} \sum_{t_1=0}^{\infty} \dots \sum_{t_m=0}^{\infty} g_m^i(t_1, \dots, t_m) e^{-j\omega_{k-k_1} t_1} e^{-j\omega_{k_2} t_2} \dots e^{-j\omega_{k_m} t_m}, \quad (9.22)$$

with

$$\begin{aligned} g_m^i(t_1, \dots, t_m) &= \sum_{\tau_1=t_1+1}^{\infty} \dots \sum_{\tau_i=t_i+1}^{\infty} \sum_{\tau_{i+1}=t_{i+1}+1-N}^{t_{i+1}} \dots \\ &\quad \sum_{\tau_m=t_m+1-N}^{t_m} h_m(\tau_1, \dots, \tau_m) \prod_{n=1}^i f(t_n - \tau_n) \prod_{p=i+1}^m u(t_p - \tau_p) \end{aligned} \quad (9.23)$$

Proof. The proof for any m^{th} order kernel follows the same steps and logic as the proof of Theorem 9.8. \square

From (9.23), it can be seen that there are no input, u , terms in $g_m^m(t_1, \dots, t_m)$, hence it can be considered as an m -dimensional free response (as in the linear and second order cases). All other responses, $g_m^i(t_1, \dots, t_m)$ for $i \neq m$, depend directly on the input in the measured window.

The mathematical structure of the response of a nonlinear system revealed in this work and given in Theorem 9.9 can be efficiently visualized in a ‘Pascal’s triangle’ arrangement, shown in Figure 9.1. In the figure, the m^{th} row represents the m^{th} order contribution to the total output spectrum, $Y(k)$. Furthermore, the left edge contains the steady-state contributions, while the right edge contains the free response transient terms. The remaining terms (shaded red) are all directly dependent on the measured input sequence, and only appear for $m \geq 2$.

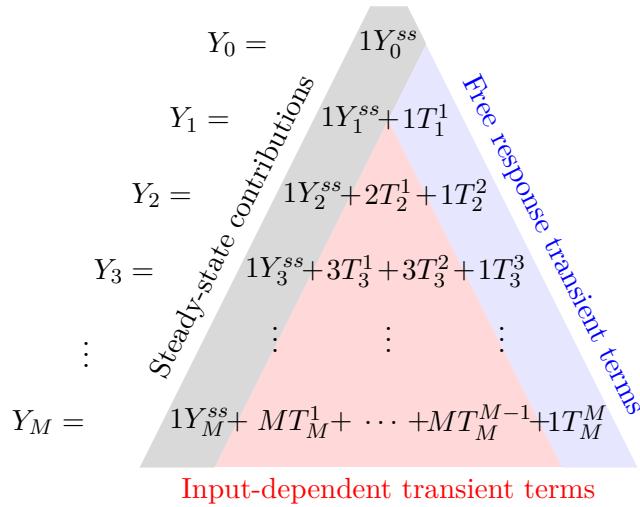


Figure 9.1: A Pascal’s Triangle interpretation of the frequency domain output response for a TIFM nonlinear system

9.5 Numerical examples

In this section, we will investigate the derived transient response expressions using a simple nonlinear example structure. We consider systems described by the Wiener structure shown in Figure 9.2, where $(\cdot)^m$ denotes exponentiation by m , and the discrete linear filter, $G(z)$, is chosen to be slightly underdamped with an effective memory length of approximately 60 samples, given by,

$$G(z) = \frac{0.75z}{z^2 - 1.8z + 0.84}. \quad (9.24)$$

The system structure is designed such that it can be represented by a single Volterra kernel, whose order we may select via our choice of $m \in \mathbb{N}$, and whose form can be obtained analytically from the impulse response of $G(z)$ (see Chapter 3.4).

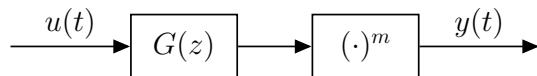


Figure 9.2: Wiener structure used for the numerical examples

First we examine how the (total) transient response changes with increasing nonlinear order, m . The total transient is defined as the difference between the measured output and the steady-state output, i.e.

$$T_m(k) = Y_m(k) - Y_m^{ss}(k) = \sum_{i=1}^m \binom{m}{i} T_m^i(k), \quad (9.25)$$

where T_m is the total transient contribution from the m^{th} nonlinear order. Using the

system structure in Figure 9.2, and setting the nonlinear exponent to $m = 1, 2$ and 3 in turn, 20 Gaussian white noise input realizations are applied to the system and the total transient is measured in a 60-sample window. Figure 9.3 shows the 20 transient realizations at each nonlinear order. It can be seen that the linear ($m = 1$) transients are smooth functions of frequency, since they are formed purely from the free response, $g_1^1(t_1)$ in (9.11). As the nonlinear order increases, however, the transients become more dominated by the white noise input in the measured window, as can be expected by observing the response structure shown in Figure 9.1.

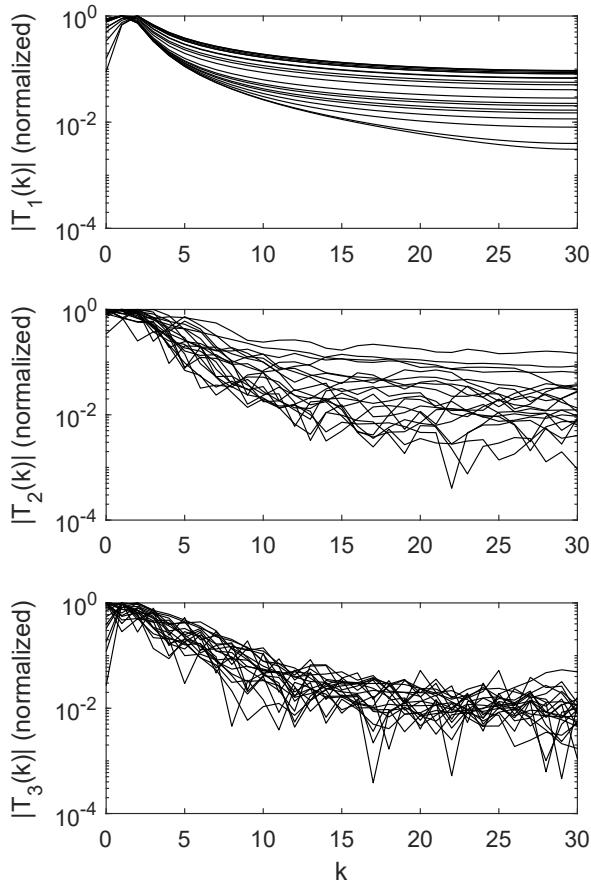


Figure 9.3: Total frequency domain transients for a Wiener system with linear (top), quadratic (middle) and cubic (bottom) nonlinearities

Next, we investigate more deeply a single realization of the system response, for the second order ($m = 2$) case. Again, using a Gaussian white noise input, we compute the two-dimensional time domain quantities, $g_2^1(t_1, t_2)$ and $g_2^2(t_1, t_2)$ from (9.14), which are visualized as surfaces in Figure 9.5. The free response, $g_2^2(t_1, t_2)$, resembles the underlying system Volterra kernel, $h_2(\tau_1, \tau_2)$ (shown in Figure 9.4) while g_2^1 is structurally more complex.

The corresponding frequency domain transient contributions, $T_2^1(k)$ and $T_2^2(k)$,

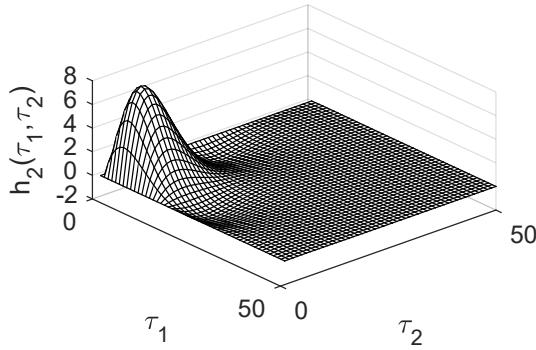


Figure 9.4: Volterra kernel, h_2 , for the Wiener system with quadratic nonlinearity

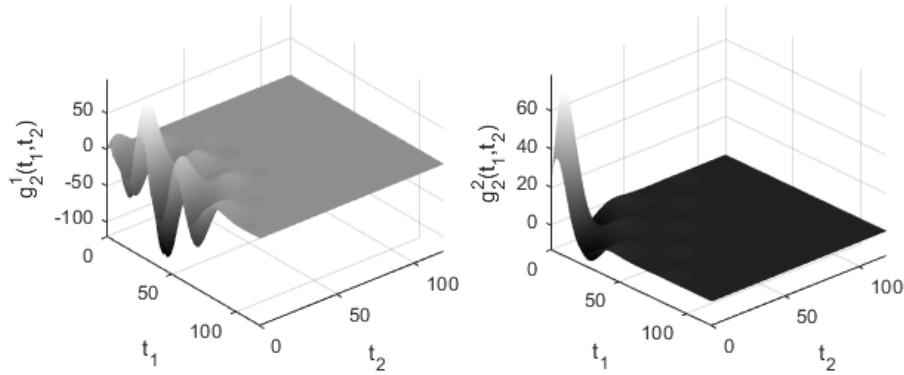


Figure 9.5: Time domain responses g_2^1 (left) and g_2^2 (right) for a Wiener system with quadratic nonlinearity

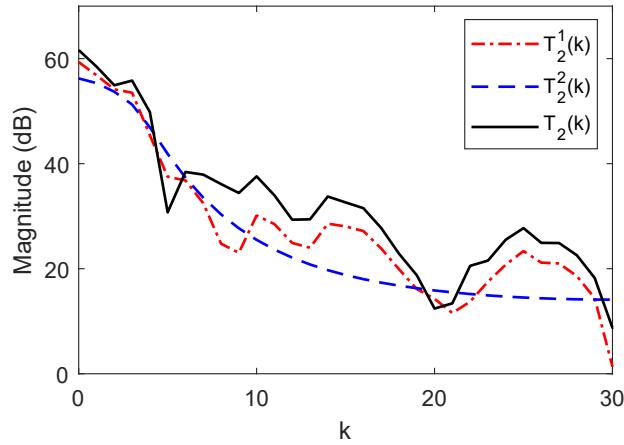


Figure 9.6: Frequency domain transient contributions for a Wiener system with quadratic nonlinearity

are also computed from (9.14), along with the total transient, $T_2(k)$ from (9.25). These quantities are plotted in Figure 9.6. It is clear that the free response transient, T_2^2 , has a smooth, simple structure like the linear transients, $T_1(k)$, in Figure 9.3. On the other hand, the input-dependent $T_2^1(k)$ has a more complex structure due

to the noisy excitation signal, which is also reflected in the total transient measured from the system.

9.6 Insights for system identification

In the case where a dynamic system is identified in the frequency domain, it is not always possible or practical to enforce steady-state conditions on the system under study. As such, it is important to simultaneously identify the transient contributions present in the measured response, otherwise the quality of the estimated model will be compromised [97].

For linear systems, several techniques have been developed for nonparametric estimation of the system FRF in the presence of transients. The local polynomial method (LPM) [96], [112] exploits the smoothness of the transient $T_1^1(k)$ in (9.11), by estimating a local polynomial around each frequency bin of the measured response in order to produce smooth estimates of the transient alongside the FRF. The local rational method (LRM) [81] extends this idea by estimating rational transient models locally at each DFT bin. The most recent development is the Gaussian Process Regression (GPR) method designed in [67], where the linear FRF and transient function are estimated simultaneously by exploiting the proportionality of transient and FRF covariance matrices (in a Bayesian setting) under white noise excitation.

It is clear that in the linear case, accurate frequency domain identification relies on exploiting some special properties of the linear transient term, $T_1^1(k)$. In the case of TIFM nonlinear systems, we have shown that there are numerous contributions to a system's transient response, and many of these contributions depend heavily on the input inside the measured window. We have also seen in Section 9.5 that when the input is non-smooth, the input-dependent transient contributions can be structurally complex. This observation suggests that LPM or LRM-based approaches would need significant modification in order to account for the increased complexity of transients at higher nonlinear orders. For the Bayesian frequency domain methods proposed in this part of the thesis, it is of course the linear GPR method which shows the most promise for extension.

9.6.1 Transients in a parallel Hammerstein system

In Chapter 7, for the parallel Hammerstein case, the linear GPR results from [67] could be used directly in each NOFRF branch. This was because it was obvious by inspection of the model structure that the transient in any given branch is generated directly from input exponents passing through a linear filter. Now that we possess a more general framework, in Theorem 9.9, for describing the system response of all TIFM nonlinear systems, it is interesting to investigate whether the derived expressions predict the linear transient behaviour of parallel Hammerstein systems.

The structure of the Volterra series terms will play a key role in the analysis.

Proposition 9.10. *The Volterra series expansion of a system with parallel Hammerstein structure has Volterra kernels which are strictly diagonal, i.e. each kernel $h_m(\tau_1, \dots, \tau_m)$ can only be non-zero for elements where $\tau_1 = \dots = \tau_m$.*

The result in Proposition 9.10 is well known in the literature, with detailed discussions in [134] and [60]. The special case of a single Hammerstein branch was also discussed in Chapter 3.4. In theory, the diagonal nature of the kernels can be exploited in the transient expressions of Theorem 9.9, to show the reduction of all transient contributions into free responses analogous to the linear free response, $T_1^1(k)$. In practice, the analysis becomes very cumbersome as the nonlinear order, m , is increased, hence only the result for the second order Volterra kernel is provided.

Theorem 9.11. *Consider a system described by (9.5) with only $h_2(\tau_1, \tau_2)$ non-zero, and where h_2 is a diagonal kernel. The N -sample measured system response can be expressed in the frequency domain as,*

$$Y_2(k) = Y_2^{ss}(k) - T_2^2(k) + 2T_2^D(k), \quad k = 0, \dots, N-1, \quad (9.26)$$

where $Y_2^{ss}(k)$ is the steady-state contribution from (9.7), $T_2^2(k)$ is the second order free response transient contribution in (9.14), and $T_2^D(k)$ is defined by

$$T_2^D(k) = \frac{1}{N} \sum_{k_2=0}^{N-1} \sum_{t_1=0}^{\infty} \sum_{t_2=0}^{\infty} \sum_{\tau_1=t_1+1}^{\infty} \sum_{\tau_2=t_2+1}^{\infty} h_2(\tau_1, \tau_2) u(t_1 - \tau_1) f(t_2 - \tau_2) e^{-j\omega_{k-k_2} t_1} e^{-j\omega_{k_2} t_2} \quad (9.27)$$

Proof. See Appendix 9.A. □

The result shows that for a second order *diagonal* kernel there are no longer any transient contributions which depend directly on the input in the measured window, and the two remaining contributions are seen to be free responses to the initial conditions as in the linear case of $T_1^1(k)$. This provides some intuition as to why the transients in a parallel Hammerstein system behave like linear system transients.

9.6.2 GFRF estimation in the presence of transients

It may be possible to extend the GFRF estimation method proposed in Chapter 8 to simultaneously estimate transient functions in the case of arbitrary excitation. While a GPR approach will not be as simple here as in the linear case, the transient expressions derived in this chapter can still provide the insight required to design appropriate Gaussian priors for each transient contribution. For example, it can be shown that for a white noise input and Gaussian Volterra kernels, the transient response $g_m^m(t_1, \dots, t_m)$ from (9.23) will also be Gaussian with covariance proportional to the covariance of $h_m(\tau_1, \dots, \tau_m)$ (as in the linear case). This result can

be observed in Figures 9.4 and 9.5, where $g_2^2(t_1, t_2)$ resembles the underlying system kernel. Such a result does not exist for the input-dependent responses, $g_m^i(t_1, \dots, t_m)$ for $i < m$, however in these cases, the covariance may be formed by combining our knowledge of the Volterra kernel covariance with the transient's known dependence on the input in the measured window. The principal challenge then is to develop a method which can construct and optimize each of the Gaussian priors for the estimation, while still maintaining accuracy and computational feasibility.

9.7 Conclusion

In this chapter, an expression was derived for the measured frequency domain system response of all fading memory nonlinear systems which can be described by the discrete Volterra series. The expression contains the steady-state response of the system at each nonlinear order, as well as a number of transient contributions which arise when the steady-state condition is not met. The different response contributions can be organized in a Pascal's triangle arrangement, revealing a highly structured view of the frequency domain response for nonlinear systems. Unlike the linear case, many of the higher order transient contributions can depend heavily on the measured input, and this may have implications for nonlinear frequency domain identification. While Hammerstein and parallel Hammerstein systems have a linear-style output transient, in general, modifications must be made to the established linear techniques if they are to be applicable in a nonlinear setting.

9.A Transients proof for a second order diagonal kernel

The following is a proof for Theorem 9.11:

From (9.14) we have that

$$\begin{aligned}
 T_2^2(k) &= \frac{1}{N} \sum_{k_2=0}^{N-1} \sum_{t_1=0}^{\infty} \sum_{t_2=0}^{\infty} \sum_{\tau_1=t_1+1}^{\infty} \sum_{\tau_2=t_2+1}^{\infty} h_2(\tau_1, \tau_2) f(t_1 - \tau_1) f(t_2 - \tau_2) e^{-j\omega_{k-k_2} t_1} e^{-j\omega_{k_2} t_2} \\
 &= \frac{1}{N} \sum_{k_2=0}^{N-1} \sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\infty} h_2(\tau_1, \tau_2) e^{-j\omega_{k-k_2} \tau_1} e^{-j\omega_{k_2} \tau_2} \sum_{t_1=-\tau_1}^{-1} f(t_1) e^{-j\omega_{k-k_2} t_1} \sum_{t_2=-\tau_2}^{-1} f(t_2) e^{-j\omega_{k_2} t_2}.
 \end{aligned} \tag{9.28}$$

The term $\sum_{t_1=-\tau_1}^{-1} f(t_1) e^{-j\omega_{k-k_2} t_1}$ in (9.28) can be expanded as follows,

$$\sum_{t_1=-\tau_1}^{-1} f(t_1) e^{-j\omega_{k-k_2} t_1}$$

$$\begin{aligned}
 &= \sum_{t_1=-\tau_1}^{-1} (u(t_1) - u(t_1 + N)) e^{-j\omega_{k-k_2} t_1} \\
 &= \sum_{t_1=-\tau_1}^{-1} u(t_1) e^{-j\omega_{k-k_2} t_1} - \sum_{t_1=-\tau_1}^{-1} u(t_1 + N) e^{-j\omega_{k-k_2} t_1} \\
 &= \sum_{t_1=-\tau_1}^{-1} u(t_1) e^{-j\omega_{k-k_2} t_1} - \sum_{t_1=N-\tau_1}^{N-1} u(t_1) e^{-j\omega_{k-k_2} t_1} \\
 &\quad (\text{due to } N\text{-periodicity of } e^{-j\omega_{k-k_2} t_1}) \\
 &= \sum_{t_1=-\tau_1}^{-1} u(t_1) e^{-j\omega_{k-k_2} t_1} - \sum_{t_1=0}^{N-1} u(t_1) e^{-j\omega_{k-k_2} t_1} + \sum_{t_1=0}^{N-\tau_1-1} u(t_1) e^{-j\omega_{k-k_2} t_1}. \quad (9.29)
 \end{aligned}$$

Substituting (9.29) into (9.28) produces the three components which make up $T_2^2(k)$:

$$\begin{aligned}
 T_2^2(k) &= T_2^D(k) - T_2^1(k) + T_2^Q(k), \\
 T_2^D(k) &= \frac{1}{N} \sum_{k_2=0}^{N-1} \sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\infty} h_2(\tau_1, \tau_2) e^{-j\omega_{k-k_2} \tau_1} e^{-j\omega_{k_2} \tau_2} \sum_{t_1=-\tau_1}^{-1} u(t_1) e^{-j\omega_{k-k_2} t_1} \\
 &\quad \cdot \sum_{t_2=-\tau_2}^{-1} f(t_2) e^{-j\omega_{k_2} t_2}, \\
 T_2^Q(k) &= \frac{1}{N} \sum_{k_2=0}^{N-1} \sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\infty} h_2(\tau_1, \tau_2) e^{-j\omega_{k-k_2} \tau_1} e^{-j\omega_{k_2} \tau_2} \sum_{t_1=0}^{N-\tau_1-1} u(t_1) e^{-j\omega_{k-k_2} t_1} \\
 &\quad \cdot \sum_{t_2=-\tau_2}^{-1} f(t_2) e^{-j\omega_{k_2} t_2}, \quad (9.30)
 \end{aligned}$$

where $T_2^1(k)$ is defined in (9.14), and the term $T_2^D(k)$ is easily shown to be equivalent to the term in (9.27). While the expression in (9.30) holds for any second order Volterra kernel, the term $T_2^Q(k)$ can be further elaborated in the parallel Hammerstein case where $h_2(\tau_1, \tau_2)$ is a diagonal kernel. Considering only the diagonal elements $h_2(\tau, \tau)$, the following reduction is made,

$$\begin{aligned}
 T_2^Q(k) &= \frac{1}{N} \sum_{k_2=0}^{N-1} \sum_{\tau=0}^{\infty} h_2(\tau, \tau) e^{-j\omega_{k-k_2} \tau} e^{-j\omega_{k_2} \tau} \sum_{t_1=0}^{N-\tau-1} u(t_1) e^{-j\omega_{k-k_2} t_1} \sum_{t_2=N-\tau}^{N-1} f(t_2 - N) e^{-j\omega_{k_2} t_2} \\
 &= \frac{1}{N} \sum_{\tau=0}^{\infty} h_2(\tau, \tau) e^{-j\omega_k \tau} \sum_{t_1=0}^{N-\tau-1} \sum_{t_2=N-\tau}^{N-1} u(t_1) f(t_2 - N) e^{-j\omega_k t_1} \sum_{k_2=0}^{N-1} e^{-j\omega_{k_2} (t_2 - t_1)} \\
 &= 0, \quad (9.31)
 \end{aligned}$$

since (observing that $(t_2 - t_1) \in \mathbb{Z}$ and $0 < t_2 - t_1 < N$ due to the bounds of the summations),

$$\begin{aligned}
 \sum_{k_2=0}^{N-1} e^{-j\omega_{k_2}(t_2-t_1)} &= \sum_{k_2=0}^{N-1} \left(e^{-j\frac{2\pi(t_2-t_1)}{N}} \right)^{k_2} = \frac{1 - \left(e^{-j\frac{2\pi(t_2-t_1)}{N}} \right)^N}{1 - e^{-j\frac{2\pi(t_2-t_1)}{N}}} \\
 &= \frac{1 - e^{-j2\pi(t_2-t_1)}}{1 - e^{-j\frac{2\pi(t_2-t_1)}{N}}} = \frac{1 - 1}{1 - \underbrace{e^{-j\frac{2\pi(t_2-t_1)}{N}}}_{\neq 1}} = 0.
 \end{aligned}$$

Thus (9.30) is reduced, in the diagonal kernel case, to

$$T_2^1(k) = T_2^D(k) - T_2^2(k). \quad (9.32)$$

Substituting (9.32) into the second order response equation (9.14) yields

$$Y_2(k) = Y_2^{ss}(k) - T_2^2(k) + 2T_2^D(k) \quad (9.33)$$

as required. \square

PART

IV

Model Predictive Control

CHAPTER

10

Computationally Efficient MPC for Arbitrary Order Volterra-Laguerre Models

Previous research conducted in the area of Volterra series model-based control has been limited to low order models. In this chapter, a state space model structure is developed to represent any arbitrary order Volterra series model expanded using Laguerre basis functions. The model structure is then used to define a model predictive control optimization problem, which is shown to reduce to a computationally efficient root-finding problem under the condition that the move horizon is equal to one. Using these results, a generic Volterra-Laguerre MPC algorithm is defined, with modifications to handle input constraints and offset-free reference tracking. The stability of the proposed control method is also investigated.

10.1 Introduction

Model predictive control (MPC) has received an explosion of interest in recent decades [78], and in particular the use of *nonlinear* prediction models presents significant and unique challenges [52]. While the use of linearized plant models in model predictive control can lead to poor performance or instability if nonlinear dynamics are strong enough, we must also avoid the construction of complex nonlinear models whose associated optimization problems are highly non-convex or computationally unfeasible [77].

In this chapter, the application of MPC to nonlinear Volterra series models will be considered. The system prediction models are assumed to be in the form of Laguerre basis function expansions, for which the standard time domain Volterra series is a special case. Robust identification of Volterra-Laguerre models was discussed in Chapter 5 of this thesis, where the proposed method could, in theory, estimate the series up to an arbitrarily large nonlinear order. However, the use of such models in model-based control is problematic, since there are no existing results on the use of high order Volterra-Laguerre models for this purpose. There has been some work published on feedback control and MPC using the time domain series [34], but with no formal controller derivations for Volterra series order higher than three. The use of basis function expanded models has been similarly limited to the second order case [92], [91].

The contributions of this chapter are as follows. Firstly, a summary of previous achievements in Volterra series model-based control is provided. A flexible nonlinear state space representation is proposed for the arbitrary order Volterra-Laguerre model structure, using Kronecker algebra to define the nonlinear output mapping. A Volterra-Laguerre MPC algorithm is then developed using a typical quadratic objective function. When the move horizon is limited to 1 (i.e. constant over the prediction horizon), the MPC optimization is shown to be reduced to a problem of finding and evaluating polynomial roots, regardless of the model's series order. Extensions to the algorithm are also developed which can handle input constraints and remove steady-state tracking errors. The stability properties of the algorithm are explored, and the relationship between the input penalty term and control stability is discussed in detail.

10.2 History of Volterra series model-based control

In the 1990s, results began to appear in the control theory literature which used Volterra series models and their variants for control design. Some of the first methods to be developed for Volterra series model-based control were proposed in [74] and [35], with a more complete discussion later included in [34]. The methods rely on a decomposition of the Volterra series into its linear and nonlinear components, such

that the input/output relationship can be expressed as

$$y = (\mathbf{L} + \mathbf{N})[u] \quad (10.1)$$

where \mathbf{L} is the linear operator corresponding to first order Volterra kernel h_1 , and \mathbf{N} is the nonlinear operator built from the higher order kernels, h_2, h_3 , etc. Model-based control typically utilizes an approximate inverse of the plant to decide the control action, and a rearrangement of the operator in (10.1) will produce the inverse expression:

$$(\mathbf{L} + \mathbf{N})^{-1} = (\mathbf{I} + \mathbf{L}^{-1}\mathbf{N})^{-1}\mathbf{L}^{-1}. \quad (10.2)$$

The expression on the right hand side of (10.2) is easily shown to be realized by the feedback structure in Figure 10.1. Thus, only the linear kernel is required to be (approximately) inverted, which was already a standard task in linear model-based control at the time of development. Combining this model inverse with the typical filters required for realizability and robustness, an effective controller could be designed.

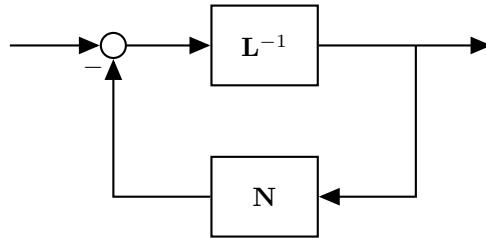


Figure 10.1: Negative feedback structure which can be used to realize the inverse of a plant with linear and nonlinear components

The model-based control results in [35] have been applied to a number of practical control problems, including glucose levels in a simulated diabetic patient [105], polymerization processes [75], [71], the temperature of a greenhouse [51] and airflow in a fuel cell [50]. In each of these problems, the Volterra series model was limited to the second order or a simplified third order structure, as these are the only orders at which analytic results are available for the control design, and no regularized methods existed for robust identification of more complex higher order models.

The use of Laguerre expansions has also been discussed in the literature for Volterra series model-based control. Indeed, Volterra-Laguerre models were applied in adaptive control as early as 1993, in [36]. The basis function approach is attractive for the same reasons which were highlighted for identification in Chapter 5, namely that an appropriate choice of the basis functions can result in a dramatic reduction in the number of parameters required to describe the system. A model-based predictive controller for second order Volterra-Laguerre models was described in [37], and extended in [92], where it was applied to a bioreactor process. The

method relies on a simplified state space model for the second order series, which neglects the constant kernel h_0 , and restricts all kernels to the same Laguerre basis for expansion. Further extensions to the control design were provided in [91], to accommodate larger move horizons, and in [2], for two-input/two-output systems. None of these methods are able to be extended to models with a higher series order.

10.3 A flexible state space representation

In this section, a nonlinear state space model will be developed for the Volterra series expanded by Laguerre basis functions. The model should be flexible enough to consider a separate basis at each nonlinear order of the series, and the constant kernel contribution should be included. The standard formulation for such a model (5.7), is repeated in (10.3) for convenience,

$$\begin{aligned} y^0(t) &= \alpha_0 + \sum_{m=1}^M y_m(t), \\ y_m(t) &= \sum_{i_1=1}^{\mathcal{B}_m} \dots \sum_{i_m=1}^{\mathcal{B}_m} \alpha_m(i_1, \dots, i_m) \prod_{j=1}^m u_{m,i_j}^f(t). \end{aligned} \quad (10.3)$$

Recall that $u_{m,l}^f(t)$ refers to a filtered version of the input, $u(t)$, where the filter is the l^{th} basis function of the m^{th} order basis. Laguerre basis functions are used in this chapter, where the filters are given in the z -domain by

$$F_{m,l}(z) = \frac{\sqrt{1 - |a_m|^2}}{z - a_m} \left(\frac{1 - a_m z}{z - a_m} \right)^{l-1}, \quad a_m \in (-1, 1). \quad (10.4)$$

To express the arbitrary order Laguerre-expanded model (10.3) in a state space framework, it will be necessary to define the Kronecker product and its associated exponent.

Definition 10.1 (Kronecker product). *The Kronecker product of two matrices, $X \in \mathbb{F}^{s \times t}$ and $Y \in \mathbb{F}^{u \times v}$, is defined as*

$$X \otimes Y = \begin{bmatrix} X(1,1) \cdot Y & \dots & X(1,t) \cdot Y \\ \vdots & \ddots & \vdots \\ X(s,1) \cdot Y & \dots & X(s,t) \cdot Y \end{bmatrix} \in \mathbb{F}^{su \times tv}, \quad (10.5)$$

where $X(i,j)$ denotes the i,j^{th} element of matrix X , and \mathbb{F} is an arbitrary field.

Definition 10.2 (Kronecker exponent). *The Kronecker exponent of a matrix X is denoted by $[X]^{\otimes m}$, $m \in \mathbb{N}$, and defined as $m - 1$ repeated Kronecker products of X*

with itself, i.e.

$$[X]^{\otimes m} = X \underbrace{\otimes X \otimes \dots \otimes X}_{m-1 \text{ products}} \quad (10.6)$$

Special cases are defined analogously to the scalar exponent: $[X]^{\otimes 1} = X$ and $[X]^{\otimes 0} = 1$.

Another function involving the Kronecker product will also be defined for use in the MPC derivation.

Definition 10.3 (Kronecker permutation-sum function). *For arbitrary matrices X_1, X_2, \dots, X_m , the function $\mathcal{K}_m(X_1, X_2, \dots, X_m)$ is defined as the sum, over all unique orderings of the matrix arguments, of the total Kronecker product of all arguments. For example,*

$$\mathcal{K}_3(X, Y, Z) = X \otimes Y \otimes Z + Y \otimes X \otimes Z + Z \otimes Y \otimes X.$$

Remark 10.4. It is useful to note that $\mathcal{K}_m(X, X, \dots, X) = [X]^{\otimes m}$.

Additionally, a vectorization operator is defined, that will be applied to the basis function kernels, $\alpha_m(i_1, \dots, i_m)$, in the state space model.

Notation 10.5 (vec(\cdot) operator). *For any m -dimensional tensor, $X(i_1, \dots, i_m) \in \mathbb{R}^{N \times \dots \times N}$, the operator $\text{vec}(X)$ is defined as the standard vectorization operator. A row vector output convention will be used for $\text{vec}(\cdot)$ in this chapter.*

Now a state space model can be constructed to capture the expressions (10.3) and (10.4). Rather than using a single state vector, individual state vectors will be defined for each nonlinear order, m . The state vector, $x_m(t)$, at a given order contains appropriate Laguerre filtered inputs at sample t , i.e.

$$x_m(t) = \begin{bmatrix} u_{m,1}^f(t) & u_{m,2}^f(t) & \dots & u_{m,\mathcal{B}_m}^f(t) \end{bmatrix}^T. \quad (10.7)$$

The full state space model can now be expressed as follows,

$$x_m(t+1) = A_m x_m(t) + B_m u(t), \quad m = 1, 2, \dots, M, \quad (10.8)$$

$$y^0(t) = \alpha_0 + \sum_{m=1}^M \text{vec}(\alpha_m) \cdot [x_m(t)]^{\otimes m}. \quad (10.9)$$

The state equation (10.8) is seen to be split into independent linear equations for each state vector. The matrices A_m and B_m at each order should be constructed so

as to carry out the Laguerre filtering operation at that order, i.e.

$$A_m = \begin{bmatrix} a_m & 0 & 0 & \dots & 0 \\ r_m & a_m & 0 & \dots & 0 \\ -a_m r_m & r_m & a_m & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-a_m)^{q_m-2} r_m & (-a_m)^{q_m-3} r_m & \dots & r_m & a_m \end{bmatrix}, \quad (10.10)$$

$$B_m = \sqrt{r_m} \cdot [1 \ -a_m \ a_m^2 \ \dots \ (-a_m)^{\mathcal{B}_m-1}]^T, \quad (10.11)$$

with $r_m = 1 - a_m^2$.

In the output equation, (10.9), there is a static nonlinear map from the state vectors to the model output, y^0 . For each nonlinear order, m , multiplication of the vectorized basis function kernel with a Kronecker exponent of the state vector allows every m^{th} order state product to be matched with its corresponding expansion coefficient.

Remark 10.6. *The model developed here in (10.8) and (10.9) is a more flexible representation than the Volterra-Laguerre state space structure used in [92] and [91]. Firstly, here the maximum nonlinear order is not limited to $M = 2$, and can be made arbitrarily large using the Kronecker algebra framework. Furthermore, a separate basis can be used for each nonlinear order, since expanding all Volterra kernels with the same Laguerre basis is almost always suboptimal from an accuracy standpoint [19]. Finally, the constant kernel contribution, α_0 , is explicitly considered here.*

10.4 Model predictive control design

A model predictive controller will be designed in this section using the proposed state space Volterra-Laguerre model. The prediction horizon, p , for the scheme is left arbitrary here, however the move horizon is fixed at $\mu = 1$, i.e. the input is assumed constant within the prediction window during each objective optimization. The move horizon is restricted such that the MPC optimization problem is able to be solved in a computationally efficient manner, as will become clear in the sequel. We consider a typical quadratic objective function, which for $\mu = 1$ can be written as,

$$\min_{u(t)} \left\{ \sum_{j=1}^p [y^0(t+j|t) - r(t)]^2 + \Gamma_u [u(t) - u(t-1)]^2 \right\} \quad (10.12)$$

subject to $u(t+j) = u(t) \ \forall j \in 1, 2, \dots, p$.

The objective contains two penalty terms. The first term penalizes the difference between the output and desired reference $r(t)$ over the prediction horizon. The second term penalizes large input changes, and the relative dominance of this term can be scaled using the design parameter Γ_u . The input penalty term is typically included for reasons of robustness.

10.4.1 State space prediction model

Given the $\mu = 1$ move horizon restriction adopted in the objective function, the state space model in (10.8) and (10.9) can be easily modified to allow prediction of the output at any future sample time within the prediction window. The prediction model for this case can be expressed as,

$$x_m(t+j) = A_m^j x_m(t) + \bar{A}_m^{(j)} B_m u(t) \quad m = 1, 2, \dots, M, \quad (10.13)$$

$$y^0(t+j) = \alpha_0 + \sum_{m=1}^M \text{vec}(\alpha_m) \cdot [x_m(t+j)]^{\otimes m}, \quad (10.14)$$

where $j > 0$ is the number of samples forward in the prediction from the current sample, t , and $\bar{A}_m^{(j)} = (A_m^{j-1} + A_m^{j-2} + \dots + I)$.

10.4.2 Minimizing the objective function

In order to choose the control input at any given sample, t , the objective function (10.12) should be minimized. Denoting this optimal input by $u^*(t)$, we have that

$$u^*(t) = \arg \min_{u(t)} \left\{ \sum_{j=1}^p [y^0(t+j|t) - r(t)]^2 + \Gamma_u [u(t) - u(t-1)]^2 \right\} \quad \forall t. \quad (10.15)$$

Now, noting that the minimizer $u^*(t)$ will correspond to a stationary point of the objective function in (10.15), and introducing the shorthand notation,

$$X_j = y^0(t+j|t) - r(t), \quad (10.16)$$

we have that

$$u^*(t) \in \left\{ u(t) : \frac{\partial}{\partial u(t)} \left(\sum_{j=1}^p [X_j]^2 + \Gamma_u [u(t) - u(t-1)]^2 \right) = 0 \right\} \quad (10.17)$$

$$\implies u^*(t) \in \left\{ u(t) : \sum_{j=1}^p X_j \cdot \frac{\partial}{\partial u(t)} X_j + \Gamma_u [u(t) - u(t-1)] = 0 \right\}. \quad (10.18)$$

Examining first the term $X_j = y^0(t+j|t) - r(t)$, and substituting the prediction equations (10.13) and (10.14) into (10.16), it can be seen that X_j is a polynomial in

$u(t)$ of order M ,

$$X_j = \gamma_{M,j} [u(t)]^M + \dots + \gamma_{2,j} [u(t)]^2 + \gamma_{1,j} u(t) + \gamma_{0,j} \quad (10.19)$$

where the polynomial coefficients are given by,

$$\begin{aligned} \gamma_{0,j} &= \alpha_0 + \sum_{m=1}^M \text{vec}(\alpha_m) \cdot [A_m^j x_m(t)]^{\otimes m} - r(t) \\ \gamma_{1,j} &= \sum_{m=1}^M \text{vec}(\alpha_m) \cdot \mathcal{K}_m(\bar{A}_m^{(j)} B_m, \underbrace{A_m^j x_m(t), \dots, A_m^j x_m(t)}_{m-1 \text{ times}}) \\ \gamma_{2,j} &= \sum_{m=2}^M \text{vec}(\alpha_m) \cdot \mathcal{K}_m(\bar{A}_m^{(j)} B_m, \bar{A}_m^{(j)} B_m, \underbrace{A_m^j x_m(t), \dots, A_m^j x_m(t)}_{m-2 \text{ times}}) \\ &\vdots \\ \gamma_{M,j} &= \text{vec}(\alpha_m) \cdot [\bar{A}_M^{(j)} B_M]^{\otimes M}, \end{aligned} \quad (10.20)$$

and where $\mathcal{K}_m(\cdot)$ is the function introduced in Definition 10.3. Thus, the term $X_j \cdot \frac{\partial}{\partial u(t)} X_j$ will also be polynomial,

$$X_j \cdot \frac{\partial}{\partial u(t)} X_j = \beta_{2M-1,j} [u(t)]^{2M-1} + \dots + \beta_{2,j} [u(t)]^2 + \beta_{1,j} u(t) + \beta_{0,j}, \quad (10.21)$$

where the $\beta_{i,j}$ coefficients can be obtained from a convolution of the X_j coefficient vector with its vector of derivatives, i.e.

$$\begin{aligned} &[\beta_{2M-1,j}, \beta_{2M-2,j}, \dots, \beta_{1,j}, \beta_{0,j}] \\ &= [\gamma_{M,j}, \gamma_{M-1,j}, \dots, \gamma_{0,j}] * [M\gamma_{M,j}, (M-1)\gamma_{M-1,j}, \dots, \gamma_{1,j}]. \end{aligned} \quad (10.22)$$

Now substituting (10.21) into (10.18), we obtain a final result for the objective function minimizer,

$$\begin{aligned} u^*(t) \in \left\{ u(t) : \left\{ \sum_{j=1}^p \beta_{2M-1,j} \right\} [u(t)]^{2M-1} + \dots + \left\{ \sum_{j=1}^p \beta_{2,j} \right\} [u(t)]^2 \right. \\ \left. + \left\{ \sum_{j=1}^p \beta_{1,j} + \Gamma_u \right\} u(t) + \left\{ \sum_{j=1}^p \beta_{0,j} - \Gamma_u u(t-1) \right\} = 0 \right\} \end{aligned} \quad (10.23)$$

We can see from (10.23) that for a move horizon $\mu = 1$, the MPC optimization problem reduces to finding the roots of a scalar polynomial, and then evaluating each root in the objective function (10.12) to identify the global minimizer $u^*(t)$. The function evaluations are computationally trivial, and the root-finding task can also be performed in a computationally efficient manner. One common approach to root-finding is to replace the polynomial by its corresponding companion matrix, and compute the eigenvalues of the matrix using established methods such as power

iteration or the QR algorithm [43], [62].

Remark 10.7. *In practice, the controller input is a strictly real quantity, hence only the strictly real roots from the set in (10.23) need to be considered. Furthermore, since the polynomial order ($2M - 1$) is guaranteed to be odd, there exists at least one strictly real root.*

Having derived a solution methodology for the Volterra-Laguerre MPC objective function, the proposed control scheme is summarized in Algorithm 3.

Algorithm 3 MPC scheme for Volterra-Laguerre models

Require: Γ_u , p , $r(t)$, initial condition $u(-1)$, state space model parameters A_m, B_m, α_m

- 1: $t \leftarrow 0$
- 2: **while** Control is running **do**
- 3: **for all** $j \in 1, 2, \dots, p$ **do**
- 4: Compute the X_j polynomial coefficients using (10.20)
- 5: Compute the $X_j \cdot \frac{\partial}{\partial u(t)} X_j$ coefficients using (10.22)
- 6: **end for**
- 7: Compute the coefficients of the minimizer polynomial in (10.23)
- 8: Find all roots of the minimizer polynomial in (10.23)
- 9: Evaluate the objective function (10.12) for all strictly real roots
- 10: Choose $u^*(t)$ as the root which produced the lowest objective function value
- 11: Apply $u^*(t)$ to the physical system
- 12: $t \leftarrow t + 1$
- 13: **end while**

10.4.3 Handling input constraints

Thus far, we have neglected the existence of input constraints in the formulation of our MPC objective function, while actuator limitations always exist in practice. In order to prevent the MPC scheme from selecting input values which violate such practical limits, modifications to the algorithm are necessary. Here we will consider the case in which the input is constrained to a particular closed region of the total scalar space, such that the constrained MPC objective function is

$$\min_{u(t)} \left\{ \sum_{j=1}^p [y^0(t+j|t) - r(t)]^2 + \Gamma_u [u(t) - u(t-1)]^2 \right\} \quad (10.24)$$

subject to $u(t+j) = u(t) \quad \forall j \in 1, 2, \dots, p$,

and $U_{min} \leq u(t) \leq U_{max} \quad \forall t$

with $U_{min}, U_{max} \in \mathbb{R}$ being the endpoints of the constrained region. In this case, the optimization problem is to minimize the objective function *within the constrained region*. This can be achieved by considering both the polynomial roots from (10.23) which lie within the constraint, *and* the constraint endpoints, as input candidates

for minimizing the constrained objective function. The method is summarized in Algorithm 4.

Algorithm 4 Objective function minimization in a constrained region

Require: U_{min} , U_{max} , all roots of the minimizer polynomial in (10.23)

- 1: Form a vector, \mathbf{u} , containing all roots from (10.23)
 - 2: Remove complex-valued entries from \mathbf{u}
 - 3: Remove entries from \mathbf{u} which are less than U_{min} or greater than U_{max}
 - 4: Augment \mathbf{u} with the constraint endpoints, i.e. $\mathbf{u} \leftarrow [\mathbf{u}, U_{min}, U_{max}]$
 - 5: **for all** entries in \mathbf{u} **do**
 - 6: Compute the corresponding MPC objective function value using (10.24)
 - 7: **end for**
 - 8: Choose $u^*(t)$ as the entry of \mathbf{u} which produced the lowest objective function value
-

10.4.4 Disturbance model for offset-free control

It is important to recognize that the MPC scheme proposed in Algorithm 3 performs *open loop* control, since the system output, $y(t)$, is not used for feedback purposes. In the case where the state space model gives an exact description of the system behaviour, and there are no disturbances or noises present, the absence of closed loop feedback is acceptable. Of course, in a practical setting, model errors and other disturbances are inevitable, and this will cause open loop control schemes to exhibit steady-state tracking errors, i.e. there will be an offset between the output and a constant reference signal in steady-state.

In the literature for model predictive control, a disturbance model is often used to ‘close the loop’ and ensure offset-free control [84]. The same technique will be suggested here, by augmenting the Volterra-Laguerre output prediction model (10.14) with an output disturbance as follows,

$$\hat{y}(t+j) = \alpha_0 + \sum_{m=1}^M \text{vec}(\alpha_m) \cdot [x_m(t+j)]^{\otimes m} + \hat{d}_o, \quad (10.25)$$

The predicted output, $\hat{y}(t+j)$, is now a summation of the model output and an output disturbance. The disturbance, \hat{d}_o , is assumed constant over the prediction horizon, but is re-estimated at each control interval using

$$\hat{d}_0(t) = y(t) - \alpha_0 - \sum_{m=1}^M \text{vec}(\alpha_m) \cdot [x_m(t+j)]^{\otimes m}, \quad (10.26)$$

where $y(t)$ is the output measured from the system, and hence the disturbance term captures the difference between the measured and model-predicted output for the current sample time. With the measured output now being used as feedback in the model, the MPC scheme can quantify and correct for any offsets which appear in

steady-state when using a constant reference signal.

With a disturbance model in place, the MPC method from Algorithm 3 can still be used, with one minor modification required in the coefficients of (10.20):

$$\gamma_{0,j} = \alpha_0 + \sum_{m=1}^M \text{vec}(\alpha_m) \cdot [A_m^j x_m(t)]^{\otimes m} - r(t) + \hat{d}_o. \quad (10.27)$$

10.5 Notes on stability

An important issue for any control scheme is whether stability can be guaranteed. Stability in MPC implies that for some desired constant output reference r , the input, output and internal states are all bounded, and the states converge to acceptable equilibrium values. Typically, stability is shown only for the case where the reference and desired state equilibriums are all 0, since this control problem can always be obtained through a suitable change of coordinates on the control problem with arbitrary reference [79].

In this section, the stability of the control scheme in Algorithm 3 will be investigated at several levels of complexity. In the simple case where the system model is linear, a control law can be obtained analytically and used to provide insights about stability, however the analysis becomes more challenging with high order nonlinear models. In all cases, the design parameter Γ_u is the key consideration in generating a stable controller.

10.5.1 Linear FIR case

Starting with a linear model expressed using the finite impulse response, the stability of the proposed MPC approach can be guaranteed for sufficiently large Γ_u . The result is summarized in the Theorem 10.8.

Theorem 10.8. *Assume that the system to be controlled can be exactly described by a linear FIR model, i.e. using the state space model in (10.8) and (10.9) with $M = 1$, $\alpha_0 = 0$, and Laguerre pole $a_1 = 0$. Further assume that the system model is known precisely, and there are no process disturbances present. Under these conditions, the following statement is true:*

$\exists \xi \in \mathbb{R}^+$ such that $\forall \Gamma_u > \xi$, the unconstrained MPC Algorithm 3 will be stable.

Proof. We can assume, without loss of generality, that the reference signal is $r(t) = 0 \ \forall t$.

Now, for a linear model, an explicit control law can be obtained from (10.23):

$$u^*(t) = \frac{-\sum_{j=1}^p \beta_{0,j} + \Gamma_u u(t-1)}{\sum_{j=1}^p \beta_{1,j} + \Gamma_u}, \quad (10.28)$$

where

$$\begin{aligned}\beta_{0,j} &= \gamma_{1,j}\gamma_{0,j}, \\ \beta_{1,j} &= [\gamma_{1,j}]^2,\end{aligned}$$

and

$$\begin{aligned}\gamma_{0,j} &= \left[\alpha_1 A_1^j \right] x_1(t), \\ \gamma_{1,j} &= \left[\alpha_1 \bar{A}_1^{(j)} B_1 \right].\end{aligned}$$

Stability can be assessed by forming the closed loop state equation,

$$x_1(t+1) = A_1 x_1(t) + B_1 u^*(t), \quad (10.29)$$

where $u^*(t)$ is a function of $x_1(t)$ substituted from (10.28), yielding

$$x_1(t+1) = A_{CL} x_1(t) + g, \quad (10.30)$$

where g is a function independent of the states $x_1(t)$.

The closed loop system will be stable if all eigenvalues of the closed loop state evolution matrix, A_{CL} , are inside the unit circle. The form of A_{CL} is easily obtained by substituting (10.28), and noting that for an FIR model we have

$$A_1 = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (10.31)$$

The closed loop matrix then takes the form

$$A_{CL} = \begin{bmatrix} -\frac{K_p(1)}{D_p+\Gamma_u} & -\frac{K_p(2)}{D_p+\Gamma_u} & \dots & -\frac{K_p(\mathcal{B}_1-1)}{D_p+\Gamma_u} & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \quad (10.32)$$

where $D_p = \sum_{j=1}^p [\alpha_1 \bar{A}_1^{(j)} B_1]^2$ and $K_p = \sum_{j=1}^p [\alpha_1 \bar{A}_1^{(j)} B_1] \cdot [\alpha_1 A_1^j]$. The matrix in (10.32) is seen to be in companion form, and thus the eigenvalues are found as the

roots of the following polynomial,

$$[D_p + \Gamma_u] \lambda^{\mathcal{B}_1} + K_p(1) \lambda^{\mathcal{B}_1 - 1} + \dots + K_p(\mathcal{B}_1 - 1) \lambda = 0. \quad (10.33)$$

Finally, Rouché's theorem can be used to bound the eigenvalues within a given radius of the origin [28], and this radius will monotonically decrease as the leading coefficient in (10.33) increases. Therefore there must exist a threshold $\xi \in \mathbb{R}^+$ which, when exceeded by Γ_u , puts all closed loop eigenvalues inside the unit circle, ensuring that the states, input and output all converge to 0. \square

The proof of Theorem 10.8 is constructive, in that it shows (via Rouché's theorem) how to compute the stability threshold, ξ , for the case where a linear FIR model is known. If a linear Laguerre-expanded model is used instead, the main result still holds, but without a direct relationship to the ξ threshold.

Corollary 10.9. *If a linear model with non-zero Laguerre pole, a_1 , is used in the MPC Algorithm 3, then the result in Theorem 10.8 will still hold.*

Proof. Follows trivially from the fact that any linear Laguerre expansion can be transformed to an equivalent impulse response model. \square

10.5.2 Numerical example

To demonstrate the effect of Γ_u on the stability of Algorithm 3, we can consider a simple linear example. The prediction horizon is set as $p = 3$, and the system to be controlled is represented by a linear FIR model with impulse response,

$$\alpha_1 = [2, 1, 2, 10]. \quad (10.34)$$

Allowing Γ_u to vary between 0 and ∞ , the locus of the closed loop eigenvalues can be calculated using (10.33) at each input penalty scaling. The result is shown in Figure 10.2, where the eigenvalue trajectories are shown in black for increasing Γ_u , and the unit circle is outlined in blue. It is clear that the control will be unstable if the input penalty scaling is small enough, since one of the eigenvalues starts outside the circle. However, when Γ_u becomes large enough, all eigenvalues are contained within the unit circle, and the controller is stable. The threshold for this example is $\xi \approx 6$.

10.5.3 General case

The analysis of stability for arbitrary order Volterra-Laguerre models is far more complex than in the linear case. The first issue is that it is not always possible to obtain an explicit control law from (10.23), since no analytic expressions exist for the roots of polynomials of order 5 and higher. Furthermore, even in cases where

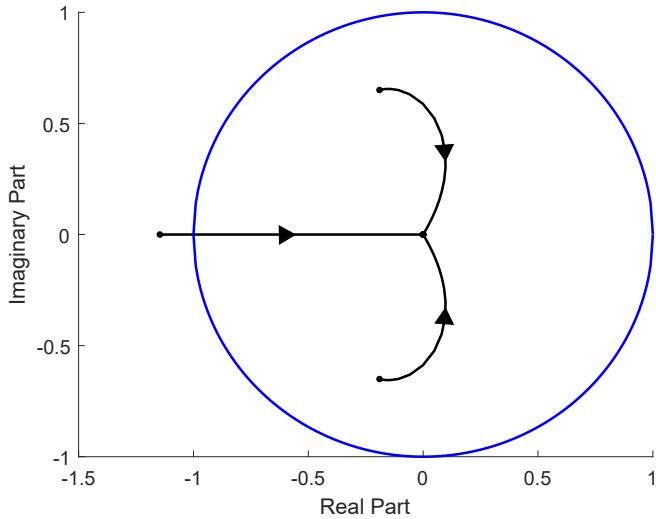


Figure 10.2: Locus of closed loop eigenvalues (black) in the proposed MPC algorithm for a linear FIR example as Γ_u varies from $0 \rightarrow \infty$. The closed loop system is unstable to begin with, but becomes stable once Γ_u is large enough to bring all eigenvalues inside the unit circle (blue).

an analytic expression can be obtained there may be multiple real roots, and the resulting control laws will not be linear functions of the state vector in general.

However, we can still look at the limiting behaviour, where $\Gamma_u \rightarrow \infty$. In this case, (10.23) reduces to

$$u^*(t) \in \left\{ u(t) : \Gamma_u u(t) - \Gamma_u u(t-1) = 0 \right\} \quad (10.35)$$

$$\implies u^*(t) = u(t-1), \quad (10.36)$$

which is trivially stable, since the input is clearly bounded and the system model is stable by definition. This provides a level of confidence that the MPC algorithm will be stable for sufficiently large Γ_u .

10.6 Conclusion

A computationally efficient model predictive control algorithm was proposed in this chapter for systems modeled using the Laguerre-expanded Volterra series. A flexible state space modeling framework was developed to represent Volterra-Laguerre systems of any nonlinear order, then a typical quadratic objective function was employed, with arbitrary prediction horizon and a move horizon of 1. The move horizon was restricted in order to reduce the MPC optimization problem to a polynomial root-finding problem, the solution of which can be computed efficiently online. Suggested modifications to the algorithm were also provided in order to handle input

constraints and enable offset-free reference tracking in steady-state. Finally, the stability of the proposed algorithm was investigated, where the linear case in particular highlights the importance of an appropriately scaled input penalty in the MPC objective function.

CHAPTER

11

Application Examples of Volterra-Laguerre MPC in Water Level Control

The efficacy of the arbitrary order Volterra-Laguerre MPC proposed in Chapter 10 is assessed here through two application examples. Both examples feature the problem of water level control in connected water tanks, where one is considered in simulation while the other is a practical implementation. For each case, the proposed MPC is tested against control using linearized and simplified model structures to reveal the superiority of the method. The disturbance rejection and reference tracking performance are also investigated with respect to the prediction horizon length.

11.1 Introduction

The efficacy of arbitrary order Volterra-Laguerre MPC, proposed in Chapter 10, will be assessed in this chapter through two application examples relevant to the process industries. In each example, the considered problem is control of the fluid level in a nonlinear system of connected water tanks. In such systems, the nonlinear dynamics arise due to Bernoulli's principle, which dictates the flow of water through valves [4]. Additionally, constraints exist on the input due to saturation, and at the output to prevent overflow.

The first application example is based on the cascaded tanks benchmark system from Chapter 6, which is investigated in this chapter via an accurate state space model simulation. Practical implementation of the MPC method is then considered in the second example; a physical system of horizontally coupled tanks with optional disturbance pump in the output tank. In both test applications, the performance of the proposed Volterra-Laguerre MPC is assessed and compared against existing MPC methods using linearized or simplified model structures. In the practical example, the effect of the prediction horizon on reference tracking and disturbance rejection is experimentally verified.

11.2 Simulation study on the cascaded tanks

We return to the cascaded tanks benchmark system, introduced in Chapter 6, to motivate our first MPC application example. A schematic overview of the system setup can be found in Figure 6.1, where $u(t)$ and $y(t)$ will become the control input and output respectively. The results in Section 6.2.5 revealed that a second order Volterra-Laguerre model can provide accurate predictions for the dynamics of the cascaded tanks. In this section, the system will be examined through simulation only, with new Volterra series models being identified prior to the control experiment. The proposed MPC algorithm will be compared against the method proposed in [92] for second order Volterra-Laguerre models, as well as a linear MPC scheme.

11.2.1 Experiment methodology

In order to simulate the cascaded tanks system, we employ a nonlinear continuous-time state space model developed in [90], whose parameters have been estimated using the real benchmark data. For simplicity, we will neglect the possibility of having process noise, measurement noise or tank overflow, such that the model can

be described by,

$$\dot{x}_1 = -k_1\sqrt{x_1} + k_4u, \quad (11.1)$$

$$\dot{x}_2 = k_2\sqrt{x_1} - k_3\sqrt{x_2}, \quad (11.2)$$

$$y = 0.5(x_2 - L), \quad (11.3)$$

where x_1 and x_2 are the water levels (in cm) of the upper and lower tanks respectively, L is the length of the valve, u is the pump input (in V), y is the output water level (in V), and k_1, k_2, k_3, k_4 are the estimated parameters. The pump is unidirectional, such that the input voltage (u) cannot be negative.

The simulation model is first used to perform an identification experiment, where 8000 samples of input/output data are collected. The input is designed as a filtered noise sequence which ranges between 0 and 8V, such that the output moves through the entire range of intended operating levels. Two second order Volterra-Laguerre models are identified for the tanks system, using the same method employed in Chapter 6, i.e. the regularized basis function approach (ReLBF) proposed in Chapter 5.4. The first model takes full advantage of the flexible state space structure, (10.8) and (10.9), developed in the previous chapter, with separate Laguerre bases at each order and a constant α_0 kernel. The second model uses the simplified structure imposed in [92], where each nonlinear order uses the same Laguerre basis and the constant kernel is neglected.

To assess the control performance of the MPC approach proposed in this thesis, the following control methods are applied to the simulated tanks system:

1. **Proposed:** MPC using Algorithms 3, 4 and a full Volterra-Laguerre model with constant kernel and unique Laguerre bases ($\mathcal{B}_1 = 15, \mathcal{B}_2 = 10$).
2. **Simplified:** MPC using Algorithms 3, 4 and the simplified Volterra-Laguerre structure ($\mathcal{B}_1 = \mathcal{B}_2 = 15, a_1 = a_2, \alpha_0 = 0$) used in [92].
3. **Linear:** Linear MPC using the MATLAB 2018a Model Predictive Control toolbox and a linearized plant model around a $y = 5V$ operating point.

The applied reference signal is chosen to be a step sequence covering the full output range. For the linear MPC method, the linearized model is obtained directly from the nonlinear state equations. The prediction horizon is $p = 50$ for all methods, and the move horizon is $\mu = 1$, with sampling period fixed at 4 seconds. The input penalty scaling is set as $\Gamma_u = 1$ in Algorithm 3.

11.2.2 Volterra-Laguerre model estimates

For the full unsimplified model, the first and second order Volterra-Laguerre kernels are shown in Figure 11.1, and the constant contribution is estimated as $\alpha_0 = -6.59$. Note that the estimate for α_0 results from the series expansion of a nonlinear state

space model, and could not have been determined and corrected for using prior system knowledge. The Laguerre poles chosen by the optimization routine are also found to be significantly different, with $a_1 = 0.80$ and $a_2 = 0.45$.

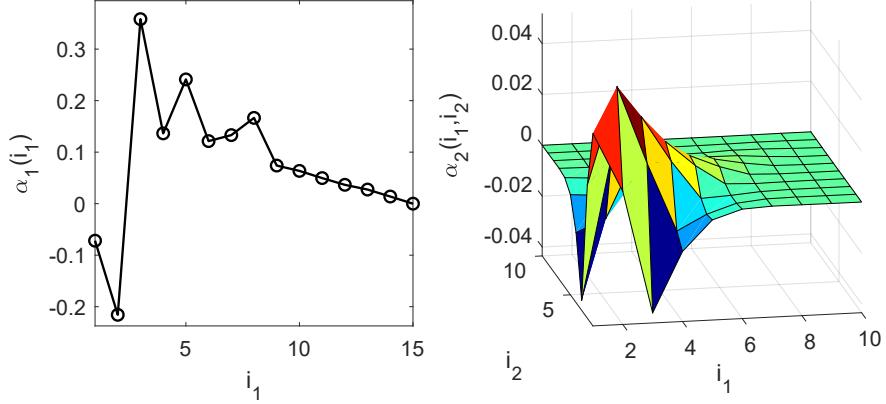


Figure 11.1: Cascaded tanks Volterra-Laguerre kernel estimates for the proposed method

For the simplified model, the two kernels are given in Figure 11.2, with α_0 neglected. Using only a single Laguerre basis, the optimized pole location is 0.68; a clear compromise between the two optimal pole locations estimated in the full model.

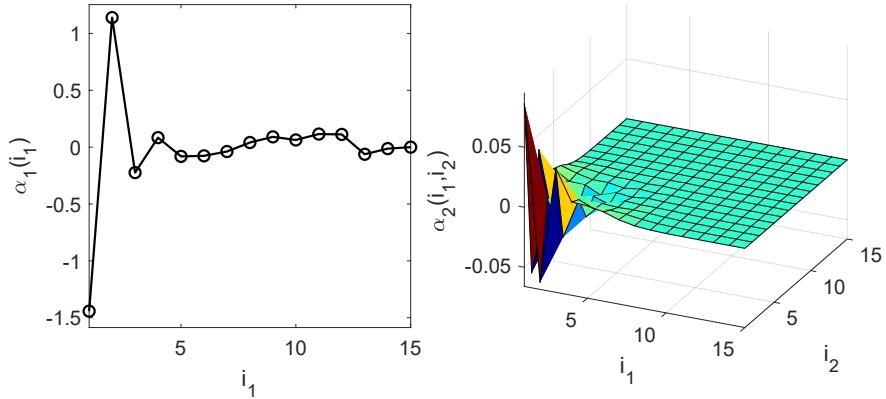


Figure 11.2: Cascaded tanks Volterra-Laguerre kernel estimates for the simplified method

11.2.3 MPC results

For a sequence of step changes applied to the reference, the controlled outputs for each method are shown in Figure 11.3, while the corresponding MPC-computed inputs are given in Figure 11.4. It is clear from the output response that the simplified Volterra-Laguerre structure does not give satisfactory control performance, with slower settling times and large steady state errors at lower water levels where the

nonlinearities are strongest. This can be attributed to the inability of such a constrained model structure to properly capture the dynamics of the nonlinear system. In contrast, the proposed method gives good dynamic and steady state performance for every step change.

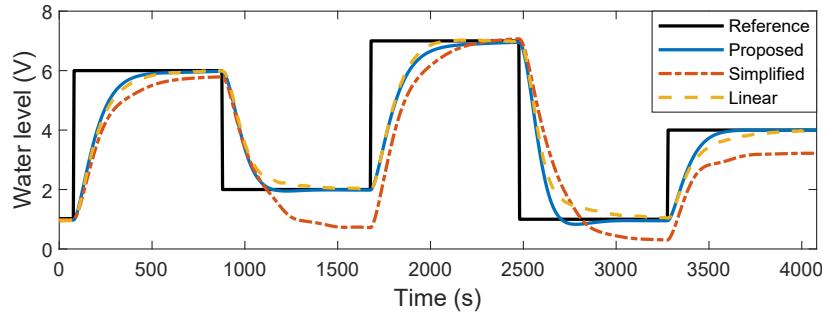


Figure 11.3: Step sequence reference tracking comparison

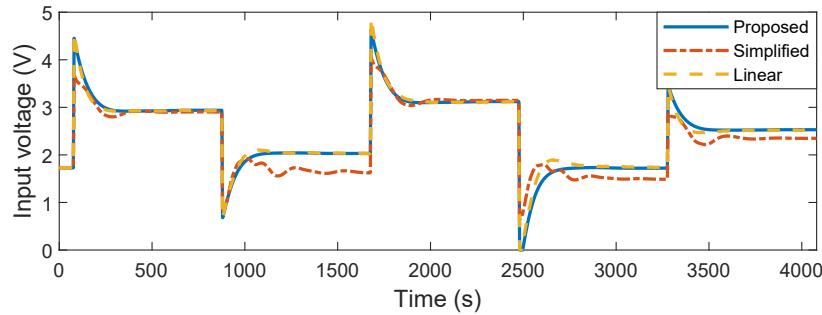


Figure 11.4: MPC inputs for step sequence reference

Comparing the proposed method with the linear MPC approach, it can be seen that the performance is comparable for high water levels, but when the water level measurement drops as low as 1V in the latter half of the record, there is a noticeable degradation in the linear MPC settling times. This result is not surprising, as the fluid dynamics will be most strongly nonlinear when the water level is very close to the exit valve, and thus there will be large deviations from the linearized model obtained from the mid-level of the tanks. The linear model does, however, provide far better results than the simplified Volterra-Laguerre model, and this is due to the use of an exact linearization rather than attempting to estimate nonlinear dynamics over the full tank range without enough model flexibility.

Furthermore, to illustrate the constraint handling performed by Algorithm 4, the MPC objective function is plotted, for the proposed method, at two important instances. Figure 11.5 plots the objective function immediately following the first positive reference step. Here we see the global minimum being rejected for violating an input constraint, and from the remaining candidates, the positive local minimum is chosen for implementation. Figure 11.6 shows the objective function output directly following the second negative reference step, which also features a

global minimum outside of the constrained region. In this case, however, there are no stationary points inside the constraint, and so the constraint boundary is chosen for implementation. This can be observed in Figure 11.4 (at 2500s), where the input briefly sits at the 0V boundary.

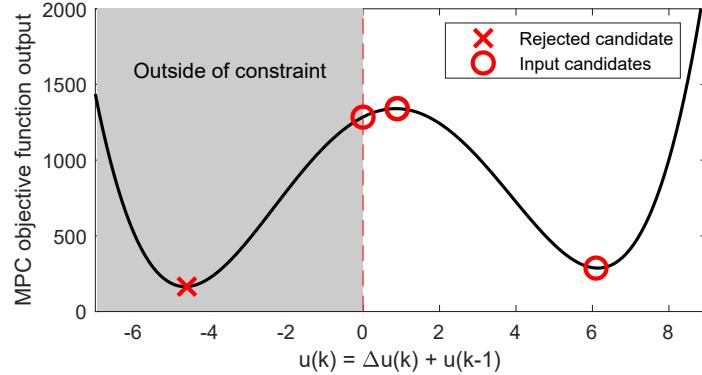


Figure 11.5: Objective function cost after the first positive reference step

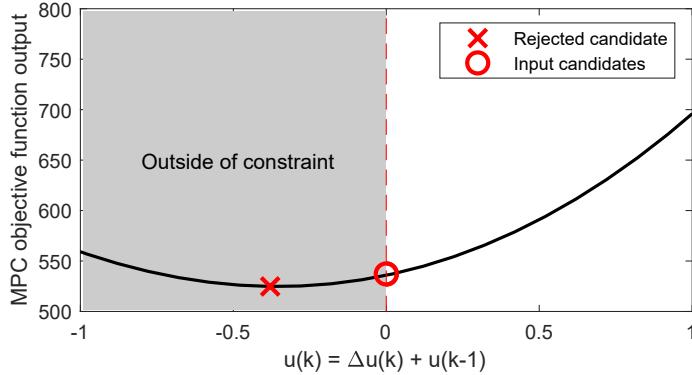


Figure 11.6: Objective function cost after the second negative reference step

11.3 Physical test on a coupled tanks apparatus

For the second application example, we move from simulation to practice, by evaluating the proposed MPC scheme on a physical system of coupled water tanks. The configuration of the physical apparatus, shown in Figure 11.7, is similar in nature to the benchmark system considered in Section 11.2, except that the movement of water from the first to the second tank is now facilitated by a horizontal valve. There is also potential for adding process disturbances via a second pump into the output tank. This section presents results on model identification, water level control and disturbance rejection for the coupled tanks, using the methods proposed in this thesis.

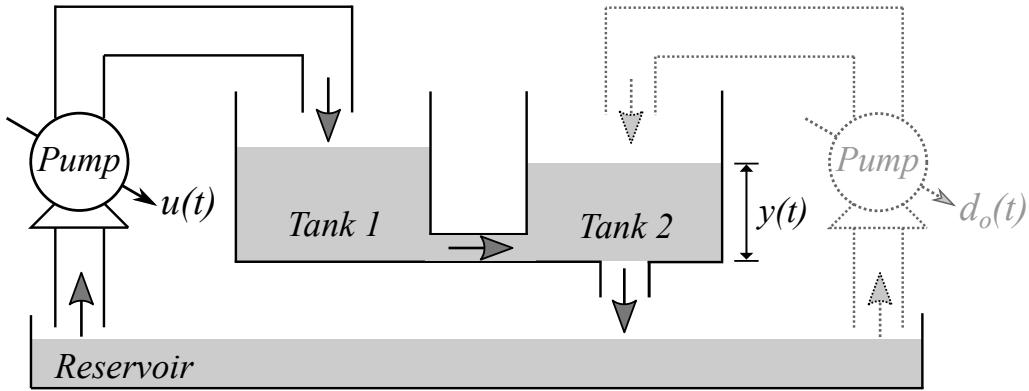


Figure 11.7: Schematic overview of the water flow dynamics in the coupled tanks test apparatus

11.3.1 Experiment methodology

In the identification stage, the excitation input, $u(t)$, to the pump is filtered Gaussian noise ranging between 1.5 and 5.5V, such that the sensor output $y(t)$ in the second tank varies over its full range of 0 to 7V. Data is collected at a sampling period of $T_s = 10\text{s}$, with 8080 samples collected from the input and output and split into two equal parts to produce the estimation and validation datasets shown in Figure 11.8.

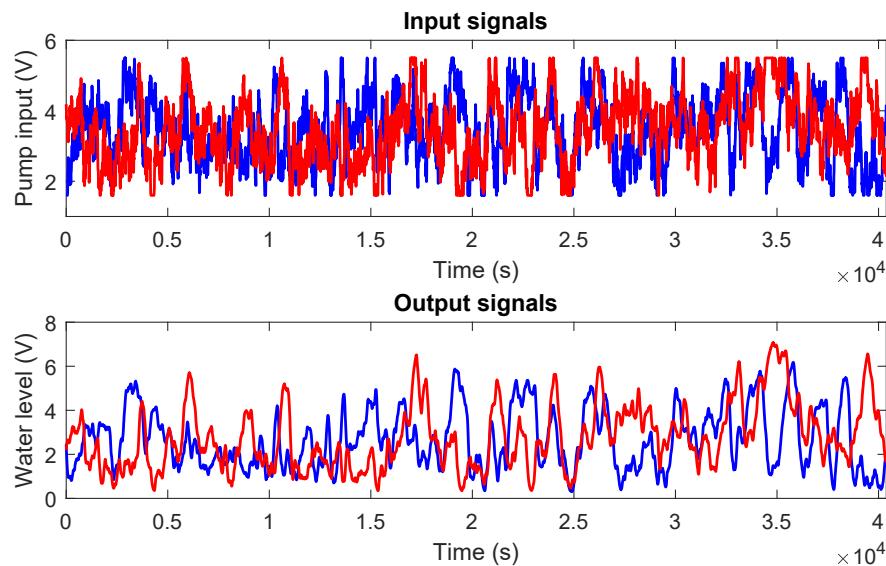


Figure 11.8: Estimation (blue) and validation (red) datasets for identification of the coupled tanks apparatus

Three Volterra-Laguerre models are identified and validated using the data: linear, 2nd order and 3rd order. The linear model contains only the 1st order LBF kernel (no constant offset), while the latter two models contain every kernel up to the maximum order considered. The models are estimated using the regularized

basis function method (ReLBF) proposed in Chapter 5, where separate Laguerre poles are identified for each kernel.

After obtaining the model estimates, the linear and 3rd order models are used to test the performance of the proposed Volterra-Laguerre MPC in Chapter 10. More specifically, control is applied to the tanks for each model using Algorithms 3, 4 and the disturbance model technique in Section 10.4.4. The input penalty is set to $\Gamma_u = 0$ as the penalty is not required for stability of the closed loop system. Evaluation of the control performance is conducted using the following experiments:

1. Reference tracking comparison of linear and 3rd order Volterra-Laguerre MPC using a step sequence reference and prediction horizon $p = 25$
2. Step response comparison for varying prediction horizons in the 3rd order Volterra-Laguerre MPC
3. Disturbance rejection comparison for varying prediction horizons in the 3rd order Volterra-Laguerre MPC

In the third experiment, the disturbance, $d_o(t)$, is a pulse from 0 to 2V and of width 1500s, which is applied to the disturbance pump shown in Figure 11.7 (dotted lines).

11.3.2 Identification results

For each of the models identified using the estimation data, prediction accuracy is assessed using the RMS validation metric (6.1), i.e.

$$e_{RMSt} = \sqrt{\frac{1}{N_v} \sum_{t=1}^{N_v} (y_{mod}(t) - y_v(t))^2}, \quad (11.4)$$

where y_v is the validation output of length N_v , and y_{mod} is the modeled output for the same validation input.

The errors for each identified model are reported in Table 11.1. The linear model is seen to be a relatively poor predictor of system behaviour, while the nonlinear Volterra-Laguerre models have much higher accuracy. The 3rd order model is observed to be the best fit as it exhibits a noticeable 28% decrease in validation error over the 2nd order model.

Table 11.1: Validation metrics for each Volterra-Laguerre model of the coupled tanks

Model	Linear	2 nd order	3 rd order
e_{RMSt}	0.6371	0.0634	0.0457

For the 3rd order model, the basis function kernel estimates $\hat{\alpha}_1$ and $\hat{\alpha}_2$ are plotted in Figure 11.9, and $\hat{\alpha}_0 = 0.774$. The Laguerre pole locations are $a_1 = a_2 = 0.85$ and $a_3 = 0.50$.

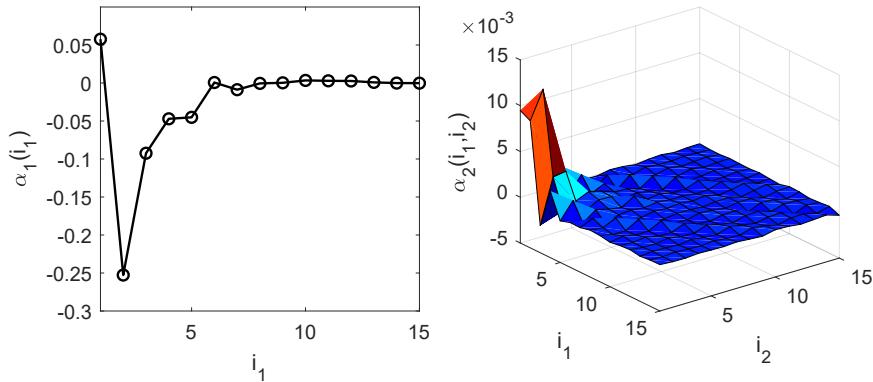


Figure 11.9: Basis function kernel estimates for the 3rd order Volterra-Laguerre model of the coupled tanks

11.3.3 MPC results

Reference tracking comparison

For a step sequence reference signal and $p = 25$, the MPC output results are shown in Figure 11.10. The linear model does not produce desirable control performance, with large overshoots and settling times on most of the reference steps. This result is not surprising given the high validation errors observed for the model in the identification stage, and suggests that a nonlinear model is required for high performance control. The results for the 3rd order model are significantly better, with a desirable response to all of the reference steps. The nonlinear model provides a good characterization of the system dynamics over the entire range of the tank, and MPC could be applied efficiently using the methods of Chapter 10.

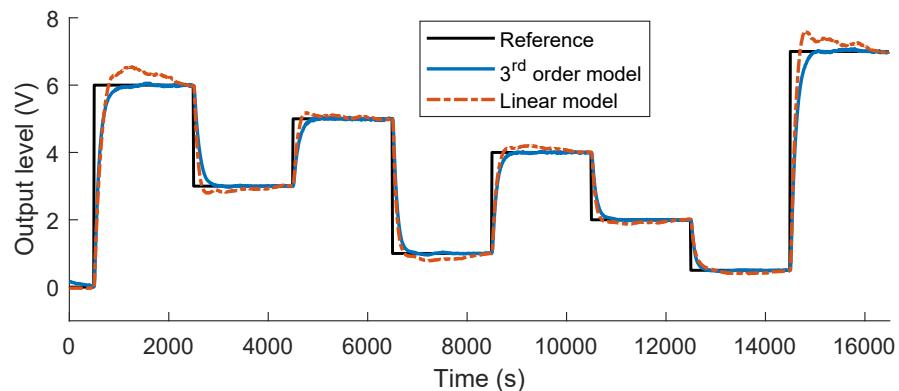


Figure 11.10: Comparison of water levels in Volterra-Laguerre MPC for a step sequence reference and $p = 25$

Step response comparison

Continuing with the 3rd order Volterra-Laguerre model only, the effect of varying the prediction horizon in the MPC Algorithm 3 is explored, with step response results shown in Figure 11.11. Because the move horizon is restricted to 1 in the method, decreasing the prediction horizon is seen to make the control response more aggressive. It is also clear that for the coupled tanks, a prediction horizon of $p = 25$ provides a good balance between low overshoot and short settling time.

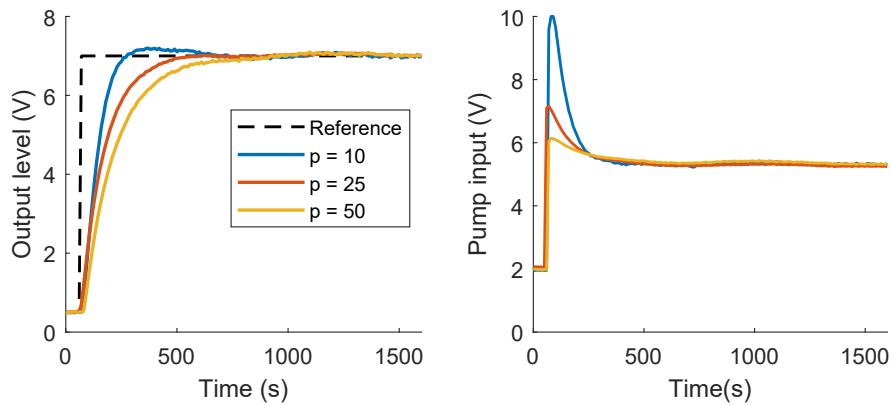


Figure 11.11: Comparison of step responses (left) for varying prediction horizon in Volterra-Laguerre MPC applied to the coupled tanks, and the corresponding MPC inputs (right)

Disturbance rejection

The addition of an output disturbance model to the MPC algorithm allowed steady-state tracking of constant reference signals in the previous two experiments, by tracking and compensating for any model errors. However, the model can also be used to reject physical disturbances such as an additional water flow from the disturbance pump (right side of Figure 11.7). Using the identified 3rd order Volterra-Laguerre model, MPC is applied to the coupled tanks at a constant 3V reference, and a 2V pulse is applied to the disturbance pump to test disturbance rejection properties. In Figure 11.12, the disturbance pulse is plotted alongside the corresponding controlled outputs for three values of the prediction horizon.

The MPC scheme successfully compensates for the disturbance when it enters, and returns to normal operation after the disturbance pulse ends. The results in Figure 11.12 reveal that the prediction horizon will affect the total deviation from the reference, since a lower value of p provides a more aggressive response.

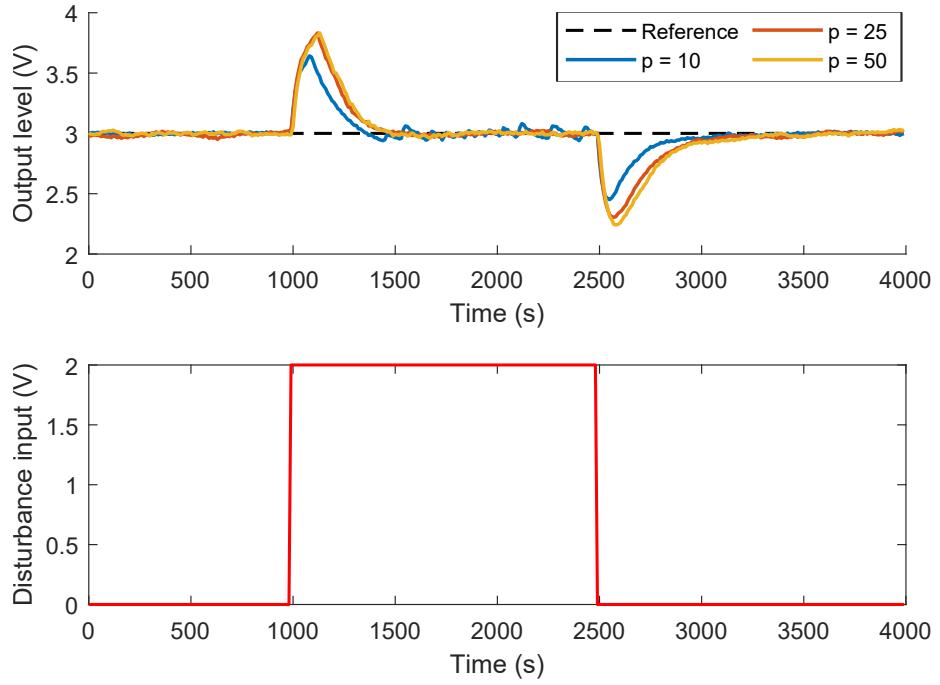


Figure 11.12: MPC outputs (top) for the coupled tanks system during a pulse on the disturbance pump (bottom)

11.4 Conclusion

The identification and MPC methods proposed in this thesis have provided the necessary tools to enable model-based control using high order Volterra-Laguerre models. Despite the multidimensional complexity of such models, MPC was applied effectively and efficiently in this chapter to the problem of water level tracking in nonlinear tank systems. Both experimentally and in simulation, the proposed Volterra-Laguerre MPC was shown to produce desirable results in reference tracking and disturbance rejection, with significant improvements compared to control using linearized and simplified model structures.

CHAPTER

12

Conclusions and Future Work

The content of this thesis focussed on the development of mathematical techniques which can help enable the widespread use of Volterra series models in data-driven modeling and control. This chapter summarizes the main contributions and results achieved in the thesis, and suggests some promising related research directions which can be pursued in the future.

12.1 Summary of contributions

The contributions in this thesis were organized into three distinct parts, with each part focusing on a specific challenge relating to the use of Volterra series models in system identification and control.

12.1.1 Part II - Time domain identification

Contributions in this part of the thesis were aimed at improving the feasibility of high order Volterra series identification in the time domain under arbitrary experimental conditions. For this particular challenge, the achieved results were as follows.

In Chapter 4, an alternate algorithm was developed for hyperparameter optimization in regularized Volterra series estimation. The proposed method uses an expectation-maximization (EM) framework to split the total optimization problem into smaller subproblems in an iterative scheme. When compared to the often used marginal likelihood maximization (MLM) approach, the EM method features better computation time scaling with Volterra series order, which was demonstrated through a Monte Carlo simulation study.

Chapter 5 proposed the use of Bayesian regularization techniques directly on basis function expansions of Volterra kernels. This new concept, which is supported both theoretically and by linear identification literature, combines the parameter reduction benefits of basis function expansions with the accuracy benefits of regularized estimation. An extension to the EM method of Chapter 4 was provided for the basis function case, whereby the covariance hyperparameters and the basis generating parameters can be simultaneously optimized in an iterative fashion. The regularized basis function approach was shown, through numerical examples, to enable feasible model estimation at much higher Volterra series orders than was previously possible with existing methods.

In Chapter 6, further evidence was provided for the performance of regularized basis function estimation of Volterra series models. The method proposed in Chapter 5 was applied to two real datasets, sourced from nonlinear benchmark systems in the system identification community. Basis function models obtained for the first system, a set of vertically cascaded tanks, were shown to provide equivalent prediction accuracy to that of an equivalent time domain model, while the computation times for the estimates were orders of magnitude lower. For the coupled electric drives benchmark, a second order model obtained using this thesis' proposal was shown to outperform a number of popular nonlinear estimation routines.

12.1.2 Part III - Frequency domain identification

This part of the thesis considered Volterra series identification in the frequency domain, where there existed no equivalent Bayesian estimation methods to those

available in the time domain. The contributions were organized into three chapters as follows.

Chapter 7 focussed on estimation using the NOFRF model, and developed a Gaussian process regression (GPR) method for the simultaneous identification of all NOFRFs from a single experiment. The scope of the method was limited to parallel Hammerstein systems, in which case the NOFRFs can be guaranteed to behave like input-independent linear filters. The proposed method was shown through numerical examples to significantly outperform a traditional multilevel excitation approach, while using less data and requiring fewer experimental restrictions.

In Chapter 8, a framework was developed for the direct estimation of generalized frequency response functions (GFRFs) using Bayesian priors. By appropriately transforming the prior covariance structures designed for time domain Volterra series estimation, an equivalent frequency domain method to that of [11] was constructed. The proposed GPR method was shown to produce accurate estimates under rank deficient conditions, and a Monte Carlo study demonstrated the superiority of the direct frequency domain approach in the case where a limited frequency band is of interest.

Chapter 9 explored the output response, in the frequency domain, of nonlinear Volterra systems when they have arbitrary excitation, i.e. they are not in steady-state. Expressions were derived in the chapter for the steady-state and transient contributions to the output spectrum at each nonlinear order. The structure of the transient expressions provided insight which is relevant to frequency domain system identification. In particular, the transients at higher nonlinear orders have a more complex structure due to their increased dependence on the measured input sequence, and this behaviour violates the assumptions of many popular linear methods for transient estimation and removal.

12.1.3 Part IV - Model predictive control

The final part of the thesis investigated the use of Volterra series models in a control context. Historically, such models could only be used for control if they were restricted to very low orders with a simplified structure. The results obtained for this control-related challenge were as follows.

In Chapter 10, a model predictive control (MPC) algorithm was constructed for arbitrary order Volterra series models expressed using Laguerre basis function expansions. Using a new Kronecker algebra framework for the nonlinear state space model, the MPC optimization problem at each time step was shown to reduce to a polynomial root finding problem, which can be solved in a computationally efficient manner. The presence of input constraints and output/model disturbances were also addressed through algorithm extensions.

Chapter 11 provided two application examples for the control method designed in

Chapter 10. The prediction models for each system were obtained once again using the regularized basis function approach proposed in Chapter 5. Control performance for each example was compared with MPC using simplified and linearized models, with positive results.

12.2 Considerations for future work

For the topics discussed in this thesis, there are numerous potential pathways for future research. Some of the promising extensions are identified below.

In the area of time domain identification, there is more to explore with regards to the use of regularized basis function expansions. In Chapter 5, Laguerre and Kautz bases were discussed as two simple but popular orthonormal sets, however there are many other choices for the basis functions which may be advantageous in different contexts. Wavelets in particular have received significant attention already in the literature for Volterra series estimation.

There are also several avenues for analyzing and extending the EM hyperparameter tuning methods in Algorithms 1 and 2. Further analysis can be performed on the modified EM Algorithm 2, to determine the convergence properties of the basis function parameters within the scheme. To increase computational efficiency, it may be useful to look at accelerated EM techniques [82]. There are also two layers of parallel computing potential in the EM algorithms. The first layer comes from the optimization problem in each iteration being partitioned into independent subproblems for each nonlinear order. The second layer resides in the individual non-convex optimizations, which can be performed in a parallel fashion using for example a particle swarm approach [59], [38].

Recursive identification schemes were not considered within this thesis, however the iterative nature of the EM-based methods in Chapters 4 and 5 make them good candidates for conversion to a recursive formulation. Such a formulation would be particularly useful in tracking the dynamics of nonlinear systems with a time-varying nature.

Moving now to the frequency domain setting, there are several open questions remaining. The results in Chapter 8 lead to an equivalent frequency domain method to that of [11], however the method only applies for periodic and steady-state estimation data. It is possible to relax this constraint by incorporating the transient behaviour, described in Chapter 9, into the GPR method. Future work would investigate how best to express the transient behaviour in a Bayesian setting while still maintaining computational feasibility.

There are also opportunities for further analysis of the transient expressions in Chapter 9. While it has been shown that the transients at high nonlinear orders are structurally more complex than linear transients, they can still be locally smooth (in the frequency domain) if the sample size is sufficiently large with respect to the

memory length of the system. This is because a larger sample size implies higher frequency resolution in the output DFT. Thus, another question for future research is whether methods such as LPM and LRM can still be used to remove the transient function when using very large sample sizes.

A final important question for frequency domain identification is whether a regularized basis function approach can also be used in this domain. The orthonormal basis functions considered in Chapter 5 possess frequency domain expressions, and so it should be possible to use these or other appropriate bases to directly estimate a compact basis function model using Bayesian regularization, equivalently to the time domain case.

With respect to the MPC contributions in Chapter 10, several improvements can be made to enrich the proposed control method. First, the state space framework can be adapted to accommodate different basis function expansions, particularly the 2-parameter Kautz case which was shown in Chapter 5 to better capture resonant dynamics. Incorporating the Kautz filters into a state space model is possible (see e.g. [80]), however the structure of the dynamic equation will be different to the Laguerre filter case.

Removing the move horizon restriction, $\mu = 1$, should also be considered in future research. There has been some progress towards this goal in [91], which looked specifically at second order Volterra-Laguerre models. The MPC was implemented for larger move horizons by using a locally linearized plant model for optimizing all but the first sample of the optimal input sequence.

Finally, for arbitrary order models, the explicit theoretical requirements for stability of Algorithm 3 have not been established. It may be possible to obtain such results using Lyapunov theory, or by bounding the sequence of objective function values by an appropriate monotonically decreasing function. There may also be additions or modifications to the algorithm which can help analyze and guarantee its stability without forfeiting the root-finding nature of the optimization problem.

Bibliography

- [1] H. Akçay and B. Ninness. Rational basis functions for robust identification from frequency and time domain measurements. *Automatica*, 34(9):1101–1117, 1998.
- [2] A. L. Antoine and R. S. Parker. An analytical solution to multivariable nonlinear MPC for second-order Laguerre systems. In *Proceedings of 6th IFAC Symposium on Advanced Control of Chemical Processes*, pages 111–116, 2006.
- [3] K. J. Åström and T. Bohlin. Numerical identification of linear dynamic systems from normal operating records. In *Proc. IFAC Symposium on Self-Adaptive Systems*, 1965.
- [4] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 2000.
- [5] E. Bedrosian and S. O. Rice. The output properties of Volterra systems (non-linear systems with memory) driven by harmonic and Gaussian inputs. *Proceedings of the IEEE*, 59(12):1688–1707, 1971.
- [6] S. A. Billings. Identification of nonlinear systems - a survey. *IEE Proceedings D - Control Theory and Applications*, 127(6):272–285, 1980.
- [7] S. A. Billings and S. Y. Fakhouri. Identification of systems containing linear dynamic and static nonlinear elements. *Automatica*, 18(1):15–26, 1982.
- [8] S. A. Billings and K. M. Tsang. Spectral analysis for non-linear systems, part ii: Interpretation of non-linear frequency response functions. *Mechanical Systems and Signal Processing*, 3(4):341–359, 1989.
- [9] G. Birpoutsoukis, P. Z. Csurcsia, and J. Schoukens. Nonparametric Volterra series estimate of the cascaded water tanks using multidimensional regularization. *IFAC-PapersOnLine*, 50(1):476–481, 2017. 20th IFAC World Congress.
- [10] G. Birpoutsoukis, P. Z. Csurcsia, and J. Schoukens. Efficient multidimensional regularization for Volterra series estimation. *Mechanical Systems and Signal Processing*, 104:896–914, 2018.

- [11] G. Birpoutsoukis, A. Marconato, J. Lataire, and J. Schoukens. Regularized nonparametric Volterra kernel estimation. *Automatica*, 82:324–327, 2017.
- [12] G. Birpoutsoukis and J. Schoukens. Nonparametric Volterra kernel estimation using regularization. In *Proc. IEEE Instrumentation and Measurement Technology Conference*, pages 222–227, 2015.
- [13] X. Bombois, G. Scorletti, M. Gevers, P. M. J. Van den Hof, and R. Hildebrand. Least costly identification experiment for control. *Automatica*, 42(10):1651–1662, 2006.
- [14] G. Bottegal, A. Y. Aravkin, H. Hjalmarsson, and G. Pillonetto. Robust EM kernel-based methods for linear system identification. *Automatica*, 67:114–126, 2016.
- [15] G. Bottegal, R. S. Risuleo, and H. Hjalmarsson. Blind system identification using kernel-based methods. *IFAC-PapersOnLine*, 48:466–471, 2015.
- [16] S. Boyd and L. O. Chua. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems*, 32(11):1150–1161, 1985.
- [17] S. Boyd, Y. Tang, and L. Chua. Measuring Volterra kernels. *IEEE Transactions on Circuits and Systems*, 30(8):571–577, 1983.
- [18] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer, 1999.
- [19] R. Campello, G. Favier, and W. do Amaral. Optimal expansions of discrete-time Volterra models using Laguerre functions. *Automatica*, 40:815–822, 2004.
- [20] J. Cao, L. Chen, J. Zhang, and W. Cao. Fault diagnosis of complex system based on nonlinear frequency spectrum fusion. *Measurement*, 46(1):125–131, 2013.
- [21] W-D. Chang. Volterra filter modeling of nonlinear discrete-time system using improved particle swarm optimization. *Digital Signal Processing*, 22(6):1056–1062, 2012.
- [22] T. Chen and L. Ljung. Implementation of algorithms for tuning parameters in regularized least squares problems in system identification. *Automatica*, 49:2213–2220, 2013.
- [23] T. Chen and L. Ljung. Regularized system identification using orthonormal basis functions. In *Proc. of CDC-ECC*, pages 1291–1296, 2015.
- [24] T. Chen, H. Ohlsson, G. Goodwin, and L. Ljung. Kernel selection in linear system identification - Part II: a classical perspective. In *Proc. of CDC-ECC*, pages 4326–4331, 2011.

- [25] T. Chen, H. Ohlsson, and L. Ljung. On the estimation of transfer functions, regularizations and gaussian processes—revisited. *Automatica*, 48(8):1525–1535, 2012.
- [26] C. M. Cheng, Z. K. Peng, W. M. Zhang, and G. Meng. Volterra-series-based nonlinear system modeling and its engineering applications: A state-of-the-art review. *Mechanical Systems and Signal Processing*, 87(A):340–364, 2017.
- [27] L. O. Chua and Y. Liao. Measuring Volterra kernels (II). *International Journal of Circuit Theory and Applications*, 17(2):151–190, 1989.
- [28] J. B. Conway. *Functions of One Complex Variable*. Springer-Verlag, 1978.
- [29] P. Csürösia. *Nonparametric identification of linear time-varying systems*. University Press, 2015.
- [30] A. da Rosa, R. Campello, and W. do Amaral. Choice of free parameters in expansions of discrete-time Volterra models using Kautz functions. *Automatica*, 43:1084–1091, 2007.
- [31] M. A. H. Darwish, G. Pillonetto, and R. Tóth. The quest for the right kernel in Bayesian impulse response identification: The use of OBFs. *Automatica*, 87:318–329, 2018.
- [32] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, pages 1–38, 1977.
- [33] M. Diestler. System identification and time series analysis: Past, present, and future. In *Lecture Notes in Control and Information Sciences*, pages 97–109. Springer Berlin Heidelberg, 2002.
- [34] F. J. Doyle. *Identification and Control Using Volterra Models*. Praxis, 2002.
- [35] F. J. Doyle, B. A. Ogunnaike, and R. K. Pearson. Nonlinear model-based control using second-order Volterra models. *Automatica*, 31(5):697–714, 1995.
- [36] G. A. Dumont and Y. Fu. Non-linear adaptive control via Laguerre expansion of Volterra kernels. *International Journal of Adaptive Control and Signal Processing*, 7(5):367–382, 1993.
- [37] G. A. Dumont, Y. Fu, and G. Lu. Nonlinear adaptive generalized predictive control and applications. In D. Clarke, editor, *Advances in Model-based Predictive Control*, pages 498–515. Oxford University Press, 1994.
- [38] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43, 1995.

BIBLIOGRAPHY

- [39] C. Evans, D. Rees, L. Jones, and M. Weiss. Periodic signals for measuring nonlinear Volterra kernels. *IEEE Transactions on Instrumentation and Measurement*, 45(2):362–371, 1996.
- [40] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the 2001 American Control Conference*, volume 6, pages 4734–4739, 2001.
- [41] J. A. Vazquez Feijoo, K. Worden, and R. Stanway. Associated linear equations for Volterra operators. *Mechanical Systems and Signal Processing*, 19(1):57–69, 2005.
- [42] J. A. Vazquez Feijoo, K. Worden, and R. Stanway. Analysis of time-invariant systems in the time and frequency domain by associated linear equations (ALEs). *Mechanical Systems and Signal Processing*, 20(4):896–919, 2006.
- [43] J. G. F. Francis. The QR Transformation A Unitary Analogue to the LR Transformation—Part 1. *The Computer Journal*, 4(3):265–271, 1961.
- [44] M. O. Franz and B. Schölkopf. A unifying view of Wiener and Volterra theory and polynomial kernel regression. *Neural Computation*, 18(12):3097–3118, 2006.
- [45] P. Gallman. An iterative method for the identification of nonlinear systems using a Uryson model. *IEEE Transactions on Automatic Control*, 20(6):771–775, 1975.
- [46] D. George. *Continuous Nonlinear Systems*. MIT RLE Technical Report, 1959.
- [47] M. Gevers and L. Ljung. Optimal experiment designs with respect to the intended model application. *Automatica*, 22(5):543–554, 1986.
- [48] G. C. Goodwin and R. L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, 1977.
- [49] G. C. Goodwin, M. E. Salgado, and S. F. Graebe. *Control System Design*. Prentice Hall, 2001.
- [50] J. K. Gruber, C. Bordons, and A. Oliva. Nonlinear MPC for the airflow in a PEM fuel cell using a Volterra series model. *Control Engineering Practice*, 20(2):205–217, 2012.
- [51] J. K. Gruber, J. L. Guzmán, F. Rodríguez, C. Bordons, M. Berenguel, and J. A. Sánchez. Nonlinear MPC based on a Volterra series model for greenhouse temperature control using natural ventilation. *Control Engineering Practice*, 19(4):354–366, 2011.

-
- [52] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control*. Springer, 2011.
 - [53] P. Heuberger, P. Van den Hof, and B. Wahlberg. *Modelling and Identification with Rational Orthogonal Basis Functions*. Springer, 2005.
 - [54] R. Hildebrand and G. Solari. Identification for control: Optimal input intended to identify a minimum variance controller. *Automatica*, 43(5):758–767, 2007.
 - [55] B. L. Ho and R. E. Kalman. Effective construction of linear state-variable models from input/output functions. *Reglerungstechnik*, 14(12):545–548, 1965.
 - [56] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
 - [57] S. Ikehara. A method of Wiener in a nonlinear circuit. Technical report, Research Laboratory of Electronics, Massachusetts Institute of Technology, 1951.
 - [58] J. C. P. Jones and S. A. Billings. Recursive algorithm for computing the frequency response of a class of non-linear difference equation models. *International Journal of Control*, 50(5):1925–1940, 1989.
 - [59] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.
 - [60] A. Y. Kibangou and G. Favier. Tensor analysis-based model structure determination and parameter estimation for block-oriented nonlinear systems. *IEEE Journal of Selected Topics in Signal Processing*, 4(3):514–525, 2010.
 - [61] M. J. Korenberg and I. W. Hunter. The identification of nonlinear biological systems: Wiener kernel approaches. *Annals of Biomedical Engineering*, 18(6):629–654, 1990.
 - [62] V. N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Computational Mathematics and Mathematical Physics*, 1(3):637 – 657, 1962.
 - [63] Z. Q. Lang and S. A. Billings. Output frequencies of nonlinear systems. *International Journal of Control*, 67(5):713–730, 1997.
 - [64] Z. Q. Lang and S. A. Billings. Energy transfer properties of non-linear systems in the frequency domain. *International Journal of Control*, 78(5):345–362, 2005.
 - [65] Z. Q. Lang, S. A. Billings, R. Yue, and J. Li. Output frequency response function of nonlinear Volterra systems. *Automatica*, 43(5):805–816, 2007.

- [66] Z. Q. Lang and Z. K. Peng. A novel approach for nonlinearity detection in vibrating systems. *Journal of Sound and Vibration*, 314(3):603–615, 2008.
- [67] J. Lataire and T. Chen. Transfer function and transient estimation by Gaussian process regression in the frequency domain. *Automatica*, 72:217–229, 2016.
- [68] Y. W. Lee and M. Schetzen. Measurement of the Wiener kernels of a non-linear system by cross-correlation. *International Journal of Control*, 2(3):237–254, 1965.
- [69] I. J. Leontaritis and S. A. Billings. Input-output parametric models for non-linear systems part i: deterministic non-linear systems. *International Journal of Control*, 41(2):303–328, 1985.
- [70] L. M. Li and S. A. Billings. Estimation of generalized frequency response functions for quadratically and cubically nonlinear systems. *Journal of Sound and Vibration*, 330(3):461–470, 2011.
- [71] Y. Li and H. Kashiwagi. High-order Volterra model predictive control and its application to a nonlinear polymerisation process. *International Journal of Automation and Computing*, 2(2):208–214, 2005.
- [72] L. Ljung. *System Identification: Theory for the User*. Prentice Hall PTR, 1987.
- [73] J. M. Maciejowski. *Predictive control with constraints*. Pearson Education, 2002.
- [74] B. R. Maner, F. J. Doyle, B. A. Ogunnaike, and R. K. Pearson. A nonlinear model predictive control scheme using second order Volterra models. In *Proceedings of 1994 American Control Conference*, volume 3, pages 3253–3257, 1994.
- [75] B. R. Maner, F. J. Doyle, B. A. Ogunnaike, and R. K. Pearson. Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models. *Automatica*, 32(9):1285–1301, 1996.
- [76] V. Z. Marmarelis. Identification of nonlinear biological systems using Laguerre expansions of kernels. *Annals of Biomedical Engineering*, 21(6):573–589, 1993.
- [77] D. Q. Mayne. Nonlinear model predictive control: An assessment. In *Proceedings of the Fifth International Conference on Chemical Process Control*, 1996.
- [78] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

-
- [79] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000.
 - [80] A. Mbarek, H. Messaoud, and G. Favier. Robust predictive control using Kautz model. In *Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems*, volume 1, pages 184–187, 2003.
 - [81] T. McKelvey and G. Guérin. Non-parametric frequency response estimation using a local rational model. In *16th IFAC Symposium on System Identification*, 2012.
 - [82] G. McLachlan and T. Krishnan. *The EM Algorithm and extensions*. John Wiley and Sons, 2007.
 - [83] K. Mohan and M. Fazel. Reweighted nuclear norm minimization with application to system identification. In *Proceedings of the 2010 American Control Conference*, pages 2953–2959, 2010.
 - [84] K. R. Muske and T. A. Badgwell. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12(5):617–632, 2002.
 - [85] J. G. Nemeth, I. Kollar, and J. Schoukens. Identification of Volterra kernels using interpolation. *IEEE Transactions on Instrumentation and Measurement*, 51(4):770–775, 2002.
 - [86] M. Nikolaou and D. Mantha. Efficient nonlinear modeling using wavelet compression. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 313–334. Birkhäuser Basel, 2000.
 - [87] B. Ninness and F. Gustafsson. A unifying construction of orthonormal bases for system identification. *IEEE Transactions on Automatic Control*, 42(4):515–521, 1997.
 - [88] R. D. Nowak and B. D. Van Veen. Random and pseudorandom inputs for Volterra filter identification. *IEEE Transactions on Signal Processing*, 42(8):2124–2135, 1994.
 - [89] H. Ogura. Orthogonal functionals of the Poisson process. *IEEE Transactions on Information Theory*, 18(4):473–481, 1972.
 - [90] S. Pan and J. S. Welsh. An application of indirect inference to the cascaded tanks nonlinear benchmark. In *Proceedings of the 18th IFAC Symposium on System Identification*, pages 790–795, 2018.

BIBLIOGRAPHY

- [91] R. S. Parker. Nonlinear model predictive control of a continuous bioreactor using approximate data-driven models. In *Proceedings of the 2002 American Control Conference*, volume 4, pages 2885–2890, May 2002.
- [92] R. S. Parker and F. J. Doyle. Nonlinear model predictive control of a continuous bioreactor at near-optimum conditions. In *Proceedings of the 1998 American Control Conference*, volume 4, pages 2549–2553, June 1998.
- [93] Z. K. Peng, Z. Q. Lang, and S. A. Billings. The nonlinear output frequency response function and its application to fault detection. In *Fault Detection, Supervision and Safety of Technical Processes 2006*, pages 36–41. Elsevier Science Ltd, Oxford, 2007.
- [94] G. Pillonetto and G. De Nicolao. A new kernel-based approach for linear system identification. *Automatica*, 46(1):81–93, 2010.
- [95] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682, 2014.
- [96] R. Pintelon and J. Schoukens. *System Identification: A Frequency Domain Approach*. John Wiley and Sons, 2012.
- [97] R. Pintelon, J. Schoukens, and G. Vandersteen. Frequency domain system identification using arbitrary signals. *IEEE Transactions on Automatic Control*, 42(12):1717–1720, Dec 1997.
- [98] R. J. Prazenica and A. J. Kurdila. Volterra kernel identification using triangular wavelets. *Journal of Vibration and Control*, 10(4):597–622, 2004.
- [99] R. J. Prazenica and A. J. Kurdila. Multiwavelet constructions and Volterra kernel identification. *Nonlinear Dynamics*, 43(3):277–310, 2006.
- [100] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [101] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.
- [102] M. J. Reed and M. O. J. Hawksford. Identification of discrete Volterra series using maximum length sequences. *IEE Proceedings - Circuits, Devices and Systems*, 143(5):241–248, 1996.
- [103] D. Rijlaarsdam, P. Nuij, J. Schoukens, and M. Steinbuch. A comparative overview of frequency domain methods for nonlinear systems. *Mechatronics*, 42(Supplement C):11–24, 2017.

-
- [104] R. S. Risuleo, G. Bottegal, and H. Hjalmarsson. A nonparametric kernel-based approach to Hammerstein system identification. *Automatica*, 85:234–247, 2017.
 - [105] J. D. Rubb and R. S. Parker. Glucose control in type i diabetic patients: A Volterra model-based approach. In *Proceedings of the 7th International Symposium on Advanced Control of Chemical Processes*, pages 353–358, 2004.
 - [106] W. J. Rugh. *Nonlinear System Theory: The Volterra-Wiener Approach*. Johns Hopkins University Press, 1981.
 - [107] M. Schetzen. Measurement of the kernels of a non-linear system of finite order. *International Journal of Control*, 1(3):251–263, 1965.
 - [108] M. Schetzen. A theory of non-linear system identification. *International Journal of Control*, 20(4):577–592, 1974.
 - [109] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems*. John Wiley and Sons, 1980.
 - [110] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
 - [111] J. Schoukens, M. Vaes, and R. Pintelon. Linear system identification in a non-linear setting: Nonparametric analysis of the nonlinear distortions and their impact on the best linear approximation. *IEEE Control Systems Magazine*, 36(3):38–69, 6 2016.
 - [112] J. Schoukens, G. Vandersteen, K. Barbé, and R. Pintelon. Nonparametric preprocessing in system identification: a powerful tool. *European Journal of Control*, 15(3):260–274, 2009.
 - [113] M. Schoukens, P. Mattsson, T. Wigren, and J. P. Noël. Cascaded tanks benchmark combining soft and hard nonlinearities. In *Workshop on Nonlinear System Identification Bechnmarks*, pages 20–23, 2016.
 - [114] M. Schoukens, R. Pintelon, and Y. Rolain. Parametric identification of parallel Hammerstein systems. *IEEE Transactions on Instrumentation and Measurement*, 60(12):3931–3938, 2011.
 - [115] P. Schreier and L. Scharf. *Statistical Signal Processing of Complex-Valued Data: The Theory of Improper and Noncircular Signals*. Cambridge University Press, 2010.
 - [116] A. Segall and T. Kailath. Orthogonal functionals of independent-increment processes. *IEEE Transactions on Information Theory*, 22(3):287–298, 1976.

- [117] T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 1989.
- [118] J. G. Stoddard, G. Birpoutsoukis, J. Lataire, and J. S. Welsh. Frequency domain response under arbitrary excitation for fading memory nonlinear systems. *Automatica*, 107:327–332, 2019.
- [119] J. G. Stoddard, G. Birpoutsoukis, J. Schoukens, and J. S. Welsh. Gaussian process regression for the estimation of generalized frequency response functions. *Automatica*, 106:161–167, 2019.
- [120] J. G. Stoddard and J. S. Welsh. Data-driven modeling of a coupled electric drives system using regularized basis function Volterra kernels. In *Intelligent Robotics and Applications*, pages 475–485. Springer International Publishing, 2018.
- [121] J. G. Stoddard and J. S. Welsh. Frequency domain estimation of parallel Hammerstein systems using Gaussian process regression. In *Proceedings of the 18th IFAC Symposium on System Identification*, pages 1014–1019, 2018.
- [122] J. G. Stoddard and J. S. Welsh. Regularized basis function estimation of Volterra kernels for the cascaded tanks benchmark. In *Proceedings of the 18th IFAC Symposium on System Identification*, pages 413–418, 2018.
- [123] J. G. Stoddard and J. S. Welsh. Improved nonlinear model predictive control with Volterra-Laguerre models. In *Proceedings of the 2019 American Control Conference*, pages 1014–1019, 2019.
- [124] J. G. Stoddard, J. S. Welsh, and H. Hjalmarsson. EM-based hyperparameter optimization for regularized Volterra kernel estimation. *IEEE Control Systems Letters*, 1(2):388–393, 2017.
- [125] J. G. Stoddard, J. S. Welsh, and H. Hjalmarsson. Volterra kernel identification using regularized orthonormal basis functions. *IEEE Transactions on Automatic Control*, Under Review, 2018.
- [126] E. E. Sutter. A practical nonstochastic approach to nonlinear time-domain analysis. In *Proceedings in Advanced Methods of Physiological System Modeling*, pages 303–315, 1987.
- [127] R. Tibshirani. Regression shrinkage and selection with the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.
- [128] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics*, 4:1035–1038, 1963.

- [129] J. A. Tropp. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.
- [130] M. P. Vlaar, G. Birpoutsoukis, J. Lataire, M. Schoukens, A. C. Schouten, J. Schoukens, and F. C. T. van der Helm. Modeling the nonlinear cortical response in EEG evoked by wrist joint manipulation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(1):205–215, 2018.
- [131] B. Wahlberg. System identification using Laguerre models. *IEEE Transactions on Automatic Control*, AC-36(5):551–562, 1991.
- [132] B. Wahlberg. System identification using Kautz models. *IEEE Transactions on Automatic Control*, AC-39(6):1276–1282, 1994.
- [133] P. E. Wellstead. *Introduction to Physical System Modelling*. Academic Press, 1979.
- [134] D. T. Westwick and R. E. Kearney. *Identification of Nonlinear Physiological Systems*. John Wiley and Sons, 2003.
- [135] N. Wiener. Response of a non-linear device to noise. Technical report, Radiation Laboratory, Massachusetts Institute of Technology, 1942.
- [136] T. Wigren and M. Schoukens. Coupled electric drives data set and reference models. Technical report, Department of Information Technology, Uppsala University, 2017.
- [137] J. J. Wray and G. G. R. Green. Calculation of the Volterra kernels of non-linear dynamic systems using an artificial neural network. *Biological Cybernetics*, 71(3):187–195, 1994.
- [138] C. F. Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.
- [139] X. Xia, J. Zhou, C. Li, and W. Zhu. A novel method for fault diagnosis of hydro generator based on NOFRFs. *International Journal of Electrical Power and Energy Systems*, 71(Supplement C):60–67, 2015.