KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

SUBSPACE IDENTIFICATION FOR LINEAR, HAMMERSTEIN AND HAMMERSTEIN-WIENER SYSTEMS

Promotor:
Prof. dr. ir. B. De Moor

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Ivan GOETHALS

May 2005

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

SUBSPACE IDENTIFICATION FOR LINEAR, HAMMERSTEIN AND HAMMERSTEIN-WIENER SYSTEMS

Jury:
Prof. dr. ir. P. Van Houtte, voorzitter
Prof. dr. ir. B. De Moor, promotor
Prof. dr. ir. J. Vandewalle
Prof. dr. ir. J. Swevers
Prof. dr. ir. J. Suykens
Prof. dr. ir. J. Schoukens
Dr. ir. H. Van der Auweraer

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Ivan GOETHALS

U.D.C. 681.3*G12

May 2005

©Katholieke Universiteit Leuven – Faculteit Ingenieurswetenschappen

Arenbergkasteel, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

ISBN 90-5682-609-3

U.D.C. 681.3*G12

D/2005/7515/43

Voorwoord

Voor u ligt het resultaat van bijna 5 jaar onderzoek. Stuk voor stuk interessante en leerzame jaren waarin verscheidene onderzoeksvragen werden gesteld en opgelost. Jaren ook, waarin de steun van vele mensen onontbeerlijk was, zij het onder de vorm van creatieve ideeën en suggesties, of voor hun bijdrage tot het scheppen van een aangename werkomgeving.

Zo wens ik in de eerste plaats mijn promotor prof. Bart De Moor te bedanken voor de mogelijkheid die hij me bood om te doctoreren in de onderzoeksgroep SCD. Bart's enthousiasme en vernieuwende ideeën waren, zeker in een beginfaze, essentieel voor het vinden van de gepaste onderzoeksuitdagingen.

Tevens wens ik mijn leescomité te bedanken, prof. J. Vandewalle, prof. J. Swevers, en dr. H. Van der Auweraer, voor de uitstekende begeleiding en de opbouwende kritiek op de uiteindelijke tekst.

Ook een woordje van dank voor de leden van de jury, prof. J. Suykens en prof. J. Schoukens, die niet enkel bereid waren zich vrij te maken op de dag van de verdediging, maar met wie ik ook een uitstekende samenwerking heb kunnen opbouwen in de loop van mijn onderzoek.

Prof. P. Van Houtte verdient mijn welgemeende dank voor het waarnemen van het voorzitterschap van de jury.

Uiteraard wens ik ook het F.W.O. te danken voor de financiele steun.

Dr. B. Cauberghe, dr. L. Mevel, prof. P. Guillaume en dr. P. Verboven wens ik te bedanken voor de vele interessante discussies in het kader van het FliTE project, de interesse in mijn onderzoek, en de concrete suggesties waarvan enkelen terug te vinden zijn in dit proefschrift. Eveneens wens ik prof. A. Benveniste en prof. M. Basseville te bedanken voor het aangename gastverblijf aan de IRISA onderzoeksinstelling te Rennes.

Mijn collega's K. Pelckmans, L. Hoegaerts en T. Van Herpe ben ik zeer erkentelijk voor de excellente samenwerking en de vele interessante inzichten die zij leverden. Vele andere collega's wens ik te bedanken voor hun ondersteuning

en het creëren van een aangename werksfeer binnen onze onderzoeksgroep. Ik denk dan in de eerste plaats aan mensen als Jos, Katrien, Jeroen, Maarten, Ivan, Tony, Bart, Ida, Pela, Ilse, Mieke, Sven, Marcelo, Evelyne, Raf, Bert, Nathalie, Frank, Frizo en Dries.

Verder dient opgemerkt dat dit doctoraat nooit tot stand was gekomen zonder een verzekerde stroom aan gekoelde dranken. Tom B., Axel, Steven B. en Steven V. V. wens ik dan ook te bedanken voor het vullen van de gemeenschappelijke frigo.

Tot slot bedank ik mijn ouders en de rest van mijn familie en vrienden voor hun steun en vertrouwen. Een heel speciaal woord van dank gaat uit naar mijn vrouw An voor al haar lieve zorgen en de vele kleine dingen van elke dag.

Ivan Goethals
Leuven, mei 2005

Abstract

In this thesis we discuss subspace identification algorithms for linear, Hammerstein and Hammerstein-Wiener systems. Although linear subspace identification algorithms have been around for several years, it is shown that under some specific experimental conditions they can break down or yield unreliable results. New solutions to known problems involving linear subspace identification and regularization will be proposed and compared to existing approaches.

In a second part of the thesis, we focus on non-linear subspace identification applied to Hammerstein and Hammerstein-Wiener systems. By combining ideas from Least Squares Support Vector Machines with classical subspace identification algorithms for linear systems it is shown that reliable subspace identification algorithms for Hammerstein and Hammerstein-Wiener systems can be obtained.

Korte inhoud

In deze thesis bespreken we deelruimte-identificatie algoritmen voor lineaire, Hammerstein en Hammerstein-Wiener systemen. Hoewel lineaire deelruimte-identificatie algoritmen reeds meerdere jaren in omloop zijn, werd recent aangetoond dat zij onder bepaalde experimentele omstandigheden kunnen falen of onbetrouwbare resultaten kunnen opleveren. Nieuwe oplossingen voor deze problemen, gesteund op lineaire deelruimte-identifcatie en regularizatie, zullen worden voorgesteld en vergeleken met bestaande benaderingen

In een tweede deel van de thesis zal de aandacht worden toegespitst op niet-lineaire deelruimte-identificatie voor Hammerstein en Hammerstein-Wiener systemen. Door het combineren van ideeën omtrent kleinste kwadraten steunvector algoritmen (LS-SVMs) met klassieke deelruimte-identificatie algoritmen voor lineaire systemen wordt aangetoond dat betrouwbare deelruimte-identificatie algoritmen voor Hammerstein en Hammerstein-Wiener systemen kunnen worden bekomen.

Notation

Parameters

Unless otherwise stated, lowercase symbols will be used in this thesis to denote column vectors. Uppercase symbols are used for matrices. Elements of matrices and vectors are selected as follows:

$A(i,j), A \in \mathbb{R}^{m \times n}$	The element at the i^{th} row and j^{th} column of A
$A(i,:), A \in \mathbb{R}^{m \times n}$	The i^{th} row of a matrix A
$A(:,j), A \in \mathbb{R}^{m \times n}$	The j^{th} column of a matrix A
$A(i:j, k:l), A \in \mathbb{R}^{m \times n}$	The part of A lying within and between rows i and j and columns k and l

Operators

$\triangleq$	Definition

Set of numbers

$\mathbb{R}$	the set of real numbers
$\mathbb{Z}$, $\mathbb{Z}^+$, $\mathbb{Z}_0^+$	The set of integers, non-negative integers, excluding zero

Matrix operations

A^T	transpose of a matrix
$\mathrm{Tr}(A)$	trace of a matrix i.e. sum of its diagonal elements
vec(A)	column-wise vectorization of a matrix
$\mathrm{Col}(A)$	Column space of a matrix A
$\mathrm{Col}(A)^{\perp}$	Orthogonal complement of the column space of a matrix A
$\mathrm{Row}(A)$	Row space of a matrix A
$\mathrm{Row}(A)^{\perp}$	Orthogonal complement of the row space of a matrix A
$\mathcal{N}(A)$	null-space of a matrix A: $Ax = 0, \forall x \in \mathcal{N}(A)$
$\otimes$	Kronecker product, $A \otimes B = [A(i,j)B]$
$\mathcal{P}_A b$	Orthogonal projection of b onto the column space of A
$\mathcal{P}_{\{B\|A\}} c$	Oblique projection of c onto $\mathrm{Col}(B)$ along $\mathrm{Col}(A)$
B/A	Orthogonal projection of $\mathrm{Row}(B)$ onto $\mathrm{Row}(A)$
$C/_{A}B$	Oblique projection of $\mathrm{Row}(C)$ onto $\mathrm{Row}(B)$ along $\mathrm{Row}(A)$

Norms and extreme singular values

$\|x\|_2, x \in \mathbb{R}^n$	2-norm of a vector $\sqrt{\sum_{i=1}^{n} x_i^2}$
$\|x\|_p, x \in \mathbb{R}^n$	p-norm of a vector $(\sum_{i=1}^{n} x_i^p)^{1/p}$
$\|A\|_F, A \in \mathbb{R}^{m \times n}$	Frobenius norm of a matrix $\sqrt{\mathrm{Tr}(AA^T)}$
$\sigma_{\min}(A), \sigma_{\max}(A)$	smallest and largest singular value of a matrix A
$\sigma_1(A), \sigma_2(A)$	First, second singular value of A (when sorted in non-ascending order)

Principal angles and directions

$\theta_{\min}$	Smallest principal angle between two spaces
$\theta_{\min}(A \triangleleft B)$	Smallest principal angle between Row(A) and Row(B)
$\theta_{\max}(A \triangleleft B)$	Largest principal angle between Row(A) and Row(B)

Expectation, covariance, variance

$E\{\}$	expectation operator
$\mathrm{Cov}(), \mathrm{var}()$	covariance, variance operator

Miscelaneous

z	Forward shift operator $zf(t) = f(t+1)$
i	imaginary unit, such that $i^2 = -1$
$\delta_{tk}, t, k \in \mathbb{Z}$	Kronecker delta: $\begin{cases} \delta_{tk} = 1, & t = k \\ \delta_{tk} = 0, & t \neq k \end{cases}$
s.t.	such that

Abbreviations

ARX	linear AutoRegressive model with eXogeneous inputs
CCA	Canonical Correlation Analysis
CVA	Canonical Variate Analysis
KCCA	Kernel Canonical Correlation Analysis
LS-SVM	Least Squares Support Vector Machines
MIMO	Multiple-input / multiple-output
NARX	non-linear AutoRegressive model with eXogeneous inputs
SISO	Single-input / single-output
N4SID	Numerical algorithms for Subspace State Space System IDentification
PI-MOESP	Past-Inputs Multivariable Output-Error State sPace
PO-MOESP	Past-Outputs Multivariable Output-Error State sPace
RBF	Radial Basis Function
SDP	Semi Definite Programming

Contents

Voorwoord i

Abstract iii

Korte inhoud v

Notation vii

Contents ix

Nederlandse samenvatting xv

1 Introduction 1
- 1.1 Subspace identification . . . 1
- 1.2 Positive realness . . . 2
- 1.3 Ill-conditioning in subspace identification . . . 3
- 1.4 Hammerstein and Hammerstein-Wiener identification . . . 4
- 1.5 Contributions . . . 5
- 1.6 Chapter-by-chapter overview . . . 7

2 Linear geometrical tools 11
- 2.1 Linear least-squares . . . 11
  - 2.1.1 Geometric interpretation of the least-squares problem . . 11
  - 2.1.2 The matrix condition number . . . 12
  - 2.1.3 Variance on the estimated parameters . . . 13
  - 2.1.4 Extension of the least-squares condition number to general linear maps . . . 13
- 2.2 The orthogonal projection . . . 14
  - 2.2.1 Condition number of the orthogonal projection . . . 14
  - 2.2.2 Orthogonal projection onto the row-space of a matrix . . 14
- 2.3 The oblique projection . . . 15
  - 2.3.1 Oblique projection onto the column-space of a matrix . . 15
  - 2.3.2 Oblique projection onto the row-space of a matrix . . . 15
- 2.4 Canonical correlation analysis . . . 16
  - 2.4.1 Principal angles between subspaces . . . 16

2.4.2 Statistical theory of canonical correlation analysis 17
2.4.3 Application: calculating the intersection of two row-spaces 18
2.5 A condition number for the oblique projection 18

I Subspace identification for linear systems 21

3 Subspace identification 23
3.1 Introduction 23
3.2 Deterministic realization 25
3.3 Stochastic realization 26
3.4 Intuition behind subspace identification 29
3.4.1 Free responses of a system 29
3.4.2 Obtaining free responses from data 30
3.5 A rigorous derivation of subspace identification 35
3.5.1 System description 35
3.5.2 Notation and input-output equations 36
3.5.3 Orthogonal projection on past and future data 37
3.5.4 Relation to the Kalman filter 38
3.5.5 A unifying theorem 39
3.5.6 Extracting the system matrices using the extended observability matrix 42
3.5.7 Extracting the system matrices using the states 44
3.6 Stochastic subspace identification 47
3.6.1 Problem definition 47
3.6.2 A unifying framework 49
3.6.3 Relation to existing stochastic subspace identification algorithms 50
3.6.4 Extracting the system matrices 51
3.7 Subspace identification for separately parameterized models . . . 52
3.7.1 Decomposition of the system in a deterministic and a stochastic subsystem 52
3.7.2 Identification of the deterministic subsystem using PI-MOESP 54
3.7.3 Identification of the stochastic subsystem 54
3.8 A practical application 55
3.9 Summary 56

4 The positive realness problem 59
4.1 Problem setting 59
4.1.1 The covariance model 59
4.1.2 Positive realness of a covariance model 60
4.1.3 Causes of positive realness problems 61
4.2 Classical approaches to solve the positive realness problem 62
4.2.1 Altering $\widehat{\Lambda}_0$ 62
4.2.2 Altering $\widehat{G}$ 63

4.2.3 Constrained optimization using Semi Definite Programming 64
4.2.4 Using the residuals in a least-squares approach 64
4.3 Imposing positive realness using Tikhonov regularization 65
4.3.1 Regularization on $\widehat{A}$ and $\widehat{C}$ 65
4.3.2 Choosing the regularization parameter 67
4.4 Comparing the presented algorithm to existing techniques 68
4.4.1 Examples 69
4.4.2 Discussion 72
4.5 A practical application 73
4.6 Conclusions 74

5 Ill-conditioning in subspace identification 77
5.1 Introduction 77
5.2 An analysis of ill-conditioning in subspace identification 79
5.2.1 Reasons for ill-conditioning 79
5.2.2 Some parallels with bad conditioning in ARX identification 84
5.3 The orthogonal decomposition method 86
5.3.1 Replacement of the oblique projection by an orthogonal projection 86
5.3.2 A separately parameterized model structure 89
5.4 A new algorithm based on regularization 91
5.4.1 A regularized oblique projection 91
5.4.2 Influence on the obtained projection 92
5.5 Performance of the regularized N4SID 94
5.5.1 Influence of regularization 94
5.5.2 Projection on the input-space 96
5.6 Weighted regularization 98
5.7 A real-life example 101
5.8 Conclusions 102

II Subspace identification for Hammerstein and Hammerstein-Wiener models 105

6 Hammerstein, Wiener and Hammerstein-Wiener systems 107
6.1 The need for structured non-linear models 107
6.2 Hammerstein model identification 108
6.2.1 Iterative approach 109
6.2.2 Stochastic approaches 109
6.2.3 The overparameterization method 109
6.3 Wiener model identification 111
6.4 Hammerstein-Wiener model identification 112

7 Hammerstein ARX identification 115

7.1 Introduction . . . 115
7.2 Least Squares Support Vector Machines for function approximation 116
7.3 Identification of ARX Hammerstein models . . . 117
7.4 An approximative method . . . 120
  7.4.1 Optimization using collinearity constraints . . . 120
  7.4.2 Optimization without collinearity constraints . . . 121
  7.4.3 Projecting the unconstrained solution onto the class of ARX Hammerstein models . . . 122
  7.4.4 Further comments . . . 123
7.5 Extension to the MIMO case . . . 123
7.6 Comparison with existing overparameterization algorithms . . . 126
7.7 Illustrative examples . . . 127
  7.7.1 SISO system . . . 127
  7.7.2 MIMO system . . . 129
7.8 Conclusions . . . 130

8 Hammerstein N4SID identification 133

8.1 Introduction . . . 133
8.2 Extending the N4SID algorithm towards identification of Hammerstein systems . . . 134
  8.2.1 Overparameterization for the oblique projection $\mathcal{O}_i$ . . . 135
  8.2.2 Calculating the oblique projection $\mathcal{O}_{i+1}$ . . . 138
  8.2.3 Obtaining estimates for the states . . . 139
  8.2.4 Extraction of the system matrices and the non-linearity f 139
  8.2.5 Practical implementation . . . 141
8.3 Illustrative examples . . . 141
  8.3.1 Comparison with the Hammerstein ARX approach in Chapter 7 . . . 141
  8.3.2 Comparison with classical subspace over-parameterization approaches . . . 143
8.4 Conclusions . . . 145

9 Hammerstein-Wiener identification using subspace intersection 147

9.1 Introduction . . . 147
9.2 Brief review of the subspace intersection algorithm . . . 148
9.3 Hammerstein-Wiener subspace intersection . . . 149
  9.3.1 Introducing the static non-linearities . . . 149
  9.3.2 Introducing the kernel . . . 150
  9.3.3 From CCA to KCCA: the state estimate . . . 151
  9.3.4 Estimation of A, B and the non-linear function f . . . 152
  9.3.5 Estimation of C, D and the non-linear function g . . . 154
  9.3.6 Practical implementation . . . 156
9.4 Illustrative examples . . . 158
  9.4.1 A SISO system . . . 158

9.4.2 A MIMO system 159
9.5 Conclusions 161

10 Conclusions, future research, and open problems 165

A About the alternative form of the Kalman filter 169
A.1 Derivation of the special form 169

B The Schur complement 171
B.1 The Schur complement of a matrix 171
B.2 Relation to positive definiteness of a matrix 171

C Tikhonov regularization 173
C.1 Tikhonov regularization and its effect on the Hessian 173
C.2 Variance on the obtained solution 174
C.3 Weighted Tikhonov regularization 174

D Proofs 177
D.1 Proof of Lemma 5.1 177
D.2 Proof of Lemma 5.2 179
D.3 Proof of Lemma 5.3 180

Bibliography 181

Curriculum Vitae 195

Publications by the author 197

Deelruimte identificatie voor lineaire, Hammerstein en Hammerstein-Wiener systemen

Hoofdstuk 1: Inleiding

Het onderzoek beschreven in dit proefschrift situeert zich in de wereld van de systeemidentificatie in het algemeen en deelruimte identificatie in het bijzonder. Het doel van systeemidentificatie is het construeren van accurate wiskundige modellen voor complexe dynamische systemen op basis van metingen uitgevoerd op deze systemen.

Veel van de momenteel gebruikte identificatietechnieken kunnen worden geclassificeerd als zogenaamde predictiefout methodes waarbij een gegeven modelstructuur wordt voorop gesteld waarna een aantal vrije parameters zodanig worden gekozen dat de opgemeten data maximaal kan worden verklaard door het model. Een gekend nadeel van predictiefout technieken is dat het op te lossen optimalisatieprobleem over het algemeen niet convex is, en dit zelfs voor de relatief beperkte klasse van lineaire systemen. Bijgevolg bestaat geen garantie dat het optimale minimum gevonden wordt. Daarenboven leidt het inherent iteratieve karakter van de gebruikte optimalisatie-algoritmen tot problemen gerelateerd aan trage convergentie of numerieke instabiliteit.

Voor lineaire systemen leveren deelruimte identificatie algoritmen een welgekomen alternatief. Deelruimte identificatie technieken werden voornamelijk ontwikkeld in het laatste decennium van de voorgaande eeuw en zijn volledig gebaseerd op numeriek robuuste operaties zoals projecties en de singuliere waarden ontbinding. Convergentieproblemen en numerieke instabiliteiten zijn daardoor in principe uitgesloten. Daarenboven maken deelruimte technieken gebruik van toestandsruimtemodellen met als enige parameter de orde van het systeem. Dit in tegenstelling tot de predictiefout methodes die een bepaalde specifieke parameterisatie verwachten die vooropgesteld wordt door de gebruiker. Het resultaat is dan ook dat deelruimte technieken sterk aan populariteit hebben gewonnen over de laatste twee decennia.

Toch blijven ondanks de huidige populariteit en de eerder vermelde numerieke robuustheid van deelruimte technieken enkele belangrijke problemen onopgelost. Zo werd het gedurende de afgelopen jaren duidelijk dat deelruimte algoritmen in bepaalde gevallen onvolledige, of onbetrouwbare resultaten opleveren. Een ander nadeel van deelruimte technieken is dat ze grotendeels beperkt zijn tot de klasse van lineaire systemen. Beide problematieken worden in het proefschrift nader toegelicht.

Hoofdstuk 2: Lineaire geometrische technieken

In dit hoofdstuk overlopen we kort kleinste kwadraten regressie, de orthogonale en schuine projectie, en de conditionering van deze laatste. Gegeven een matrix $A \in \mathbb{R}^{N \times n}$ met $N \geq n$ en $b \in \mathbb{R}^N$, het doel van kleinste kwadraten regressie is het vinden van een schatting $x_{\text{LS}} \in \mathbb{R}^n$ zodat:

$$(x_{\text{LS}}) = \arg\min_x \|Ax - b\|_2. \tag{0.1}$$

De oplossing voor dit probleem is uniek indien en enkel indien A van volle kolom-rang is en wordt gegeven door

$$x_{\text{LS}} = A^\dagger b,$$

waarbij $A^\dagger$ de zogenaamde pseudo-inverse is van A.

De orthogonale en schuine projectie

Lineaire geometrische projecties volgen dadelijk uit het concept van kleinste kwadraten regressie. Er kan immers aangetoond worden dat de oplossing Ax_{LS} met x_{LS} de oplossing van het kleinste kwadraten probleem (0.1) de loodrechte of orthogonale projectie is van de vector b op Col(A). In dit proefschrift wordt echter vooral gewerkt met rij-ruimtes. De orthogonale projectie van de rij-ruimte van een matrix B op de rij-ruimte van A wordt gegeven als

$$B/A = \widehat{X}_A A = BA^\dagger A,$$

met $\widehat{X}_A$ bekomen uit het kleinste kwadraten probleem:

$$(\widehat{X}_A) = \arg\min_{X_A} \|B - X_A A\|_F.$$

De schuine projectie van de rij-ruimte van C op de rij-ruimte van B via de rij-ruimte van A, een centrale operatie in vele deelruimte identificatie algoritmen, wordt op zijn beurt gegeven als

$$C/_{A}B = \widehat{X}_B B = C \begin{bmatrix} A \\ B \end{bmatrix}^\dagger \begin{bmatrix} 0 \\ B \end{bmatrix},$$

met $\widehat{X}_B$ bekomen uit het kleinste kwadraten probleem

$$(\widehat{X}_A, \widehat{X}_B) = \arg \min_{X_A, X_B} \left\| C - \begin{bmatrix} X_A & X_B \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right\|_F .$$

Conditionering van de schuine projectie

De conditionering van de schuine projectie zal een belangrijke rol spelen in de analyse van deelruimte algoritmen. Met $A \in \mathbb{R}^{n_A \times N}$, $B \in \mathbb{R}^{n_B \times N}$, $n_A + n_B \leq N$ en ervan uit gaande dat $n_A \leq n_B$ en $\text{rank}(B) = n_B$ definiëren we het conditiegetal van de lineaire operator in de schuine projectie $C \underset{A}{/} B$ als

$$\text{Cond}_L \left(\begin{bmatrix} A \\ B \end{bmatrix}^\dagger \begin{bmatrix} 0 \\ B \end{bmatrix} \right) = \frac{\sigma_1 \left(\begin{bmatrix} A \\ B \end{bmatrix}^\dagger \begin{bmatrix} 0 \\ B \end{bmatrix} \right)}{\sigma_{n_B} \left(\begin{bmatrix} A \\ B \end{bmatrix}^\dagger \begin{bmatrix} 0 \\ B \end{bmatrix} \right)} = \frac{1}{\sin(\theta_{\min})},$$

met $\theta_{\min}$ de kleinste principale hoek tussen $\text{Row}(A)$ en $\text{Row}(B)$. Er kan aangetoond worden dat het op deze manier gedefinieerde conditiegetal een maat geeft voor de sensitiviteit van de projectie $C \underset{A}{/} B$ aan variaties op C. Het concept van principale hoeken wordt hieronder nader toegelicht.

Principale hoeken en richtingen

Principale hoeken vormen in weze de multidimensionele uitbreiding van de hoek tussen twee vectoren. Het is geweten dat de hoek $a \sphericalangle b$ tussen twee vectoren $a, b \in \mathbb{R}^N$ kan bekomen worden als:

$$\cos[a \sphericalangle b] = \frac{|a^T b|}{\|a\|_2 \|b\|_2}.$$

Deze notie van een hoek wordt als volgt uitgebreid naar hoeken tussen multidimensionele ruimtes. Neem aan dat $S_1 \in \mathbb{R}^{d_1 \times N}$, $d_1 \leq N$ en $S_2 \in \mathbb{R}^{d_2 \times N}$, $d_2 \leq N$ twee rij-ruimtes opspannen in $\mathbb{R}^N$ zodat $\text{rank}(S_1) = r_1$ and $\text{rank}(S_2) = r_2$. We kiezen een eenheidsvector $v_1 \in \mathbb{R}^N$ uit $\text{Row}(S_1)$ en een eenheidsvector $u_1 \in \mathbb{R}^N$ uit $\text{Row}(S_2)$ zodat de hoek tussen beide vectoren wordt geminimaliseerd. De vectoren v_1 en u_1 worden de eerste principale richtingen genoemd en de hoek ertussen de eerste principale hoek $0 \leq \theta_1 \leq \pi/2$. De tweede principale hoek en richtingen kunnen worden bekomen door de selectie van eenheidsvectoren $v_2 \in \text{Row}(S_1)$ en $u_2 \in \text{Row}(S_2)$ loodrecht op respectievelijk v_1 en u_1, en opnieuw zodat de onderlinge hoek minimaal is. Deze procedure wordt herhaald tot $r = \min(r_1, r_2)$ hoeken en bijhorende principale richtingen gevonden zijn.

Hoofdstuk 3: Deelruimte identificatie

Deelruimte identificatie methodes identificeren systemen van de vorm:

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_t, \\ y_t &= Cx_t + Du_t + v_t, \end{aligned} \tag{0.2}$$

waarbij $u_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}^l$ de ingangen en uitgangen van het systeem zijn op tijdstip t. De zogenaamde toestand op tijdstip t wordt genoteerd als $x_t \in \mathbb{R}^n$. Tenzij anders vermeld worden de procesruis en meetruis wit verondersteld met gemiddelde nul en tweede orde momenten gegevens als

$$E\left\{\begin{bmatrix} w_t \\ v_t \end{bmatrix}\begin{bmatrix} w_k^T & v_k^T \end{bmatrix}\right\} = \begin{bmatrix} Q & R \\ R^T & S \end{bmatrix}\delta_{tk}.$$

Verder worden w en v ongecorreleerd verondersteld met de ingangen;

$$E\left\{w_t u_k^T\right\} = 0, \quad E\left\{v_t u_k^T\right\} = 0, \quad \forall t, k.$$

De representatie (0.2) is gekend als de toestandsruimterepresentatie.

Deelruimte identificatie op ingangs/uitgangsdata

Het basisidee achter deelruimte identificatie algoritmen is dat schattingen voor de uitgebreide observeerbaarheidsmatrix en de toestanden van het bestudeerde systeem kunnen bekomen worden door het combineren van een initiële projectie met een singuliere waarden ontbinding. Eens de observeerbaarheidsmatrix en de toestanden bekomen zijn worden de systeem matrices A, B, C en D bekomen door het oplossen van een kleinste kwadraten probleem. Schattingen voor Q, R en S volgen als de residuals van dit probleem. Een deelruimte identificatie algoritme ziet er dan ook typisch als volgt uit:

- Uit ingangs/uitgangsdata worden bepaalde gestructureerde Hankel matrices Y_f, Y_f^-, U_f, U_f^-, W_p, W_p^+ gevormd. De rij-ruimtes van deze matrices worden geprojecteerd door middel van schuine projecties

  $$\mathcal{O}_i = Y_f /_{U_f} W_p, \qquad \mathcal{O}_{i+1} = Y_f^- /_{U_f^-} W_p^+.$$

  Men kan bewijzen dat indien de zo bekomen projecties $\mathcal{O}_i$ en $\mathcal{O}_{i+1}$ rank-deficient zijn, dezen kunnen worden ontbonden in de zogenaamde uitgebreide observeerbaarheidsmatrix en een schatting voor de toestanden

  $$\mathcal{O}_i = \Gamma_i \widehat{X}_i, \qquad \mathcal{O}_{i+1} = \underline{\Gamma_i}\widehat{X}_{i+1}.$$

  Deze stap wordt typisch uitgevoerd door middel van een singuliere waarden ontbinding.

- In een tweede stap worden A, B, C en D berekend. Dit kan op verscheidene manieren gebeuren. Een cruciale observatie is dat indien zowel de in- en uitgangen als de toestanden in (0.2) bekend zijn, het vinden van A, B, C en D in principe neerkomt op het oplossen van een kleinste kwadraten probleem als volgt

  $$(\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}) = \arg \min_{A,B,C,D} \left\| \begin{bmatrix} \widehat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \widehat{X}_i \\ U_{i|i} \end{bmatrix} \right\|_F^2 .$$

  Uit (0.2) volgt ook dadelijk dat schattingen voor Q, R en S kunnen bekomen worden uit de residuals van dit kleinste kwadraten probleem.

Merk op dat er in bovenstaande uiteenzetting steeds van wordt uit gegaan dat opgemeten ingangen aanwezig zijn. Nochtans is het onder bepaalde omstandigheden ook mogelijk schattingen te bekomen voor de systeemmatrices A en C in (0.2) voor systemen zonder ingangen. Het bekomen identificatie probleem staat bekend als het stochastisch identificatieprobleem.

Stochastische identificatie

Het stochastisch identificatieprobleem kan met deelruimte technieken worden opgelost als volgt:

- Projecteer de rij-ruimtes van matrices Y_f, Y_f^-, Y_p en Y_p^+ als

  $$\mathcal{O}_i = Y_f / Y_p, \qquad \mathcal{O}_{i+1} = Y_f^- / Y_p^+ .$$

  Opnieuw kan bewezen worden dat indien de zo bekomen projecties rank-deficiënt zijn, zij kunnen worden ontbonden in de uitgebreide observeerbaarheidsmatrix en bijhorende schattingen voor de toestanden door het gebruiken van volgende relaties:

  $$\mathcal{O}_i = \Gamma_i \widehat{X}_i, \qquad \mathcal{O}_{i+1} = \underline{\Gamma_i} \widehat{X}_{i+1} .$$

- In een tweede stap worden A en C bepaald uit de kleinste kwadraten regressie

  $$(\widehat{A}, \widehat{C}) = \arg \min_{A,C} \left\| \begin{bmatrix} \widehat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} A \\ C \end{bmatrix} \cdot \widehat{X}_i \right\|_F^2 . \tag{0.3}$$

  Opnieuw kunnen schattingen voor Q, R en S bekomen worden uit de residuals van dit kleinste kwadraten probleem. Nochtans zijn de zo bekomen schattingen voor Q, R en S doorgaans niet consistent. Een alternatieve methode bestaat erin een zogenaamd covariantiemodel A, G, C, L_0 te schatten en vervolgens een Riccati probleem van de volgende vorm op te lossen

  $$P = APA^T + (G - APC^T)(\Lambda_0 - CPC^T)^{-1}(G - APC^T)^T , \tag{0.4}$$

waarna Q, R en S kunnen berekend worden als

$$\begin{aligned} Q &= (G - APC^T)(\Lambda_0 - CPC^T)^{-1}(G - APC^T)^T, \\ R &= (G - APC^T). \end{aligned}$$

Als dusdanig wordt een gepast ruismodel gevonden.

Hoofdstuk 4: Het probleem van gebrek aan reële positiviteit

Zoals eerder vermeld bestaan er deelruimte identificatietechnieken voor systemen met en zonder gemeten ingangen. Vooral voor deze laatsten is het zeer belangrijk niet enkel schattingen voor de systeemmatrices A en C te bekomen, maar ook voor de covariantiematices van proces- en meetruis Q, R en S. In [33] werd aangetoond dat stochastische deelruimte algoritmen kunnen falen indien de Riccati vergelijking (0.4) geen positief definiete oplossing P heeft. In dit geval wordt gezegd dat het covariantiemodel A, G, C, L_0 niet reëel positief is.

Bestaande oplossingen voor het gebrek aan reële positiviteit

Gebrek aan reële positiviteit is een relevant probleem in praktische toepassingen. Mede door deze praktische relevantie is het probleem over de laatste jaren actief bestudeerd. Een belangrijk resultaat in dit verband is dat indien het covariantiemodel stabiel is de volgende equivalenties gelden [50]:

- Het covariantiemodel is reëel positief.
- De spectrale densiteit $\Lambda_0 + C(zI_n - A)^{-1}G + G^T(z^{-1}I_n - A)^{-T}C^T$ is positief semi-definiet voor alle z op de eenheidscirkel.
- De Riccati vergelijking (0.4) heeft een positief definiete opossing P.

Uit deze equivalenties kan dadelijk worden afgeleid dat indien L_0 uit het covariantiemodel kunstmatig wordt verhoogd, zodat de spectrale densiteit gegarandeerd positief semi-definiet wordt, het resulterende covariantiemodel wel reëel positief zal zijn. Deze oplossing voor het probleem van de reële positiviteit werd reeds opgetekend in [120]. Evenzo kan worden aangetoond dat een gepaste aanpassing van G tot een reëel positief covariantiemodel zal leiden [139]. Een groot nadeel van deze, en vele andere voorgestelde [102, 147] methodes is dat zij enkel werken indien het covariantiemodel reeds stabiel is. Verder is de performantie niet altijd optimaal.

Opleggen van reële positiviteit d.m.v. Tikhonov regularisatie

In dit proefschrift wordt een nieuwe methode voorgesteld [65] voor het opleggen van reële positiviteit. De methode steunt op het concept van complexiteitscontrole of regularisatie. In de meest brede zin van het woordt staat regularisatie voor de techniek waarbij een optimalisatieprobleem lichtjes wordt aangepast zodat de onzekerheid op de bekomen oplossing (bvb. de variantie op een verzameling van bekomen model parameters) sterk gereduceerd wordt. Hoewel de aanpassing van het optimalisatieprobleem in het algemeen leidt tot het invoeren van een verwachte fout (bias) is de grootte van de totale fout vaak kleiner dan zonder regularisatie, precies dankzij de vermindering van de variantie. Dit concept staat ook bekend onder de naam van de bias/variantie afweging. Een ander voordeel van het gebruik van regularisatie is dat bepaalde voorwaarden kunnen opgelegd worden op de oplossing van een optimalisatieprobleem. Vooral deze laatste eigenschap is uiteraard nuttig voor het oplossen van het probleem van de reële positiviteit.

De voorgestelde oplossing bestaat erin het standaard optimalisatieprobleem (0.3) voor de schatting van A en C te vervangen door

$$(\widehat{A}, \widehat{C}) = \arg\min_{A,C} \left(\left\| \begin{bmatrix} \widehat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} A \\ C \end{bmatrix} \cdot \widehat{X}_i \right\|_F^2 + c\mathrm{Tr} \left(\begin{bmatrix} A \\ C \end{bmatrix} W \begin{bmatrix} A \\ C \end{bmatrix}^T \right) \right),$$

met $c \geq 0$ een positieve scalar en W een positief definiete matrix van geschikte dimensie die voldoet aan $W - \widehat{G}\widehat{\Lambda}_0^{-1}\widehat{G}^T \geq 0$. Dit type van regularisatie wordt ook wel eens Tikhonov regularisatie genoemd. Er kan bewezen worden dat reële positiviteit kan opgelegd worden op het covariantiemodel in stochastische deelruimte identificatie indien c voldoende groot wordt gekozen. Eveneens blijkt de performantie van de voorgestelde methode beter dan deze van eerder gepubliceerde algoritmen.

Hoofdstuk 5: Slecht geconditioneerdheid van deelruimte identificatie problemen

Ondanks het feit dat deelruimte identificatie algoritmen gestoeld zijn op numeriek robuuste geometrische operaties, zoals projecties en de singuliere waarden ontbinding, zijn de ingangs/uitgangs-varianten en vooral het welgekende N4SID algoritme mogelijk slecht geconditioneerd onder bepaalde experimentele omstandigheden. Dit laatste doet zich vooral voor indien de ingangen sterk gekleurd zijn [22, 25].

Twee redenen voor dit fenomeen worden besproken in dit proefschrift. De eerste is van toepassing op het N4SID identificatie algoritme, de tweede reden is ook van toepassing op de meeste andere deelruimte identificatie algoritmen zoals de PO-MOESP [155] en de CVA [94].

Reden 1: Een slecht geconditioneerde schuine projectie

Het N4SID deelruimte identificatie algoritme wordt gedomineerd door een schuine projectie van waaruit de uitgebreide observeerbaarheidsmatrix en de toestanden van het systeem kunnen worden bekomen. Een belangrijke maat voor de conditionering van deze schuine projectie is de volgende:

$$\mathrm{Cond}_L\left(\mathcal{P}^T_{\{W_p^T|U_f^T\}}\right) = \frac{1}{\sin(\theta_{\min})},$$

met $\theta_{\min}$ de kleinste canonische hoek tussen W_p en U_f. Uit [35–37] volgt dadelijk dat deze hoek klein zal zijn indien de ingangen sterk gekleurd zijn. Bijgevolg kan worden verwacht dat deelruimte identificatie algoritmen ondermaats presteren voor dit type ingangen.

Reden 2: Correlatie tussen de stochastische toestand en de ingangen

Deelruimte identificatie algoritmen schatten een interne toestand die zowel de bijdragen van de ingangen van het systeem (de deterministische bijdragen) als de bijdragen ten gevolge van de storingen (de stochastische bijdragen) bevat. Hoewel theoretisch gezien de correlatie tussen het stochastisch gedeelte van de toestand en de ingangen van het systeem nul is, zal dit niet noodzakelijk het geval zijn indien gewerkt wordt met een eindige hoeveelheid meetdata. Er kan worden aangetoond dat onder invloed van sterk gekleurde ingangen een zwakke correlatie tussen de stochastische component van de toestand en de ingangen van het systeem reeds kan leiden tot onbetrouwbare resultaten.

De orthogonale decompositiemethode

Een voorgesteld algoritme om met beide problemen om te gaan is de zogenaamde orthogonale decompositie methode zoals voorgesteld in [26]. In tegenstelling tot de meeste bestaande deelruimte identificatie algoritmen bevat de orthogonale decompositie methode een decompositie van de opgemeten data in een stochastisch en een deterministisch gedeelte, uitgevoerd als

$$\begin{aligned} Y_f^d &= Y_f/\begin{bmatrix} U_p^T & U_f^T \end{bmatrix}^T, \\ Y_f^s &= Y_f/\begin{bmatrix} U_p^T & U_f^T \end{bmatrix}^{T\perp}. \end{aligned}$$

Er kan worden aangetoond dat deze initiële decompositie toelaat het probleem van de zwakke correlaties tussen de stochastische toestand en de ingangen van het systeem te omzeilen. De slecht-geconditioneerdheid van de schuine projectie wordt op zijn beurt vermeden door het vervangen van de schuine projectie door een orthogonale projectie, welke typisch wordt gevonden in algoritmen uit de MOESP klasse. Samenvattend kan gesteld worden dat de uiteindelijke orthogonale decompositie methode de projectie $\mathcal{O}_i = Y_f/_{U_f} W_p$ vervangt door

$$\mathcal{O}_i = Y_f/(U_p/U_f^\perp),$$

wat leidt tot meer accurate schattingen.

Regularisatie ter verbetering van de conditionering

In dit proefschrift bestuderen we een alternatieve benadering dan de orthogonale decompositie methode. Hoewel daarbij nog steeds gesteund wordt op de orthogonale decompositie van de opgemeten data in een stochastisch en een deterministisch deel, wordt de schuine projectie behouden als de sleutel voor het bekomen van de toestand. Het probleem van de slechte conditionering van de schuine projectie wordt aangepakt door het toepassen van regularisatie in de schuine projectie. De schuine projectie wordt bekomen als $\mathcal{O}_i = \widehat{L}_2^\gamma W_p$, waarbij:

$$(\widehat{L}_1^\gamma, \widehat{L}_2^\gamma) = \operatorname*{arg\,min}_{L_1,L_2} \left(\left\| Y_f - \begin{bmatrix} L_1 & L_2 \end{bmatrix} \begin{bmatrix} U_f \\ W_p \end{bmatrix} \right\|_F^2 + \gamma \|L_2 W_p\|_F^2 \right).$$

Het uiteindelijk bekomen algoritme presteert beter dan de orthogonale decompositie methode en levert, gecombineerd met resultaten uit hoofdstuk 4, voldoende bewijs dat regularisatie een nuttige bijdrage kan leveren in de wereld van de systeemidentificatie.

Hoofdstuk 6: Hammerstein, Wiener en Hammerstein-Wiener systemen

Zoals eerder vermeld is een nadeel van veel deelruimte identificatie algoritmen dat zij in toepassing beperkt zijn tot de klasse van lineaire systemen. Nochtans is een uitbreiding van het deelruimte-raamwerk naar bepaalde klassen van niet-lineaire systemen mogelijk. In [51] werd bijvoorbeeld een deelruimte identificatie algoritme voor bilineaire systemen ingevoerd. Een andere interessante ontwikkeling is de introductie van deelruimte identificatie algoritmen voor Hammerstein, Wiener en Hammerstein-Wiener systemen [75, 156, 159].

Hammerstein, Wiener en Hammerstein-Wiener systemen zijn samengesteld uit een lineair dynamisch gedeelte met transfer functie $H(z)$, vooraf gegaan en/of gevolgd door statische niet-lineariteiten f en g respectievelijk, of nog

$$y_t = g(\tilde{y}_t), \qquad \tilde{y}(z) = H(z)\tilde{u}(z), \qquad \tilde{u}_t = f(u_t).$$

Aangezien het dynamische gedeelte van dergelijke systemen lineair is, vormen zij een zeer aantrekkelijk doelwit voor de uitbreiding van lineaire systeemidentificatie algoritmen naar niet-lineaire systemen. Dit terwijl de aanwezigheid van de statische niet-lineariteiten toch toelaat een bredere klasse van gedragingen te beschrijven dan hetgeen mogelijk is door gebruik te maken van lineaire modellen. We beschouwen hieronder de identificatie van Hammerstein, Wiener en Hammerstein-Wiener systemen in iets meer detail.

Hammerstein identificatie

Hammerstein systemen bestaan uit een statische niet-lineariteit f gevolgd door een lineair dynamisch systeem, of nog

$$y(z) = H(z)\tilde{u}(z), \qquad \tilde{u}_t = f(u_t).$$

Technieken voor de identificatie van Hammerstein systemen onderscheiden zich voornamelijk in de manier waarop de statische niet-lineariteit wordt voorgesteld en het optimalisatieprobleem dat uiteindelijk wordt opgelost. Een gekend probleem met de identificatie van Hammerstein systemen is dat de uiteindelijke kostenfunctie doorgaans kruisproducten bevat tussen parameters die de statische niet-lineariteit beschrijven en parameters die het lineaire dynamische systeem beschrijven. Het opleggen van een criterium van maximale waarschijnlijkheid resulteert dan in een zogenaamd bi-convex optimalisatieprobleem waarvoor globale convergentie niet gegarandeerd is [131].

Het bi-convex optimalisatieprobleem wordt typisch opgelost door middel van iteratieve algoritmen, door het maken van stochastische aannames (zoals witheid van de ingangen), of door toepassing van een techniek die gekend staat als overparameterisatie. In deze laatste worden producten van parameters $b_j c_k$ zoals we die kunnen vinden in Hammerstein systemen van de vorm

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} \sum_{k=1}^{n_f} b_j c_k f_k(u_{t-j}) + e_t$$

vervangen door nieuwe parameters $\theta_{j,k} = b_j c_k$ zodat het model lineair wordt in zijn parameters:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} \sum_{k=1}^{n_f} \theta_{j,k} f_k(u_{t-j}) + e_t. \tag{0.5}$$

Voordeel van deze werkwijze is dat het bekomen optimalisatieprobleem convex is en dus eenvoudig kan opgelost worden. Schattingen voor de b_j en c_k worden daarna gevonden door het toepassen van een singuliere waarden ontbinding op:

$$\begin{bmatrix} \hat{\theta}_{0,1} & \hat{\theta}_{0,2} & \dots & \hat{\theta}_{0,n_f} \\ \hat{\theta}_{1,1} & \hat{\theta}_{1,2} & \dots & \hat{\theta}_{1,n_f} \\ \vdots & \vdots & & \vdots \\ \hat{\theta}_{m,1} & \hat{\theta}_{m,2} & \dots & \hat{\theta}_{m,n_f} \end{bmatrix}. \tag{0.6}$$

Het grote voordeel van het gebruik van overparameterisatie is zonder twijfel de bekomen convexiteit, zoals eerder vermeld. Een belangrijk nadeel van de overparameterisatietechniek is dat het aantal te schatten parameters stijgt wat leidt tot een grote variantie op de bekomen resultaten. Tenslotte is geweten dat in bepaalde omstandigheden meerdere oplossingen $\theta_{j,k}$ bestaan die de residuals in (0.5) minimaliseren. Er is dan geen garantie dat de schattingen voor $\theta_{j,k}$ nog steeds voldoen aan $\theta_{j,k} = b_j c_k$, of nog, dat de matrix (0.6) rank-deficiënt is. In Hoofdstuk 7 zullen we zien dat dit probleem kan vermeden worden door het opleggen van zogenaamde centreringsbeperkingen.

Wiener model identificatie

Wiener systemen zijn zeer verwant aan Hammerstein systemen. Zij bestaan uit een lineair systeem gevolgd door een statische niet-lineariteit g, of nog

$$y_t = g(\tilde{y}_t), \qquad \tilde{y}(z) = H(z)u(z).$$

Wiener-systemen worden geïdentificeerd met gelijkaardige technieken als Hammerstein-systemen. We onderscheiden iteratieve technieken, stochastische technieken en overparameterisatietechnieken. In dit proefschrift zullen we niet verder ingaan op Wiener model identificatie. Over het algemeen kan echter gesteld worden dat veel van de technieken besproken in dit proefschrift toepasbaar zijn op Wiener systemen met een inverteerbare functie g.

Hammerstein-Wiener model identificatie

Hammerstein-Wiener systemen worden bekomen door het plaatsen van een Hammerstein-systeem en een Wiener-systeem in cascade. Een statische niet-lineariteit aan de ingang wordt dan gevolgd door een lineair dynamisch systeem en een statische niet-lineariteit aan de uitgang, of nog

$$y_t = g(\tilde{y}_t), \qquad \tilde{y}(z) = H(z)\tilde{u}(z), \qquad \tilde{u}_t = f(u_t).$$

In tegenstelling tot de literatuur rond Hammerstein en Wiener identificatie is de beschikbare literatuur rond Hammerstein-Wiener identificatie eerder beperkt. In [12] wordt een schema uitgewerkt voor de identificatie van SISO (enkele ingang, enkele uitgang) Hammerstein-Wiener systemen op basis van overparameterisatie. Een nadeel van deze methode is dat een specifieke modelstructuur wordt voorop gesteld, hetgeen de praktische toepasbaarheid negatief beïnvloedt. Gebaseerd op [12] werd een meer algemene zogenaamde blinde methode voor de identificatie van SISO systemen voorgesteld in [14]. Een identificatiemethode voor Hammerstein-Wiener MIMO (Meerdere ingangen, meerdere uitgangen) systemen werd voorgesteld in [29,30], maar steunt op eerder beperkende restricties op de ingangen en is bovendien iteratief van aard. Andere bijdragen zoals [48, 166] zijn gelimiteerd tot SISO systemen en/of iteratief van aard.

Er kan dan ook gesteld worden dat heden ten dage geen betrouwbaar MIMO identificatie algoritme voorhanden is dat niet iteratief is en bovendien niet steunt op restrictieve assumpties op de ingangen. Een poging tot het bekomen van een dergelijk algoritme, door het combineren van het kern canonische correlatie analyse raamwerk en het deelruimte intersectie algoritme, zal worden voorgesteld in Hoofdstuk 9.

Hoofdstuk 7: Hammerstein ARX identificatie

In dit hoofdstuk beschouwen we allereerst de identificatie van SISO Hammerstein systemen in ARX vorm (AutoRegressieve modellen met eXterne ingangen):

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j f\left(u_{t-j}\right) + e_t. \tag{0.7}$$

We zullen daarbij gebruik maken van het zogenaamde LS-SVM formalisme (kleinste kwadraten steun-vector machines). Het idee van dit formalisme is dat een in essentie niet-lineair probleem linear kan gemaakt worden door een projectie van meetgegevens in een hoog-, mogelijks oneindig-, dimensionele ruimte. In deze ruimte kunnen dan klassieke lineaire technieken worden toegepast. Deze techniek wordt hieronder toegelicht in het kader van statische regressie of functieschatting.

Kleinste kwadraten steun-vector machines voor functieschatting

Laat $\{(x_t, y_t)\}_{t=1}^{N} \subset \mathbb{R}^d \times \mathbb{R}$ een set van ingangs/uitgangs-trainingsdata zijn met ingang x_t en uitgang y_t. Beschouw het regressiemodel $y_t = f(x_t) + e_t$ waarbij $x_1, \ldots, x_N$ deterministische punten zijn, $f : \mathbb{R}^d \rightarrow \mathbb{R}$ een ongekende gladde functie met beeld in de reële getallen (i.e. Lipschitz continu) is, en de $e_1, \ldots, e_N$ ongecorreleerde random fouten met $E\left[e_t\right] = 0$, $E\left[e_t^2\right] = \sigma_e^2 < \infty$ zijn. Het volgende model wordt verondersteld:

$$f(x) = w^T \varphi(x) + b,$$

waarbij $\varphi(x) : \mathbb{R}^d \rightarrow \mathbb{R}^{n_H}$ een mogelijks oneindigdimensionele ($n_H = \infty$) kenmerkfunctie en $w \in \mathbb{R}^{n_H}$, $b \in \mathbb{R}$. De geregulariseerde kostenfunctie van de LS-SVM [135] wordt gegeven als

$$\min_{w,b,e} \mathcal{J}(w, e) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{t=1}^{n} e_t^2, \tag{0.8}$$

$$\text{s.t.} : y_t = w^T \varphi(x_t) + b + e_t, \quad t = 1, \ldots, N. \tag{0.9}$$

Het relatieve belang van de gladheid van de oplossing ten opzichte van de accuraatheid van de fit aan de data wordt in hoofdzaak bepaald door de scalar $\gamma \in \mathbb{R}_0^+$, waarnaar wordt gerefereerd als de regularisatieconstante.

De uitgevoerde optimalisatie staat gekend onder de naam van richelregressie [68] in de kenmerkruimte. Om het beperkte optimalisatieprobleem op te lossen wordt een Lagrangiaan geconstrueerd:

$$\mathcal{L}(w, b, e; \alpha) = \mathcal{J}(w, e) - \sum_{t=1}^{N} \alpha_t \{w^T \varphi(x_t) + b + e_t - y_t\},$$

met α_t de Lagrangevermenigvuldigers. Na het opleggen van de condities voor optimaliteit $\frac{\partial\mathcal{L}}{\partial w} = 0, \frac{\partial\mathcal{L}}{\partial b} = 0, \frac{\partial\mathcal{L}}{\partial e_t} = 0, \frac{\partial\mathcal{L}}{\partial \alpha_t} = 0$ en de gepaste substituties leidt dit tot het volgende duale probleem (d.i. het probleem uitgedrukt in de Lagrangevermenigvuldigers):

$$\left[\begin{array}{c|c} 0 & 1_N{}^T \\ \hline 1_N & \Omega + \gamma^{-1} I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right], \tag{0.10}$$

waarbij $y = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^T$, $1_N = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T$, $\alpha = \begin{bmatrix} \alpha_1 & \dots & \alpha_N \end{bmatrix}^T$, $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j), \ \forall i, j = 1, \dots, N$, met K de positief definiete kern. Merk op dat bij het oplossen van het optimalisatieprobleem de kenmerkfunctie φ niet gebruikt werd, en dus niet expliciet dient gedefiniëerd te worden. Enkel het inwendig product, een positief definiete Mercer kern, is nodig. Dit wordt de kerntruc genoemd [127, 150]. Voor de keuze van de kern $K(\cdot,\cdot)$, zie bvb. [127]. Het resulterende kleinste kwadraten steun-vector machine model voor functieschatting kan geëvalueerd worden in een nieuw punt x_* als volgt:

$$\hat{f}(x_*) = \sum_{t=1}^{N} \hat{\alpha}_t K(x_*, x_t) + \hat{b},$$

waarbij $\hat{a}$ an $\hat{b}$ oplossingen zijn van (0.10). Naast functieschatting is het ook mogelijk met behulp van LS-SVMs classificatie uit te voeren, alsook kern PCA (principale component analyse), kern CCA (canonische correlatie analyse), kern PLS (partiële kleinste kwadraten), recurrente netwerken en oplossingen voor niet-lineaire optimale controleproblemen. Voor een overzicht met betrekking tot toepassingen rond het kleinste kwadraten steun-vector machines raamwerk wordt de lezer doorverwezen naar [80, 135–137].

LS-SVMs voor Hammerstein ARX identificatie

Voor het toepassen van het LS-SVM raamwerk op het ARX model worden termen van de vorm $b_j f(u)$ in (0.8) vervangen worden door functies $w_j^T \varphi(u)$ waarbij $\varphi(u)$ de kenmerkfunctie is. Het schatten van de termen b_j en f wordt zo vervangen door het schatten van vectoren w_j in een hoogdimensionele ruimte. Merk op dat deze stap als een overparameterisatiestap kan beschouwd worden. Uit

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} w_j^T \varphi(u_{t-j}) + d + e_t,$$

volgt het primale LS-SVM probleem

$$\min_{w_j, a, d, e} \mathcal{J}(w_j, e) = \frac{1}{2} \sum_{j=0}^{m} w_j^T w_j + \gamma \frac{1}{2} \sum_{t=r}^{N} e_t^2,$$

met als beperkingen

$$\sum_{j=0}^{m} w_j^T \varphi(u_{t-j}) + \sum_{i=1}^{n} a_i y_{t-i} + d + e_t - y_t = 0, \tag{0.11}$$

$$\sum_{t=1}^{N} w_j^T \varphi(u_t) \quad = \quad 0, \tag{0.12}$$

waarbij noodzakelijke centreringsbeperkingen (0.12) werden toegevoegd (zie ook Hoofdstuk 6). De oplossing van het primale probleem wordt gegeven door het volgende lemma:

Lemma 0.1. *Gegeven het systeem (0.7), worden de kleinste kwadraten steunvector schattingen voor de niet-lineaire functies* $w_j^T\varphi : \mathbb{R} \to \mathbb{R}$, $j = 0, \ldots, m$, *gegeven als:*

$$w_j^T \varphi(u_*) = \sum_{t=r}^{N} \alpha_t K(u_{t-j}, u_*) + \beta_j \sum_{t=1}^{N} K(u_t, u_*),$$

waarbij de parameters $\alpha_t, t = r, \ldots, N$, $\beta_j, j = 0, \ldots, m$, *en de lineaire modelparameters* $a_i, i = 1, \ldots, n$ *en* d *worden bekomen uit de volgende set van lineaire vergelijkingen:*

$$\left[\begin{array}{c|c|c|c} 0 & 0 & 1^T & 0 \\ \hline 0 & 0 & \mathcal{Y}_p & 0 \\ \hline 1 & \mathcal{Y}_p^T & \mathcal{K} + \gamma^{-1} I & K^0 \\ \hline 0 & 0 & {K^0}^T & 1_N^T \Omega 1_N \cdot I_{m+1} \end{array}\right] \left[\begin{array}{c} d \\ \hline a \\ \hline \alpha \\ \hline \beta \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline \mathcal{Y}_f \\ \hline 0 \end{array}\right], \tag{0.13}$$

met $\mathcal{K}$ *en* K^0 *afhankelijk van de kernel* K, *en* $\mathcal{Y}_p$ *een Hankel matrix gevuld met uitgangsmetingen.*

De projectie van het bekomen overgeparemeteriseerde model op de klasse van de Hammerstein systemen gaat als volgt: Schattingen voor de autoregressieve parameters $a_i, i = 1, \ldots, n$ worden onmiddelijk bekomen uit (0.13). Tenslotte hebben we voor een set van ingangen $\begin{bmatrix} u_1 & \ldots & u_N \end{bmatrix}$, dat:

$$\begin{aligned} \begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix} \begin{bmatrix} \underline{\hat{f}}(u_1) \\ \vdots \\ \underline{\hat{f}}(u_N) \end{bmatrix}^T &= \begin{bmatrix} \alpha_N & \ldots & \alpha_r & & & 0 \\ & \alpha_N & \ldots & \alpha_r & & \\ & & \ddots & & \ddots & \\ 0 & & & \alpha_N & \ldots & \alpha_r \end{bmatrix} \\ &\times \begin{bmatrix} \Omega_{N,1} & \Omega_{N,2} & \ldots & \Omega_{N,N} \\ \Omega_{N-1,1} & \Omega_{N-1,2} & \ldots & \Omega_{N-1,N} \\ \vdots & \vdots & & \vdots \\ \Omega_{r-m,1} & \Omega_{r-m,2} & \ldots & \Omega_{r-m,N} \end{bmatrix} + \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_m \end{bmatrix} \sum_{t=1}^{N} \begin{bmatrix} \Omega_{t,1} \\ \vdots \\ \Omega_{t,N} \end{bmatrix}^T, \end{aligned} \tag{0.14}$$

met $\underline{\hat{f}}(u)$ een schatting voor

$$\underline{f}(u) = f(u) - \frac{1}{N} \sum_{t=1}^{N} f(u_t).$$

Zodus kunnen schattingen voor de b_j en de statische niet-lineariteit f bekomen worden uit een rank 1 benadering van de rechterhandzijde van (0.14), bijvoorbeeld door toepassing van een singuliere waarden algoritme. Deze stap correspondeert met de singuliere waarden stap die ook in klassieke overparameterisatie-algoritmen wordt aangetroffen.

Een gelijkaardige afleiding als zonet beschreven kan worden uitgevoerd voor zogenaamde MIMO systemen. Een vergelijking van het kleinste kwadraten steun-vector algoritme met bestaande overparameterisatietechnieken leert dat door de inherente aanwezigheid van een regularisatieraamwerk in kleinste kwadraten steun-vector algoritmes, en het feit dat centreringsbeperkingen op de oplossingen eenvoudig kunnen worden opgelegd, de bekomen modellen typisch beter zijn dan dezen bekomen via reeds langer bestaande overparameterisatietechnieken. Dit gecombineerd met een heldere afleiding van de basisresultaten en de vrijheid die bekomen wordt door de actieve keuze van een geschikte positief definiete kernfunctie maakt van de voorgestelde techniek een prima kandidaat voor Hammerstein model identificatie.

Hoofdstuk 8: Hammerstein N4SID identificatie

Gebasseerd op de resultaten in Hoofdstuk 7, wordt in Hoofdstuk 8 een Hammerstein N4SID algoritme voorgesteld. Het eerder voorgestelde ARX algoritme heeft immers als belangrijk nadeel dat het gebruik van ARX modellen niet toelaat bepaalde types van verstoringen te beschouwen zoals bijvoorbeeld meetruis. Dit laatste is wel mogelijk indien gebruik gemaakt wordt van deelruimte algoritmen zoals het bekende N4SID-algoritme.

Een eerste stap naar de ontwikkeling van een Hammerstein N4SID algoritme is de vervanging van de schuine projectie door een kleinste kwadraten steunvector regressieprobleem. Termen $w_{h,s}$ en de matrices L_y worden daarbij geschat in vergelijkingen van de volgende vorm:

$$Y_f(s,t) = L_y(s,:)Y_p(:,t) + \sum_{h=1}^{2i} w_{h,s}^T \varphi(u_{h+t-2}) + E(s,t),$$

waarbij E een te minimaliseren matrix met residuals is. Het LS-SVM primaire probleem wordt dan geformuleerd als een beperkt optimalisatieprobleem:

$$\min_{w_{h,s},L_y,E,\delta_y} \mathcal{J}(w_{h,s},L_y,E,\delta_y) = \frac{1}{2}\sum_{s=1}^{il}\sum_{h=1}^{2i} w_{h,s}^T w_{h,s} + \frac{\gamma}{2}\sum_{s=1}^{il}\sum_{t=1}^{j} E(s,t)^2,$$

$$\text{s.t.} \begin{cases} Y_f(s,t) + [1_i \otimes \delta_y](s) = L_y(s,:)(Y_p(:,t) + 1_i \otimes \delta_y) & (a) \\ \quad + \sum_{h=1}^{2i} w_{h,s}^T \varphi(u_{h+t-2}) + E(s,t), & \\ \qquad \forall s = 1,\ldots,il,\ t = 1,\ldots,j, & \\ \sum_{t=0}^{N-1} w_{h,s}^T \varphi(u_t) = 0, & (b) \\ \qquad \forall h = 1,\ldots,2i, s = 1,\ldots,li. & \end{cases}$$

Na oplossen van dit primaire probleem kunnen schattingen voor de schuine projectie en vervolgens de interne toestanden van het systeem worden bekomen. In een tweede stap worden de systeemmatrices A, B, C en D, en de statische niet-lineariteit f geschat, opnieuw door het oplossen van een kleinste kwadraten steun-vector regressie probleem van de volgende vorm:

$$\min_{\omega_s, E, \Theta_{AC}} \mathcal{J}(\omega, E) = \frac{1}{2}\sum_{s=1}^{n+l} \omega_s^T \omega_s + \frac{\gamma_{BD}}{2}\sum_{s=1}^{n+l}\sum_{t=1}^{j} E(s,t)^2,$$

$$\text{s.t.} \begin{cases} \mathcal{X}_{i+1}(s,t) = \Theta_{AC}(s,:)\tilde{X}_i(:,t) + \omega_s^T \varphi(u_{i+t-1}), \\ \sum_{t=0}^{N-1} \omega_s^T \varphi(u_t) = 0, \end{cases}$$

met

$$\mathcal{X}_{i+1} = \begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} - \delta_y \end{bmatrix}, \ \Theta_{AC} = \begin{bmatrix} A \\ C \end{bmatrix}, \ \Theta_{BD} = \begin{bmatrix} B \\ D \end{bmatrix}.$$

Er kan eenvoudig experimenteel aangetoond worden dat het uiteindelijk bekomen Hammerstein N4SID algoritme veel beter overweg kan met zaken als meetruis dan het Hammerstein ARX algoritme gepresenteerd in Hoofdstuk 7. Nadeel is uiteraard de grotere complexiteit en het toegenomen aantal parameters in deelruimte-algoritmen.

Hoofdstuk 9: Hammerstein-Wiener identificatie met deelruimte intersectie

De resultaten in Hoofdstukken 7 en 8 bleven beperkt tot Hammerstein systemen. Hoewel kan aangetoond worden dat Wiener identificatie algoritmen kunnen afgeleid worden steunende op gelijkaardige principes als deze gebruikt in Hoofdstukken 7 en 8, mag er gesteld worden dat de identificatie van Hammerstein-Wiener systemen heel wat complexer is. De literatuur rond identificatie van Hammerstein-Wiener systemen is eerder beperkt en het gros van de voorgestelde algoritmen is ofwel essentieel iteratief van aard, of gebaseerd op eerder restrictieve aannames wat betreft de structuur van de ingangen (bvb. witheid).

Opnieuw kijken we naar kleinste kwadraten steun-vector algoritmen voor de ontwikkeling van een Hammerstein-Wiener deelruimte identificatie algoritme. Een belangrijk nieuw element is het gebruik van canonische correlatie analyse, en meer bepaald een niet-lineaire variant ervan, gekend als kern canonische correlatie analyse. Deze laatste steunt op gegeneraliseerde eigenwaardenproblemen van de volgende vorm:

$$K_p K_f \ \mathcal{V}_f = K_p K_p \ \mathcal{V}_p \Lambda,$$
$$K_f K_p \ \mathcal{V}_p = K_f K_f \ \mathcal{V}_f \Lambda,$$

waarbij K_p en K_f gepaste kernfuncties zijn. Men kan aantonen dat indien de niet-lineariteit aan de uitgang van het Hammerstein-Wiener systeem inverteerbaar is, een interne toestand van het bestudeerde systeem kan bekomen worden via een kern canonische correlatie analyse -stap. De schatting van de systeemmatrices A en B en de statische niet-lineariteit f volgt daarna ongeveer hetzelfde verloop als in Hoofdstuk 8 en steunt volledig op het volgende regressieprobleem:

$$\begin{aligned} \min_{w,E,A} \mathcal{J}(w,E) &= \tfrac{1}{2}\textstyle\sum_{s=1}^{n} w_{f,s}^T w_{f,s} + \frac{\gamma_u}{2}\sum_{s=1}^{n}\sum_{t=1}^{j-1} E(s,t)^2, \\ \text{s.t.} \qquad & \widehat{X}_{i+1}(s,t) = A\widehat{X}_i(:,t) + w_{f,s}^T \mathcal{U}_{\varphi}(:,t) + E(s,t). \end{aligned}$$

Schattingen voor de matrices C en D en de statische niet-lineariteit g vinden we via:

$$\begin{aligned} \min_{w,E,C,D} \mathcal{J}(w,E) &= \tfrac{1}{2}\textstyle\sum_{s=1}^{l} w_{g,s}^T w_{g,s} + \frac{\gamma_y}{2}\sum_{s=1}^{n}\sum_{t=1}^{j-1} E(s,t)^2, \\ \text{s.t.} \qquad & X_i(1,t) = w_{g,s}^T \mathcal{Y}_{\varphi}(:,t) - C(s,2:n)X_i(2:n,t) \\ & \quad -D(s,:)\mathcal{U}_f(:,t) - E(s,t). \end{aligned}$$

Zoals eerder vermeld heeft het uiteindelijk bekomen algoritme tot groot voordeel met betrekking tot bestaande algoritmen dat geen restrictieve aannames moeten gemaakt worden wat betreft de ingangen van het systeem. Tevens is het voorgestelde algoritme niet iteratief van aard.

Hoofdstuk 10: Besluiten

Algemene besluiten

In dit proefschrift werden technieken bestudeerd voor deelruimte-identificatie van lineaire, Hammerstein en Hammerstein-Wiener systemen. Voor lineaire systemen werd aangetoond dat ondanks de algemeen aanvaarde robuustheid van deelruimte algoritmen, onder specifieke experimentele condities, problemen kunnen optreden met betrekking tot conditionering, of het volledig falen van het algoritme. Verscheidene oplossingen werden voorgesteld en getest in dit proefschrift. Nieuwe voorgestelde methodes voor het oplossen van het zogenaamde reële positiviteit probleem bleken beter te presteren dan bestaande oplossingen. Het toevoegen van een regularisatieterm aan de schuine projectie in ingangs/uitgangs-deelruimte-algoritmen bleek dan weer een positief effect te hebben op de conditionering van deze laatsten.

Voor Hammerstein en Hammerstein-Wiener systemen werden betrouwbare deelruimte identificatie algoritmen ontwikkeld door het combineren van ideeën ontrent kleinste kwadraten kern-vector machines met de belangrijkste projecties die in deelruimte algoritmen aanwezig zijn. Ook hier werd aangetoond dat de nieuwe voorgestelde algoritmen enkele belangrijke voordelen hebben ten opzichte van bestaande technieken. Dit onder andere door het mechanisme van regularisatie dat inherent aanwezig is in LS-SVMs, en het feit dat extra beperkingen op de oplossingen van een kleinste kwadraten steun-vector regressie eenvoudig kunnen worden opgelegd.

Toekomstig onderzoek

De resultaten omtrent het gebruik van regularisatie als remedie voor slechte conditionering in gecombineerd stochastisch-deterministische deelruimte identificatie kennen en zouden verder onderzocht moeten worden. Een niet exhaustieve lijst van mogelijkheden ziet eruit als volgt:

1. Bestudeer het effect van regularisatie in de schuine projectie op de bekomen toestand. Blijven de basis-eigenschappen omtrent deelruimte-identificatie zoals het zogenaamde unificatietheorema behouden? Leidt het gebruik van regularisatie tot een verandering van de basis waarin de toestand wordt uitgedrukt?

2. Gebruik regularisatie in de schuine projectie maar tracht het gebruik van een gescheiden parameterisatie voor het deterministische en het stochastische deelsysteem te vermijden. Is het mogelijk een tweede regularisatiestap te gebruiken ter vervanging van de gescheiden parameterisatie?

Wat niet-lineaire deelruimte technieken betreft mag het duidelijk zijn dat niet alle mogelijkheden zijn uitgeput. Drie duidelijke mogelijkheden voor toekomstig onderzoek tekenen zich af:

1. In Hoofdstuk 7 hebben we gezien dat het gebruik van centrerings-beperkingen noodzakelijk is teneinde een goede schatting voor o.a. Hammerstein ARX systeem te bekomen. Beter zou echter zijn om dadelijk collineariteitsbeperkingen op te leggen op de verscheidene vectoren w_j die figureren in het algoritme.

2. In [151] werden enkele preliminaire resultaten gepresenteerd waarin de ideeën rond LS-SVM Hammerstein-Wiener identificatie worden uitgebreid naar algemeen niet-lineaire systemen. Dit is een beloftevol onderzoeksgebied aangezien de onderzochte technieken in principe toelaten deelruimte-identificatie toe te passen op nagenoeg eender welk niet-lineair systeem. Langs de andere kant zal het gebrek aan structuur in de bestudeerde modellen leiden tot een explosie in het aantal parameters met een grote onzekerheid op de bekomen modellen tot gevolg. Het blijft dus af te wachten of dergelijke deelruimte algoritmen voor algemeen niet-lineaire systemen nuttig zijn in de praktijk.

3. In plaats van het uitbreiden van enkele voorgestelde resultaten naar algemeen niet-lineaire systemen is het wellicht interessant te onderzoeken of de algoritmen bestudeerd in dit proefschrift kunnen worden uitgebreid naar andere gestructureerde niet-lineaire modelklassen zoals de Wiener-Hammerstein klasse, gekarakteriseerd door een Wiener model gevolgd door een Hammerstein model.

Chapter 1

Introduction

In this introduction, we will briefly discuss the importance of subspace identification algorithms in the system identification context. It will be argued that subspace identification algorithms offer many advantages over classical algorithms when presented with a system identification task. Nevertheless, we will also see that under certain experimental conditions, subspace identification algorithms may break down, or produce unreliable results. Another drawback of subspace identification algorithms will be found in the fact that they are largely limited to linear systems. The focus of this thesis, namely the study of the reliability of linear subspace identification algorithms, and an extension of the subspace framework to Hammerstein- and Hammerstein-Wiener systems follows naturally from these observations.

1.1 Subspace identification

System identification in its broadest sense is a powerful technique for building accurate mathematical models of complex systems from noisy data. It distinguishes itself from mathematical modeling approaches based on the combination of a set of scientific laws, in that no detailed knowledge of the inner-workings of the system is needed. Because of this, system identification algorithms often offer a cheap alternative over more complex modeling approaches based on first principles.

Many of the system identification algorithms in use today can be classified as so called “predictor error”-methods. Typically, a certain model structure is assumed and a set of free parameters is estimated by optimizing the predictive performance of the corresponding models on measured data-sequences. A well known drawback of these approaches, is that the resulting optimization problem is in general non-convex, and this even for the relatively limited class of linear systems. Consequently, many “predictor error”-methods are not guaranteed to deliver an optimal solution due to the presence of local minima in the cost-function. Furthermore, the inherently iterative nature of the employed optimization algorithms can lead to problems related to lack of convergence, slow convergence or numerical instability.

For linear systems, subspace identification algorithms offer an alternative to the classical “predictor error”-methods. Subspace identification algorithms were mainly developed in the last decade of the former century and are entirely based on numerically robust linear geometrical operations such as projections and the singular value decomposition. As such, no convergence problems or numerical instabilities will occur. Furthermore, in contrast to “predictor error” approaches which require a certain user specified parameterization, subspace identification algorithms use full state space models and the only parameter is the order of the system. As a result subspace algorithms for the identification of linear systems have strongly gained in popularity over the last two decades and are currently used in a vast range of applications such as structural identification and fault detection [15, 16].

However, despite the current popularity and the aforementioned robustness of subspace identification algorithms, evidence has emerged over the last few years that in some specific cases, subspace algorithms may fail, or yield unreliable results. Another drawback of subspace algorithms is that they are largely constrained to the class of linear systems. Both issues will briefly be discussed in the following sections.

1.2 Positive realness

Subspace identification algorithms exist for input/output as well as output-only system. Especially in the output-only case, the aim of identification, including subspace identification, is to obtain not only a linear model for the observed dynamics, but also an estimate for the statistics of the driving noise sources. Although the former is not a problem when using output-only subspace identification, it was shown [33] that the latter can fail if certain conditions are not met by one of the intermediate results in the algorithm. Namely when the so-called covariance model is not positive real. In case of a failure, the covariance model is said to suffer from a lack of positive realness.

Lack of positive realness is a relevant problem in practical applications. In this thesis we will show that it occurs, even for some seemingly trivial tasks such as the modeling of an ambiently excited vibrating structure. Because of its practical relevance, the positive realness problem has received considerable attention over the last few years. Besides discussing some already existing solutions, in this thesis we will introduce a new algorithm to impose positive realness using the concept of Tikhonov regularization.

In its broadest sense, regularization denotes the act of slightly altering a given optimization problem such that the uncertainty on the obtained solution (e.g. the variance on a set of obtained model parameters) is significantly reduced. Although altering the optimization problem in general leads to the introduction of a bias, the total expected error is often seen to decrease as a result of the decrease in variance. This concept is known as the bias/variance trade-off. Another advantage of regularization is that certain conditions can be imposed on the solution of an optimization problem. It is this property that will turn out to be particularly useful for the positive realness problem.

By using a special form of regularization, known as Tikhonov regularization, it will be shown that positive realness can be imposed on the covariance model in output-only subspace identification. Furthermore, the obtained model and the statistics of the driving noises will be seen to be better than what can be obtained using already existing solutions. A graphical description of the positive realness problem is given in Figure 1.1.

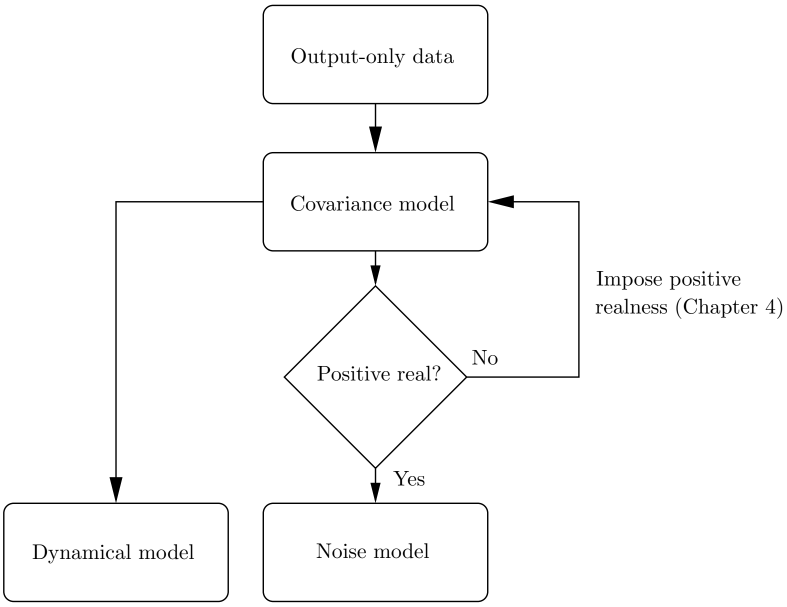


Figure 1.1: The covariance model is obtained as an intermediate step in output only subspace identification. In order to be able to extract statistics for the noise sources acting on the system, it needs to satisfy the positive realness assumption. Positive realness can be imposed using results presented in Chapter 4 of this thesis.

1.3 Ill-conditioning in subspace identification

Despite the fact that they are based on numerically robust geometrical operations such as projections and the singular value decomposition, subspace identification algorithms for input-output systems, and especially the well known N4SID algorithm [144], are ill-conditioned under certain experimental conditions involving highly colored inputs [22, 25]. Two reasons for this phenomenon will be discussed in this thesis. The first one only applies to the N4SID identification algorithm, the second one also applies to most other subspace identification algorithms such as the PO-MOESP [155] and the CVA [94].

- *Ill-conditioned oblique projection:* The N4SID subspace identification algorithm is dominated by an oblique projection, which enables the estimation of an internal state based on input-output measurements. It will be shown that this oblique projection is ill-conditioned for highly colored inputs, leading to an unreliable state and model.
- *Correlation between the stochastic system state and the input:* Most subspace identification algorithms yield an internal state which contains contributions due to the system inputs (the deterministic contributions) and contributions due to the disturbances acting on the system (the stochastic contributions). Although theoretically the correlation between the stochastic part of the state and the system inputs is zero, when working with a finite amount of measurement data, this is not automatically the case. It will be shown that in the presence of highly colored inputs, even a weak correlation between stochastic components of the state and the system inputs can lead to a serious deterioration of the obtained results.

A proposed algorithm to deal with both problems, the so-called orthogonal decomposition method, was presented in [26]. In contrast to most existing subspace identification algorithms, the orthogonal decomposition method features a decomposition of the measured data in a stochastic and a deterministic part to deal with the problem of weak correlations between the stochastic state and the system inputs. The ill-conditioning of the oblique projection is avoided by replacing the oblique projection by an orthogonal projection which is commonly found in MOESP type of algorithms.

In this thesis we will study an alternative to this approach, still involving an orthogonal decomposition of the measured data in a stochastic and a deterministic part, but maintaining the oblique projection as the key to obtaining the state. It will be seen that the problem of ill-conditioning of the oblique projection can be dealt with by applying regularization to the oblique projection. The resulting algorithm will be seen to perform better than the orthogonal decomposition method and, together with results obtained for the positive realness problem, serves to highlight the opportunities that emerge when using regularization in a subspace identification context.

1.4 Hammerstein and Hammerstein-Wiener identification

As mentioned earlier, a drawback of the subspace identification framework is that its practical use is largely limited to linear systems. Nevertheless, an extension to some classes of non-linear systems is possible. In [51] for instance, a subspace identification algorithm for the identification of bilinear systems was introduced. Another interesting development is the introduction of subspace identification algorithms for Hammerstein, Wiener and Hammerstein-Wiener systems [75,156,159]. Hammerstein-, Wiener- and Hammerstein-Wiener systems are composed of a linear dynamical part, preceded and/or followed by a static non-linearity such as shown in Figure 1.2. Their dynamical part being linear, these systems are very attractive targets for the extension of linear system identification algorithms to non-linear systems. Meanwhile, the presence of static non-linearities allows to describe a much wider range of dynamics than what can be described by purely linear models. Unfortunately, most subspace identification algorithms for use with Hammerstein, Wiener or Hammerstein-Wiener models impose rather restrictive assumptions on the inputs of the system (such as whiteness), or are iterative in nature.

An alternative is found in so-called overparameterization approaches which are non-iterative, do not impose restrictive assumptions on the inputs, and lead to trivially solvable convex-optimization problems. However, overparameterization approaches will be seen to suffer from an explosion in the number of parameters with large uncertainties on the resulting model as a consequence.

In this thesis, we will introduce a new framework for the identification of Hammerstein- and Hammerstein-Wiener systems based on methods of Least Squares Support Vector Machines (LS-SVMs) [135]. Most results will be introduced in a Hammerstein ARX setting and later be extended to subspace identification in a Hammerstein- and a Hammerstein-Wiener setting. It will be seen that the newly introduced algorithms are to some extent related to the overparameterization approach but avoid the explosion in the number of parameters due to the availability of a strong regularization framework in the LS-SVM formalism. As such, the algorithms that are introduced in this thesis will in general outperform existing overparameterization algorithms while keeping their main advantages such as convexity and the fact that no restrictive assumptions are imposed on the inputs. As an additional note, we mention that although Wiener-model identification is not explicitly treated in this thesis, most results for Hammerstein-model identification can easily be applied to Wiener systems with an invertible output non-linearity.

1.5 Contributions

This thesis is composed of two parts. Part I will deal with subspace identification in a linear framework, and largely revolve around the issues of positive-realness in output-only subspace identification and possible ill-conditioning in input-output subspace identification. The main contributions of this part are summarized as follows:

- Imposing positive realness on a covariance model by using Tikhonov regularization [65, 66].

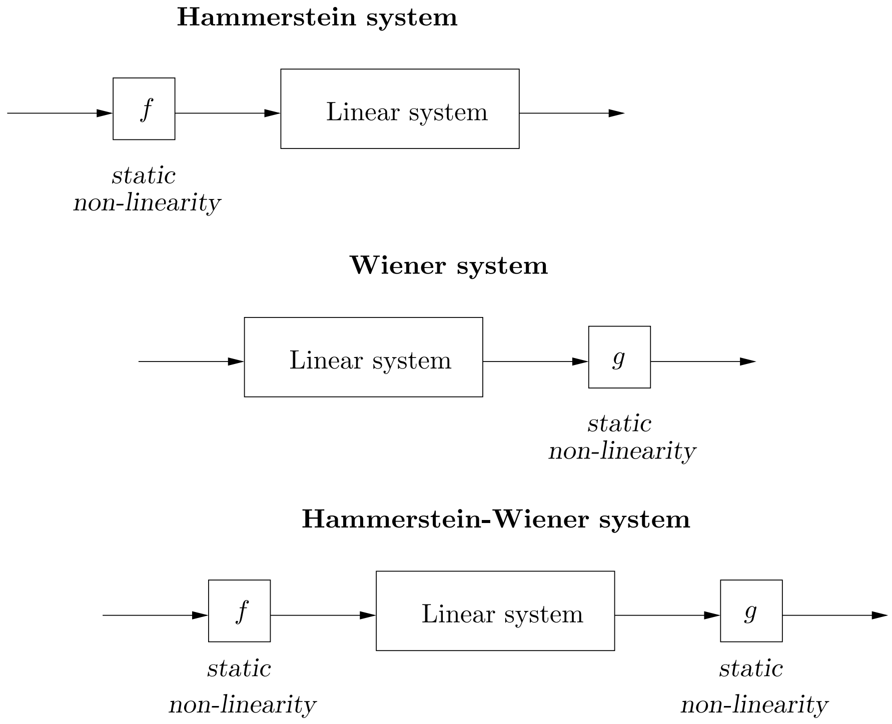


Figure 1.2: Hammerstein systems (top), Wiener systems (middle) and Hammerstein-Wiener systems (bottom) are composed of a linear dynamical model preceded and/or followed by static non-linearities

- Showing that regularization can play an important role in dealing with ill-conditioning in input-output subspace identification.

Part II will be concerned with an extension of subspace identification algorithms to the class of Hammerstein-systems and the class of Hammerstein-Wiener systems. This extension will be performed by means of the LS-SVM formalism, first in a relatively intuitive ARX setting, thereafter applied to various existing subspace identification algorithms. The contributions of this part are summarized as follows:

- Introducing an algorithm for the identification of Hammerstein models using LS-SVMs in an ARX setting [62, 64].
- Introducing an extension of the N4SID subspace identification algorithm to the class of Hammerstein systems [63].
- Introducing an extension of the subspace intersection algorithm to the class of Hammerstein-Wiener systems [60].

Besides the results listed above it is useful to note that in order to keep the discussion in this thesis concise and focused, several results obtained during the doctoral research work were omitted from the text. A list of the most relevant items is found here:

- The derivation of a recursive version of the stochastic realization algorithm by using subspace tracking algorithms [61].
- Automatic separation of meaningful resonances from noise-induced phenomena in modal analysis [58, 59, 61, 67, 129].
- A linear sensitivity analysis of N4SID subspace identification algorithms [57].
- An improved condition number for the total least squares problem [56].

The interested reader is kindly referred to the relevant references.

1.6 Chapter-by-chapter overview

In this section, we provide an overview of the different chapters in this thesis and the relations between them. A graphical outline of the relations between the chapters is also given in the overview Figure 1.3.

Chapter 2: Linear geometrical tools

The reader is assumed to be familiar with the basic linear geometrical tools such as the singular value decomposition and linear least-squares. However, since Tikhonov regularization, as a solution to ill-conditioned least-squares problems will play a central role in this thesis, it was judged useful to commence this chapter with a brief review of the main properties of linear least squares and the matrix condition number. Other linear geometrical tools that will be discussed in this chapter are the orthogonal projection, the oblique projection and canonical correlation analysis, all three key components of subspace identification algorithms.

Chapter 3: Subspace identification

In Chapter 3, and starting from a conceptual overview of the theory of deterministic and stochastic realization, the key ideas behind subspace identification algorithms will be introduced. We will show that the state of a linear system can directly be obtained from projections of structured data-matrices containing input- and output-measurements on the system. Most of the discussion in this chapter will center around so-called combined stochastic-deterministic models where the state contains information from the stochastic as well as the deterministic part of the system. Nevertheless, at the end of the chapter subspace identification algorithms based on a separate parameterization of the stochastic and the deterministic part of the system will be introduced. These algorithms will play a major role in the discussion on ill conditioned subspace algorithms in Chapter 5.

Chapter 4: The positive realness problem

We will start the discussion on the positive realness problem in output-only subspace identification with a description of the problem and the parameters that influence its occurrence. After the introduction of a number of existing approaches to impose positive realness on a covariance model we will propose a new approach based on Tikhonov regularization. It will be proven that using this new approach, positive realness can be guaranteed provided the amount of regularization is appropriately chosen. Using tests on simulation datasets our new approach will be seen to outperform existing approaches. Furthermore, tests will be performed on a real-life application in the field of modal-analysis to demonstrate the practical relevance of the approach.

Chapter 5: Ill-conditioning in subspace identification

At the beginning of this chapter we will argue that subspace identification algorithms can perform very badly under certain experimental conditions, and especially if the inputs are highly colored. Two main causes for this problem will be discussed in detail and illustrated with an appropriate example. Thereafter, it will be shown that the conditioning problems discussed in this chapter are also found in more classical settings such as ARX model identification.

After discussing the similarities and differences of the conditioning problems in subspace and ARX, the orthogonal decomposition method proposed in [26] will be introduced as a possible solution. The key components of the orthogonal decomposition method, the use of a separately parameterized model structure and an orthogonal projection to obtain the state, instead of the oblique projection which is commonly found in N4SID algorithms, will be discussed.

In an attempt to avoid having to replace the oblique projection by an orthogonal projection, we will thereafter introduce a regularized version of the N4SID which deals with the ill-conditioning in the oblique projection by employing regularization. It will however be argued that the use of a separate parameterization remains necessary in many cases. At the end of the chapter the resulting regularized N4SID algorithm will be seen to outperform the orthogonal decomposition method on a set of examples.

Chapter 6: Hammerstein, Wiener and Hammerstein-Wiener systems

In this chapter, the state of the art in Hammerstein, Wiener and Hammerstein-Wiener identification will briefly be reviewed. The overparameterization approach which will prove to be essential in the following chapters will be introduced, and its main weaknesses discussed.

Chapter 7: Hammerstein NARX identification

After an introduction of the method of LS-SVM function approximation, an ARX Hammerstein identification based on LS-SVMs will be introduced. Several advantages of the LS-SVM based algorithm such as an inherently available regularization framework, and the possibility to impose extra constraints on the estimated non-linearities will be discussed, and this for SISO as well as MIMO models. Finally, the newly proposed algorithm will be compared to existing overparameterization approaches to highlight its advantages.

Chapter 8: Hammerstein N4SID identification

Based on the results in Chapter 7, in Chapter 8 a Hammerstein N4SID subspace identification algorithm will be proposed. We will show that the oblique projection in the N4SID algorithm can be replaced by an LS-SVM regression problem similar to the least squares regression that was found in Chapter 7. Following this strategy, a reliable N4SID subspace identification algorithm for Hammerstein systems is obtained. At the end of the chapter, the newly proposed algorithm is evaluated on a set of examples.

Chapter 9: Hammerstein-Wiener identification using subspace intersection

The results in Chapter 7 and 8 were limited to Hammerstein systems. Although it can be shown that Wiener identification algorithms can be derived along the lines of the derivations in these two chapters, identification of Hammerstein-Wiener systems is a completely different matter altogether. The literature on Hammerstein-Wiener identification is rather sparse, and most proposed algorithms are either restrictive in the kind of inputs that can be used, or are fundamentally iterative in nature.

In Chapter 9, a new Hammerstein-Wiener identification algorithm is proposed which can conveniently be applied without imposing overly restrictive assumptions on the system inputs and maintains convexity in the optimization problem by using results from the theory of kernel canonical correlation analysis. After an introduction into kernel canonical correlation analysis at the beginning of the chapter, the classical subspace intersection algorithm is extended to a Hammerstein-Wiener setting and tested on a number of examples.

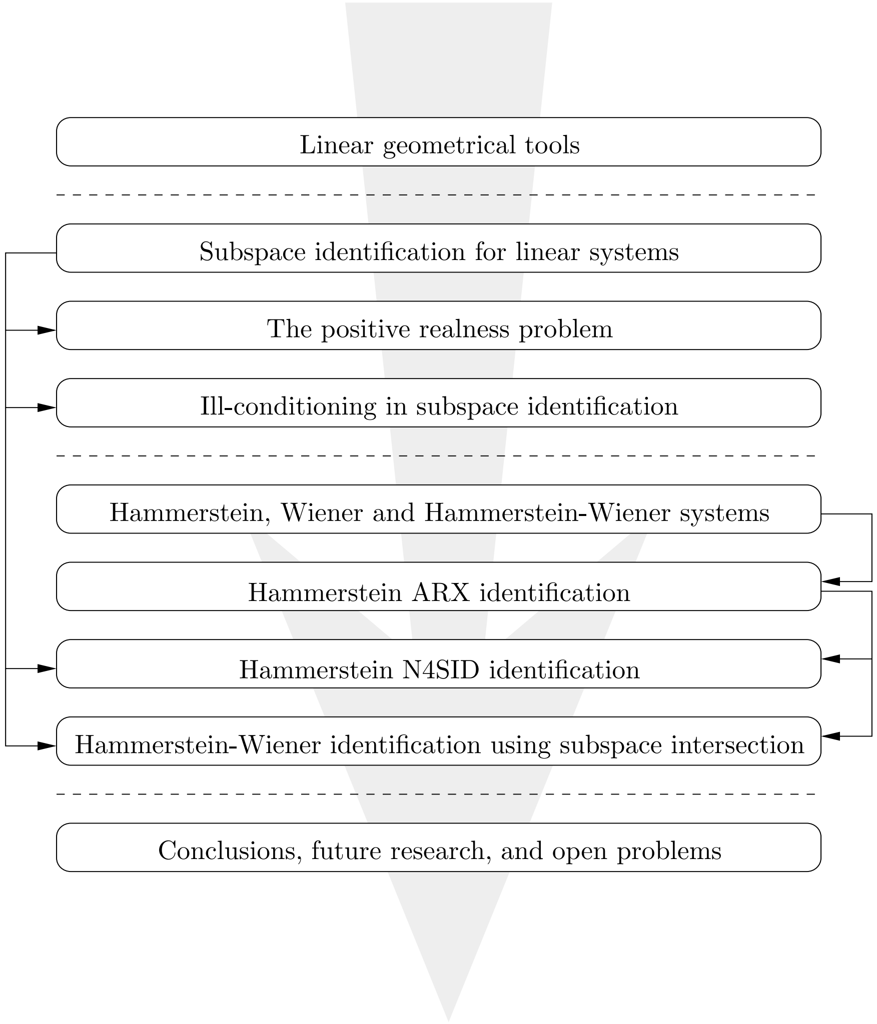

Linear geometrical tools
Subspace identification for linear systems
The positive realness problem
Ill-conditioning in subspace identification
Hammerstein, Wiener and Hammerstein-Wiener systems
Hammerstein ARX identification
Hammerstein N4SID identification
Hammerstein-Wiener identification using subspace intersection
Conclusions, future research, and open problems

Figure 1.3: Graphical overview of the relations between the different chapters. The first part of this thesis is largely concerned with linear subspace identification. In the second part an extension of linear subspace identification to the identification of Hammerstein and Hammerstein-Wiener systems is proposed. The thesis starts with a short description of some commonly used linear geometrical tools. Conclusions, openings for further research and a list of open problems are provided at the end of the thesis. Arrows in the figure indicate that the results of a chapter are heavily used in other chapters of the thesis.

Chapter 2

Linear geometrical tools

Linear geometrical tools play an important role in this thesis. A brief introduction to some key concepts related to least-squares regression, various types of projections and the theory of Canonical Correlation Analysis is therefore given in this chapter. For a more elaborate overview of the topics discussed in this chapter, we refer the reader to the extensive literature on the subject such as for instance found in [5, 34, 82, 119].

2.1 Linear least-squares

Given $A \in \mathbb{R}^{N \times n}$ with $N \geq n$ and $b \in \mathbb{R}^N$, the aim of linear least-squares is to find an estimate $x_{\mathrm{LS}} \in \mathbb{R}^n$ such that:

$$(x_{\mathrm{LS}}) = \arg\min_x \|Ax - b\|_2. \tag{2.1}$$

It can be shown that a solution to the problem (2.1) can always be found. The solution is unique if and only if A has full column rank and is given by [34]

$$(x_{\mathrm{LS}}) = A^\dagger b, \tag{2.2}$$

with $A^\dagger$ the Moore-Penrose pseudo-inverse of A [119]. If $A^T A$ is invertible, the pseudo-inverse is given as $A^\dagger = (A^T A)^{-1} A^T$. Alternatively, for matrices $A \in \mathbb{R}^{N \times n}$ with $N \leq n$, and if AA^T is invertible, the pseudo-inverse of A is given as $A^\dagger = A^T (AA^T)^{-1}$.

2.1.1 Geometric interpretation of the least-squares problem

Let $A \in \mathbb{R}^{N \times n}$ with $N \geq n$ and $\mathrm{rank}(A) = n$. Then A is a linear mapping of $\mathbb{R}^n \to \mathbb{R}^N$. We know that every vector $u \in \mathrm{Col}(A)$ can be written as $u = Ax$ for some $x \in \mathbb{R}^n$. Let $b \in \mathbb{R}^N$. Then $\|b - Ax_{\mathrm{LS}}\|_2$ is the Euclidean distance between the endpoints of b and Ax_{LS}. It is clear that this distance is minimal if and only if $b - Ax_{\mathrm{LS}}$ is perpendicular to $\mathrm{Col}(A)$. Hence, the solution to the least-squares problem (2.1) can easily be understood as the vector x_{LS} such that Ax_{LS} is the orthogonal projection of b onto $\mathrm{Col}(A)$ (see also Figure 2.1).

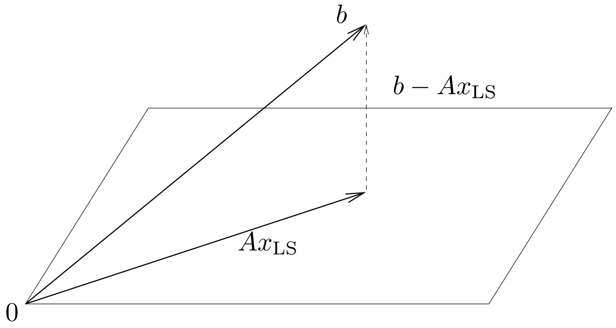


Figure 2.1: The solution x_{LS} of the least-squares problem $Ax_{\mathrm{LS}} \simeq b$ is such that Ax_{LS} is the orthogonal projection of b onto the column-space of A.

2.1.2 The matrix condition number

It is instructive to study the influence of a small perturbation δb on b on the solution x_{LS} of the least-squares problem (2.1). The condition number of a matrix A is defined as its largest singular value divided by its smallest singular value:

$$\mathrm{Cond}(A) = \frac{\sigma_1(A)}{\sigma_n(A)} = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)},$$

With $b = b_R + b_N$ where b_R is the part of b which lies in $\mathrm{Col}(A)$ and b_N is the part which lies in its orthogonal complement denoted as $\mathrm{Col}(A)^{\perp} = \mathcal{N}(A^{\dagger})$, it can be shown that for every δb [34]

$$\frac{\|\delta x_{\mathrm{LS}}\|_2}{\|x_{\mathrm{LS}}\|_2} \leq \mathrm{Cond}(A) \frac{\|\delta b\|_2}{\|b_R\|_2}, \tag{2.3}$$

with $x_{\mathrm{LS}} = A^{\dagger} b$ and $\delta x_{\mathrm{LS}} = A^{\dagger} \delta b$.

Hence, the condition number of a matrix plays an important role in linear algebra. A large condition number is usually the result of a near collinearity in the columns of A. As long as A has full column rank, the solution to the least-squares problem is known to be unique. However, with growing dependency between the columns of A, multiple different solutions x_{LS} will lead to good approximations Ax_{LS} for the projection of b onto $\mathrm{Col}(A)$. In this case the problem (2.1) is said to be ill-conditioned and the resulting estimates for x_{LS} are unreliable. As such, the condition number of the matrix A is a measure for the conditioning of the problem (2.1). Note that this conditioning measure does not depend on the the vector b. This due to the fact that b_R in (2.3) is constrained to $\mathrm{Col}(A)^{\perp} = \mathcal{N}(A^{\dagger})$. The following relations can be shown to hold for the matrix condition number of a matrix $A \in \mathbb{R}^{N\times n}$ with $\mathrm{rank}(A) = n$:

$$\begin{aligned} \mathrm{Cond}(A^T A) &= \mathrm{Cond}(A)^2, & \tfrac{\sigma_1(AA^T)}{\sigma_n(AA^T)} &= \mathrm{Cond}(A)^2, \\ \mathrm{Cond}(A^T) &= \mathrm{Cond}(A), & \mathrm{Cond}(A^{\dagger}) &= \mathrm{Cond}(A). \end{aligned}$$

2.1.3 Variance on the estimated parameters

Following the discussion in 2.1.2 and assuming that b is perturbed by an amount δb with statistical properties $E\{\delta b\} = 0$ and $E\left\{\delta b(\delta b)^T\right\} = \sigma_b^2 I_N$, the covariance matrix of the resulting perturbation on x_{LS} is easily calculated as:

$$\begin{aligned} E\left\{\delta x_{\mathrm{LS}}(\delta x_{\mathrm{LS}})^T\right\} &= \sigma_b^2 A^{\dagger} {A^{\dagger}}^T \\ &= \sigma_b^2 (A^T A)^{-1}(A^T A)(A^T A)^{-1} = \sigma_b^2 (A^T A)^{-1} = \sigma_b^2 H^{-1}. \end{aligned} \tag{2.4}$$

In the equation above, the matrix $H = A^T A$ is the Hessian of the least-squares problem (2.1). In general, the Hessian is defined as the square matrix of second order partial derivatives of a scalar-valued function. The Hessian is important in optimization theory since it offers a good insight into the cost-function's shape at a particular point. This is especially useful close to a local minimum, where lines of equal cost roughly describe ellipsoids in the solution space, with their main axes found as the eigenvectors of the Hessian, and the length of those axes equal to the inverse of the respective eigenvalues. The condition number of the Hessian is therefore a particularly useful measure to assess the sensitivity of a local optimum. Large condition numbers point to heavily stretched ellipsoids, meaning that in some directions the obtained solutions are far better defined that in others. A condition number close to 1 signifies an almost spherical symmetry in the local cost. For the least-squares problem, the relation between the sensitivity of the solution and the condition number of the Hessian is also apparent from relation (2.4). We have:

$$\mathrm{Cond}\left(E\left\{\delta x_{\mathrm{LS}}(\delta x_{\mathrm{LS}}^T)\right\}\right) = \mathrm{Cond}(H) = \mathrm{Cond}(A)^2, \tag{2.5}$$

2.1.4 Extension of the least-squares condition number to general linear maps

The reasoning in 2.1.2 for the least-squares problem can be extended to obtain a condition number for any linear operation $L : \mathbb{R}^N \to \mathbb{R}^M$ with $M \leq N$ and $\mathrm{rank}(L) = n$. Replacing $A^{\dagger}$ in 2.1.2 by L, x_{LS} by x, and taking $b, \delta b \in \mathbb{R}^N$ and $b = b_R + b_N$ with $b_N \in \mathcal{N}(L)$ and $b_R \in \mathcal{N}(L)^{\perp}$, we have $\delta x = L\delta b$ and $x = Lb = Lb_R$ so that

$$\begin{aligned} \|\delta x\|_2 &\leq \sigma_1(L)\|\delta b\|_2, \\ \|x\|_2 &\geq \sigma_n(L)\|b_R\|_2. \end{aligned}$$

Hence

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\sigma_1(L)}{\sigma_n(L)} \frac{\|\delta b\|}{\|b_R\|}. \tag{2.6}$$

For $E\left\{\delta b(\delta b)^T\right\} = \sigma_b^2 I_N$ we have $E\left\{\delta x(\delta x)^T\right\} = \sigma_b^2 L L^T$, so that we can introduce the condition number of the linear operator as:

$$\mathrm{Cond}_L(L) = \frac{\sigma_1(L)}{\sigma_n(L)} = \sqrt{\frac{\sigma_1(LL^T)}{\sigma_n(LL^T)}} = \sqrt{\frac{\sigma_1(E\left\{\delta b(\delta b)^T\right\})}{\sigma_n(E\left\{\delta b(\delta b)^T\right\})}}, \tag{2.7}$$

which will be used as a basis for the condition number of the orthogonal and the oblique projection in the following sections. Note again that the vector b does not appear in (2.7) due to the fact that b_R in (2.6) is an element of $\mathcal{N}(L)^\perp$.

2.2 The orthogonal projection

The orthogonal projection of a vector b onto the space spanned by the columns of A (with A of full column rank) is defined as the vector y in $\mathrm{Col}(A)$ which is closest to b in Euclidean norm. From Subsection 2.1.1, we know that y satisfies $y = Ax$ with x the solution to the least-squares problem (2.1). Hence, we have

$$y = AA^\dagger b = A(A^T A)^{-1} A^T b = \mathcal{P}_A b, \tag{2.8}$$

with an implicit definition for the linear projection operator $\mathcal{P}_A$.

2.2.1 Condition number of the orthogonal projection

The condition number of the linear operator $\mathcal{P}_A$ and by extension of the orthogonal projection is given by:

$$\mathrm{Cond}_L\left(\mathcal{P}_A\right) = 1, \tag{2.9}$$

which can easily be verified by plugging in the singular value decomposition $A = USV^T$ in $\mathcal{P}_A$. Hence, the orthogonal projection is in general considered to be perfectly conditioned.

2.2.2 Orthogonal projection onto the row-space of a matrix

In the derivation of subspace identification algorithms in Chapter 3, we will mostly work with row-spaces of matrices instead of column-spaces. The orthogonal projection of the row-space of B onto the row-space of A can easily be derived from the results. Its row-space is found as follows:

$$\boxed{B/A = B\mathcal{P}_{A^T} = BA^\dagger A.}$$

2.3 The oblique projection

2.3.1 Oblique projection onto the column-space of a matrix

The oblique projection of the column-space of a matrix $C \in \mathbb{R}^{N \times n_C}$ onto the column-space of a matrix $B \in \mathbb{R}^{N \times n_B}$ along the column-space of a matrix $A \in \mathbb{R}^{N \times n_A}$ is given by $C = B\widehat{X}_B$ whereby $\widehat{X}_B$ is obtained from the following least-squares problem:

$$(\widehat{X}_A, \widehat{X}_B) = \arg \min_{X_A, X_B} \left\| C = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} X_A \\ X_B \end{bmatrix} \right\|_F,$$

and calculated as:

$$B\widehat{X}_B = \mathcal{P}_{\{B|A\}} C = \begin{bmatrix} 0_{N \times n_A} & B \end{bmatrix} \begin{bmatrix} A & B \end{bmatrix}^{\dagger} C.$$

This projection is graphically depicted in Figure 2.2.

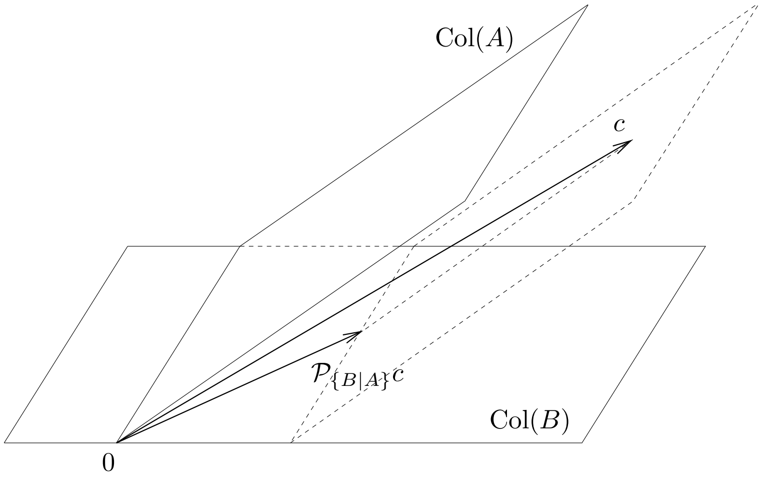


Figure 2.2: The oblique projection of a vector c onto the column space of B along the column space of A.

2.3.2 Oblique projection onto the row-space of a matrix

When working with row-spaces, the oblique projection of C onto Row(B) along Row(A) can easily be derived along the lines of the results above. With $\widehat{X}_B$ obtained from the least-squares problem:

$$(\widehat{X}_A, \widehat{X}_B) = \arg \min_{X_A, X_B} \left\| C - \begin{bmatrix} X_A & X_B \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right\|_F,$$

the projection is given as follows:

$$\boxed{C/_{A} B = C\mathcal{P}^{T}_{\{B^T|A^T\}} = \widehat{X}_B B = C \begin{bmatrix} A \\ B \end{bmatrix}^{\dagger} \begin{bmatrix} 0 \\ B \end{bmatrix}.}$$

Furthermore, it can be shown that

$$C/_{A} B = [C/A^{\perp}] \cdot [B/A^{\perp}]^{\dagger} \cdot B.$$

A condition number for the oblique projection will be derived in Section 2.5. This derivation will be based on results from canonical correlation analysis which will be presented in Section 2.4.

2.4 Canonical correlation analysis

In this section, we will briefly introduce the theory of canonical correlation analysis or CCA. Although CCA was originally developed in a statistical setting [82] we will mainly focus on its geometrical interpretation as the search for principal angles and directions between two subspaces. The concept of principal angles and directions will be introduced in Subsection 2.4.1. A brief discussion on the statistical interpretation itself will be given in Subsection 2.4.2.

2.4.1 Principal angles between subspaces

It is well known that the angle $a \sphericalangle b$ between two vectors $a, b \in \mathbb{R}^N$ can be obtained from:

$$\cos[a \sphericalangle b] = \frac{|a^T b|}{\|a\|_2 \|b\|_2}.$$

This notion of an angle can be generalized to angles between subspaces. Suppose $S_1 \in \mathbb{R}^{d_1 \times N}$, $d_1 \leq N$ and $S_2 \in \mathbb{R}^{d_2 \times N}$, $d_2 \leq N$ span two row-spaces in $\mathbb{R}^N$ such that $\mathrm{rank}(S_1) = r_1$ and $\mathrm{rank}(S_2) = r_2$. A natural extension of the one-dimensional angle is to choose a unit vector $v_1 \in \mathbb{R}^N$ from $\mathrm{Row}(S_1)$ and a unit vector $u_1 \in \mathbb{R}^N$ from $\mathrm{Row}(S_2)$ such that the angle between v_1 and u_1 is minimized. The vectors v_1 and u_1 are called the first principal directions and the angle between them is the first principal angle $0 \leq \theta_1 \leq \pi/2$. The second principal angle and direction can be found by choosing v_2 and u_2 perpendicular to v_1 and u_1, again such that the angle between them is minimized. This procedure is continued until $r = \min(r_1, r_2)$ angles and corresponding principal vectors have been found. The procedure is graphically depicted in Figure 2.3.

Theoretically the search for principal angles and directions can be summarized as the search for two matrices $U \in \mathbb{R}^{r \times N}$ and $V \in \mathbb{R}^{r \times N}$ such that $\mathrm{Row}(U) \subset \mathrm{Row}(S_1)$ and $\mathrm{Row}(V) \subset \mathrm{Row}(S_2)$ and

$$UU^T = I_r, \;\; VV^T = I_r, \;\; UV^T = \Lambda, \tag{2.10}$$

with $\Lambda \in \mathbb{R}^{r \times r}$ a diagonal matrix containing the cosines of the principal angles. From (2.10) we obtain:

$$UV^T \;\; = \;\; UU^T \Lambda,$$

$$VU^T = VV^T\Lambda.$$

Choosing $A \in \mathbb{R}^{r\times d_1}$ and $B \in \mathbb{R}^{r\times d_2}$ such that $U = AS_1$ and $V = BS_2$, this reduces to:

$$\begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} 0 & S_1S_2^T \\ S_2S_1^T & 0 \end{bmatrix} \begin{bmatrix} A^T \\ B^T \end{bmatrix} = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} S_1S_1^T & 0 \\ 0 & S_2S_2^T \end{bmatrix} \begin{bmatrix} A^T \\ B^T \end{bmatrix} \Lambda. \tag{2.11}$$

A sufficient condition for (2.11) to hold is that

$$\begin{bmatrix} 0 & S_1S_2^T \\ S_2S_1^T & 0 \end{bmatrix} \begin{bmatrix} A^T \\ B_T \end{bmatrix} = \begin{bmatrix} S_1S_1^T & 0 \\ 0 & S_2S_2^T \end{bmatrix} \begin{bmatrix} A^T \\ B^T \end{bmatrix} \Lambda. \tag{2.12}$$

It was shown (see e.g. [69]) that equation (2.12), which can be solved as a generalized eigenvalue problem, can be used to determine the principal angles and directions associated with the row-spaces of S_1 and S_2. More computationally efficient methods exist [35] but are outside the scope of this thesis.

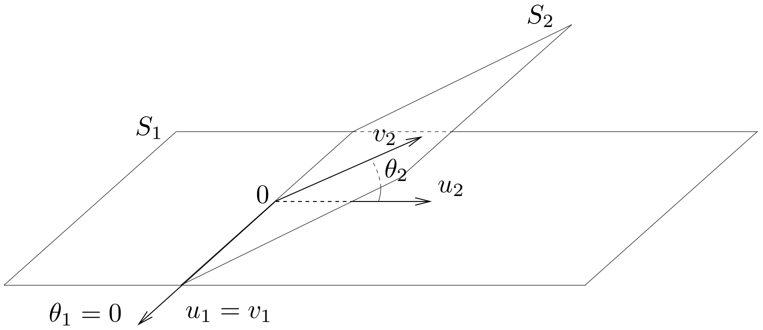


Figure 2.3: The principal angles between the two-dimensional subspaces S_1 and $S_2 \subset \mathbb{R}^3$. The first principal angle, θ_1, is zero and the first principal vectors, $u_1 \in S_1$ and $v_1 \in S_2$, coincide, revealing a one-dimensional intersection of S_1 and S_2. The second principal directions are orthogonal to the first principal directions: $u_2 \perp u_1$ and $v_2 \perp v_1$. The angle between $u_2 \in S_1$ and $v_2 \in S_2$ is θ_2.

2.4.2 Statistical theory of canonical correlation analysis

Given two statistical variables $x \in \mathbb{R}^{d_1}$ and $y \in \mathbb{R}^{d_2}$ with $E\{x\} = E\{y\} = 0$, Canonical Correlation Analysis is concerned with finding two matrices $A \in \mathbb{R}^{r\times d_1}$ and $B \in \mathbb{R}^{r\times d_2}$ such that the statistical variables Ax and By are maximally correlated. It is thereby required that $AE\left\{xx^T\right\}A^T = BE\left\{yy^T\right\}B^T = I_r$ and $AE\left\{xy^T\right\}B^T = \Lambda$ with Λ a diagonal matrix. The elements on the diagonal of Λ are called the canonical correlations, and the rows of Ax and By the canonical variates. It can be shown that the canonical correlations and variates of the random variables x and y can directly be obtained from the following generalized eigenvalue problem

$$\begin{bmatrix} 0 & E\left\{xy^T\right\} \\ E\left\{yx^T\right\} & 0 \end{bmatrix} \begin{bmatrix} A^T \\ B^T \end{bmatrix} = \begin{bmatrix} E\left\{xx^T\right\} & 0 \\ 0 & E\left\{yy^T\right\} \end{bmatrix} \begin{bmatrix} A^T \\ B^T \end{bmatrix} \Lambda. \qquad (2.13)$$

which bears a remarkable similarity with the generalized eigenvalue problem (2.12). From a comparison between (2.12) and (2.13) it is seen that if n measurements of the variables x and y are available and stacked in the columns of matrices S_1 and S_2, estimates for the canonical correlations and variates are directly found from (2.12). For $N \to \infty$ both generalized eigenvalue problems are equivalent. Due to this equivalence, in this thesis the term canonical correlation analysis will somewhat loosely be used not only to describe the statistical analysis itself but also as the search for principle angles and directions in a geometrical setting. From the context it will be clear which setting is intended.

2.4.3 Application: calculating the intersection of two row-spaces

In this thesis canonical correlation analysis will mostly be used to obtain the intersection of two row-spaces Row(A) and Row(B). As was also seen in Figure 2.3, in this intersection the principal directions of Row(A) and Row(B) are the same and their corresponding principal angles are zero. Hence, from a canonical correlation analysis on the rows of A and B, the intersection can directly be obtained as the space spanned by the principal directions corresponding to zero principal angles.

2.5 A condition number for the oblique projection

As was seen in Subsection 2.3.2, the oblique projection of the row-space of a matrix C onto the row space of a matrix $B \in \mathbb{R}^{n_B \times N}$ along the row-space of a matrix $A \in \mathbb{R}^{n_A \times N}$, with $n_A + n_B \le N$ and assuming that $n_A \le n_B, \operatorname{rank}(B) = n_B$, is given as:

$$C \underset{A}{/} B = C\mathcal{P}^T_{\{B^T|A^T\}} = C \begin{bmatrix} A \\ B \end{bmatrix}^\dagger \begin{bmatrix} 0 \\ B \end{bmatrix}.$$

From the discussion in Subsection 2.4.1 we know that bases V_A and V_B exist for Row(A) and Row(B) respectively so that:

$$V_A V_A^T = I_{n_A}, \;\; V_B V_B^T = I_{n_B}, \;\; V_A V_B^T = \begin{bmatrix} \Lambda & 0 \end{bmatrix}. \qquad (2.14)$$

Clearly, the choice of basis does not influence the result of the oblique projection. Hence,

$$\mathcal{P}^T_{\{B^T|A^T\}} = \mathcal{P}^T_{\{V_B^T|V_A^T\}} = \begin{bmatrix} V_A \\ V_B \end{bmatrix}^\dagger \begin{bmatrix} 0 \\ V_B \end{bmatrix}$$

and

$$\mathrm{Cond}_L\left(\mathcal{P}^T_{\{B^T|A^T\}}\right)^2 = \frac{\sigma_1\left(\mathcal{P}_{\{B^T|A^T\}}\mathcal{P}^T_{\{B^T|A^T\}}\right)}{\sigma_{n_B}\left(\mathcal{P}_{\{B^T|A^T\}}\mathcal{P}^T_{\{B^T|A^T\}}\right)},$$

with

$$\begin{aligned}
\mathcal{P}_{\{B^T|A^T\}}\mathcal{P}^T_{\{B^T|A^T\}} &= \left[\begin{array}{c|c} 0 & V_B^T \end{array}\right] \left[\begin{array}{c|cc} I & \Lambda & 0 \\ \hline \Lambda & I & 0 \\ 0 & 0 & I \end{array}\right]^{-1} \left[\begin{array}{c} 0 \\ \hline V_B \end{array}\right] \\
&= \left[\begin{array}{c|c} 0 & V_B^T \end{array}\right] \left[\begin{array}{c|cc} \frac{I}{I-\Lambda^2} & \frac{-\Lambda}{I-\Lambda^2} & 0 \\ \hline \frac{-\Lambda}{I-\Lambda^2} & \frac{I}{I-\Lambda^2} & 0 \\ 0 & 0 & I \end{array}\right] \left[\begin{array}{c} 0 \\ \hline V_B \end{array}\right] \\
&= V_B^T \begin{bmatrix} \frac{I}{I-\Lambda^2} & 0 \\ 0 & I \end{bmatrix} V_B .
\end{aligned}$$

Hence, the condition number of the linear operator $\mathcal{P}^T_{\{B^T|A^T\}}$ is given as:

$$\mathrm{Cond}_L\left(\mathcal{P}^T_{\{B^T|A^T\}}\right) = \frac{1}{\sin(\theta_{\mathrm{min}})}, \tag{2.15}$$

with θ_{min} the smallest principal angle between Row(A) and Row(B).

Remark: Note, as in the least squares case, that the condition number (2.15) is independent of the matrix C. The reason for this is explained in Subsections 2.1.2 and 2.1.4. As such, this does not mean that the conditioning of an oblique projection of the form $C/_{A}B$ is independent of C. In general, matrices C and δC can easily be found such that $C/_{A}B = 0$, $\delta C/_{A}B \neq 0$ and $\left\|\delta C/_{A}B\right\|_F / \left\|C/_{A}B\right\|_F = \infty$. The condition number (2.15) solely reflects the conditioning of the linear operator $\mathcal{P}^T_{\{B^T|A^T\}}$ according to the derivation in 2.1.4 where the possibility of $C/_{A}B = 0$ is explicitly excluded. Throughout this thesis, this conditioning measure will turn out to be sufficient to understand some practical problems that might turn up in subspace identification algorithms (see Chapter 5).

Part I

Subspace identification for linear systems

Chapter 3

Subspace identification

In this chapter linear subspace identification is introduced. Subspace identification is treated as an extension to the idea of realization, where the impulse response matrix is replaced by a set of free responses. This chapter will mainly focus on the identification of so-called combined stochastic-deterministic models, which are excited by measured inputs, as well as unmeasured disturbances or noise. The stochastic case will be seen to be a trivial extension with the inputs set to zero. At the end of the chapter, subspace identification algorithms based on separately parameterized stochastic and deterministic subsystems will be introduced as they will play an important role in the discussion on ill-conditioned subspace identification algorithms in Chapter 5

3.1 Introduction

This chapter is concerned with the identification of systems of the form

$$\begin{array}{rcl} x_{t+1} & = & Ax_t + Bu_t + w_t, \\ y_t & = & Cx_t + Du_t + v_t, \end{array} \tag{3.1}$$

with $u_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}^l$ the input and output of the system at time t, respectively. The so-called system state at time t is denoted by $x_t \in \mathbb{R}^n$. The system's dynamics are governed by the matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$ and $D \in \mathbb{R}^{l\times m}$. Unless otherwise stated, the process noise $w_t \in \mathbb{R}^n$ and output noise $v_t \in \mathbb{R}^l$ will be considered white, zero mean with second order moments

$$E\left\{\begin{bmatrix} w_t \\ v_t \end{bmatrix}\begin{bmatrix} w_k^T & v_k^T \end{bmatrix}\right\} = \begin{bmatrix} Q & R \\ R^T & S \end{bmatrix}\delta_{tk}. \tag{3.2}$$

Furthermore, w and v will be considered to be uncorrelated with the inputs:

$$E\left\{w_t u_k^T\right\} = 0, \;\; E\left\{v_t u_k^T\right\} = 0, \;\; \forall t, k. \tag{3.3}$$

The representation (3.1) is known as the state space representation [86]. The state space representation provides a convenient and compact way to model and analyze systems with multiple inputs and outputs. In addition, state space models are the preferred representation in modern control engineering [53, 88] where the control action is often written in terms of the state. Hence, many efforts have been undertaken in the last couple of decades to come up with system identification algorithms that directly estimate state space models from measured data. This as opposed to the so-called input-output representations, generally produced by predictor error identification methods [99]. Two powerful methodologies to estimate state space models from data are discussed in this chapter. They are known as the realization approach, and the subspace identification approach.

The estimation of the system matrices A, B, C and D from impulse response measurements, the so-called deterministic realization problem [79,165], is discussed in 3.2. In 3.3, we show that also in the case where no measured inputs are available, estimates for the system dynamics, and the noise covariance matrices Q, R and S can be obtained from a set of auto-covariances. The corresponding technique is known as stochastic realization [3, 49, 54].

In Section 3.4, an intuitive introduction into the theory of subspace identification is provided. It is shown that subspace identification algorithms are based on the same underlying ideas as realization algorithms. However, subspace identification algorithms offer many advantages, such as the fact that they can directly be applied to various kinds of measured data (no need to obtain impulse responses), and their greater robustness with respect to process- and measurement-noise.

A more rigorous description of subspace identification is provided in Sections 3.5 and 3.7. This description involves the classical unifying theorem for combined stochastic-deterministic subspace identification algorithms (Section 3.5), a unifying theorem for stochastic subspace identification algorithms (Section 3.6), and a description of separately parameterized state-space models and their usefulness (Section 3.7). Another point of discussion with respect to the unifying theorems just mentioned is the non-uniqueness of the state space representation (3.1). Replacing the state x_t by $\xi_t = Tx_t$ with $T \in \mathbb{R}^{n\times n}$ an invertible matrix in (3.1), we have:

$$\begin{array}{rcl} \xi_{t+1} & = & TAT^{-1}\xi_t + TBu_t + Tw_t, \\ y_t & = & CT^{-1}\xi_t + Du_t + v_t. \end{array} \tag{3.4}$$

Hence, the so-called similarity transformation T converts the system (3.1) into an equivalent representation with state ξ_t and system matrices TAT^{-1}, TB, CT^{-1} and D. Evidently, realization and subspace identification algorithms return but one of the possible representations. Ways to influence the particular representation that is obtained will be discussed. An application of subspace identification algorithms to a dataset from a glass oven is discussed in Section 3.8 A short summary of subspace identification algorithms, finally, will be given in Section 3.9.

3.2 Deterministic realization

The term "deterministic system" is generally used to describe systems which are purely driven by measured inputs. Hence, no disturbances or noise are taken into account. In deterministic realization one deals with linear deterministic systems of the form:

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t, \\ y_t &= Cx_t + Du_t. \end{aligned} \tag{3.5}$$

Given a finite series of impulse response matrices (also called the Markov parameters)

$$H_0 = D, \ \ H_t = CA^{t-1}B, \ \ t > 0, \tag{3.6}$$

the deterministic realization problem is formulated as follows:

> Find the minimal system order n and the system matrices A, B, C and D up to within a similarity transformation based on a finite number of impulse response samples.

The key to the solution of the deterministic realization problem is the factorization of a block Hankel matrix constructed from the impulse response matrices (see [79, 165]). It can easily be seen that for a given $i_1, i_2 \in \mathbb{Z}_0^+$

$$\begin{aligned} \mathcal{H}_{i_1,i_2} &= \begin{bmatrix} H_1 & H_2 & \dots & H_{i_2} \\ H_2 & H_3 & \dots & H_{i_2+1} \\ \vdots & \vdots & & \vdots \\ H_{i_1} & H_{i_1+1} & \dots & H_{i_1+i_2-1} \end{bmatrix} \\ &= \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i_1-1} \end{bmatrix} \begin{bmatrix} B & AB & \dots & A^{i_2-1}B \end{bmatrix} = \Gamma_{i_1}\mathcal{C}_{i_2}, \end{aligned}$$

with an implicit definition for the so-called extended observability matrix $\Gamma_{i_1}, i_1 \in \mathbb{Z}_0^+$ and the extended controllability matrix $\mathcal{C}_{i_2}, i_2 \in \mathbb{Z}_0^+$. For i_1 and i_2 sufficiently large, $\mathcal{H}_{i_1,i_2}$ is rank deficient and its rank is equal to the minimal system order. A possible realization for the matrices Γ_{i_1} and $\mathcal{C}_{i_2}$ can be obtained from a singular value decomposition

$$\mathcal{H}_{i_1,i_2} = USV^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},$$

as $\Gamma_{i_1} = U_1 S_1^{\frac{1}{2}} T$ and $\mathcal{C}_{i_2} = T^{-1} S_1^{\frac{1}{2}} V_1^T$, where S_1 contains the n non zero singular values of $\mathcal{H}_{i_1,i_2}$. T is a non-singular $n \times n$ matrix which symbolizes the freedom in the choice of basis as noted in the introduction of this chapter. Once Γ_{i_1} and

$\mathcal{C}_{i_2}$ are known, C and B can readily be extracted. In order to obtain the system matrix A, the following relation is commonly used

$$\underline{\Gamma_{i_1}} A = \overline{\Gamma_{i_1}}, \tag{3.7}$$

where $\underline{\Gamma_{i_1}}$ and $\overline{\Gamma_{i_1}}$ symbolize Γ_{i_1} with respectively the last l and first l rows removed. Hence, provided that Γ_{i_1} has full column rank, A can be obtained as $A = \underline{\Gamma_{i_1}}^{\dagger} \overline{\Gamma_{i_1}}$.

Obtaining the impulse response matrices from data

The theory of deterministic realization relies heavily on the availability of the impulse response matrices H_t, $t \in \mathbb{Z}^+$. In practical situations, these impulse response matrices are usually not available and need to be estimated from measured data $\{u_t, y_t\}, t = 0, \ldots, N-1$, e.g. by solving a least squares problem of the following form (for $i \in \mathbb{Z}_0^+$):

$$\begin{aligned}\begin{bmatrix}\widehat{H}_i & \widehat{H}_{i-1} & \ldots & \widehat{H}_0\end{bmatrix} = \arg \min_{H_i, \ldots, H_0} \Bigg\| & \begin{bmatrix} y_i & y_{i+1} & \cdots & y_{i+j-1}\end{bmatrix} \\ & - \begin{bmatrix} H_i & H_{i-1} & \ldots & H_0 \end{bmatrix} \begin{bmatrix} u_0 & u_1 & \ldots & u_{j-1} \\ u_1 & u_2 & \ldots & u_j \\ \vdots & \vdots & & \vdots \\ u_i & u_{i+1} & \ldots & u_{i+j-1} \end{bmatrix} \Bigg\|_F^2, \end{aligned} \tag{3.8}$$

with i and j arbitrary constants such that $i + j \leq N$. A drawback with this approach is that for finite i and a non-white input sequence u, one can easily show that the impulse response coefficients will in general not be consistently estimated. A more robust alternative will be found in the subspace identification methods introduced in Section 3.4. Hankel matrices as the one encountered in (3.8) will play a vital role in the derivation of subspace identification methods. It is therefore convenient to introduce a common notation for these matrices. We define:

$$U_{i|k} \triangleq \begin{bmatrix} u_i & u_{i+1} & \ldots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & \ldots & u_{i+j} \\ \vdots & \vdots & & \vdots \\ u_k & u_{k+1} & \ldots & u_{k+j-1} \end{bmatrix}, \quad i, k \in \mathbb{Z}^+, \;\; j \in \mathbb{Z}_0^+,$$

with a similar definition for $Y_{i|k}$. With these definitions problem (3.8) can be rewritten as

$$\begin{bmatrix}\widehat{H}_i & \widehat{H}_{i-1} & \ldots & \widehat{H}_0\end{bmatrix} = \arg \min_{H_i, \ldots, H_0} \left\| Y_{i|i} - \begin{bmatrix} H_i & H_{i-1} & \ldots & H_0 \end{bmatrix} U_{0|i} \right\|_F^2 .$$

3.3 Stochastic realization

The term "stochastic system" is generally used to describe systems which are purely driven by unmeasured inputs. In stochastic realization one deals with linear stochastic systems of the form:

$$\begin{aligned} x_{t+1} &= Ax_t + w_t, \\ y_t &= Cx_t + v_t, \end{aligned} \tag{3.9}$$

with w_t and v_t as in Section 3.1. Defining $\Lambda_i \triangleq E\left\{y_{t+i}y_t^T\right\}$ as the output covariance matrices, the stochastic realization problem is now formulated as follows:

> Given a finite set of output covariance matrices, find the minimal system order n, the system matrices A and C up to within a similarity transformation, and the noise covariance matrices Q, R and S for the system (3.9).

This problem was studied intensively in [3, 49, 54]. A solution is found by introducing $G = E\left\{x_{t+1}y_t^T\right\}$. With this definition it can easily be seen that

$$\Lambda_i = CA^{i-1}G, \;\; i > 0.$$

Hence, the output covariance matrices can be considered as the Markov parameters of a deterministic system with system matrices A, G, C and Λ_0 (see also (3.6)). Applying the theory of deterministic realization (with $i_1 = i_2 = i$) leads to

$$\begin{aligned} \mathcal{H}_{i,i} &= \begin{bmatrix} \Lambda_1 & \Lambda_2 & \dots & \Lambda_i \\ \Lambda_2 & \Lambda_3 & \dots & \Lambda_{i+1} \\ \vdots & \vdots & & \vdots \\ \Lambda_i & \Lambda_{i+1} & \dots & \Lambda_{2i-1} \end{bmatrix} \\ &= \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix} \begin{bmatrix} G & AG & \dots & A^{i-1}G \end{bmatrix}, \end{aligned} \tag{3.10}$$

from which A, G, C and Λ_0 can easily be recovered. Extracting the noise covariances Q, S and R is somewhat more involved. One possibility is to use a particular representation of the system (3.9), known as the forward innovation model by applying a Kalman filter [88] to (3.9). The resulting model in forward innovation form is:

$$\begin{aligned} \xi_{t+1} &= A\xi_t + Ke_t, \\ y_t &= C\xi_t + e_t, \end{aligned}$$

with $\{e_t\}$ a white noise sequence with covariance matrix $\Sigma = E\left\{e_te_t^T\right\}$ and $K = (G - APC^T)\Sigma^{-1}$ the so-called Kalman filter gain. It can be shown that the forward innovation model has the same statistical properties as the system (3.9) [54, 113]. Denoting $P = E\left\{\xi_t \xi_t^T\right\}$ one has the following important relations:

$$\begin{aligned} P &= APA^T + K\Sigma K^T, && (3.11)\\ \Lambda_0 &= CPC^T + \Sigma, && (3.12)\\ K &= (G - APC^T)\Sigma^{-1}, && (3.13) \end{aligned}$$

from which

$$\begin{aligned} P &= APA^T + (G - APC^T)\Sigma^{-1}(G - APC^T)^T \\ &= APA^T + (G - APC^T)(\Lambda_0 - CPC^T)^{-1}(G - APC^T)^T. \quad (3.14) \end{aligned}$$

The latter equation is known as a Riccati equation and can be solved for P with fairly standard techniques (see e.g. [95] for a robust algorithm). Once P is known, Σ can be extracted from (3.12) and K from (3.13). A realization for the noise covariance matrices Q, S and R is now found as:

$$Q = K\Sigma K^T, \quad R = K\Sigma, \quad S = \Sigma. \tag{3.15}$$

Obtaining the output auto-covariance matrices from data

The output-covariance matrices can conveniently be obtained from data as follows:

$$\lim_{j\to\infty} \frac{1}{j} Y_{i|2i-1} Y_{0|i-1}^T = \begin{bmatrix} \Lambda_i & \dots & \Lambda_2 & \Lambda_1 \\ \Lambda_{i+1} & \dots & \Lambda_3 & \Lambda_2 \\ \vdots & & \vdots & \vdots \\ \Lambda_{2i-1} & \dots & \Lambda_{i+1} & \Lambda_i \end{bmatrix} = \Gamma_i \begin{bmatrix} A^{i-1}G & \dots & AG & G \end{bmatrix},$$

which yields exactly the same decomposition as seen for $\mathcal{H}_{i,i}$ except for an inversion of the order of the columns. Introducing the following convenient notation:

$$Y_p \triangleq Y_{0|i-1}, \qquad Y_f \triangleq Y_{i|2i-1}, \tag{3.16}$$

which we will refer to as the past and future output block Hankel matrices, this can also be written as:

$$\lim_{j\to\infty} \frac{1}{j} Y_f Y_p^T = \Gamma_i \begin{bmatrix} A^{i-1}G & \dots & AG & G \end{bmatrix}. \tag{3.17}$$

Note that an inversion of the rows of Y_p would yield $\lim_{j\to\infty} \frac{1}{j} Y_f Y_p^T = \mathcal{H}_{i,i}$. However, in subspace identification algorithms, it is more common to define Y_p as in (3.16). In order to obtain a consistent notation throughout this thesis, we will therefore continue to work with the definition (3.16). Similarly, for the inputs we define the past and future block Hankel matrices as

$$U_p \triangleq U_{0|i-1}, \qquad U_f \triangleq U_{i|2i-1}.$$

Two further notations will be commonly used in the following discussion, namely the joint past W_p and the joint future W_f, defined as

$$W_p \triangleq \begin{bmatrix} U_p \\ Y_p \end{bmatrix}, \quad W_f \triangleq \begin{bmatrix} U_f \\ Y_f \end{bmatrix}.$$

3.4 Intuition behind subspace identification

In this section, we will provide an intuitive overview of subspace identification algorithms. It will be shown that the key idea behind subspace identification algorithms is that estimates for the extended observability matrix and the system states can be obtained from a set of free responses of the system. Several ways to obtain such free responses are introduced and their relation to existing subspace identification algorithms discussed. It is important to stress that the sole aim of this section is to outline the basic concepts behind subspace identification algorithms in an as easily accessible way as possible. Where necessary, complexity will be traded for clarity in the presentation to obtain this goal. It is also important to note that this section will not discuss the extraction of the system matrices A, B, C and D from the extended observability and the system states. This extraction, combined with a more rigorous mathematical analysis of subspace algorithms is postponed until Section 3.5.

3.4.1 Free responses of a system

Given a noiseless linear $n-$th order state-space system of the form

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t, \\ y_t &= Cx_t + Du_t, \end{aligned} \tag{3.18}$$

we have for any $t_0 \in \mathbb{Z}$ and $i \in \mathbb{Z}_0^+$ that

$$\begin{aligned} \begin{bmatrix} y_{t_0} \\ y_{t_0+1} \\ \vdots \\ y_{t_0+i-1} \end{bmatrix} &= \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix} x_{t_0} + \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \vdots & \vdots & & \vdots \\ CA^{i-2}B & CA^{i-3}B & \dots & D \end{bmatrix} \begin{bmatrix} u_{t_0} \\ u_{t_0+1} \\ \vdots \\ u_{t_0+i-1} \end{bmatrix} \\ &= \Gamma_i x_{t_0} + H_i \begin{bmatrix} u_{t_0}^T & \dots & u_{t_0+i-1}^T \end{bmatrix}^T, \end{aligned} \tag{3.19}$$

with an implicit definition for H_i. Hence, the system outputs $y_{t_0} \dots y_{t_0+i-1}$ are fully determined by the state x_{t_0} and the inputs $u_{t_0} \dots u_{t_0+i-1}$. The vector $\Gamma_i x_{t_0}$ contains the so-called free response of the system generated by x_{t_0}. The free response is the system output that would be obtained if the inputs $u_{t_0} \dots u_{t_0+i-1}$ were identically zero, and it will play the same central role in subspace identification algorithms as the impulse response in realization algorithms [103]. Its potential can be seen by assuming that multiple free responses for the system (3.18) are available (e.g. through measurements). Denoting the r^{th} free response as $y^{(r)}$, equation (3.19) becomes

$$\begin{bmatrix} y_{t_0}^{(1)} & y_{t_0}^{(2)} & \cdots & y_{t_0}^{(j)} \\ y_{t_0+1}^{(1)} & y_{t_0+1}^{(2)} & \cdots & y_{t_0+1}^{(j)} \\ \vdots & \vdots & & \vdots \\ y_{t_0+i-1}^{(1)} & y_{t_0+i-1}^{(2)} & \cdots & y_{t_0+i-1}^{(p)} \end{bmatrix} = \Gamma_i \begin{bmatrix} x_{t_0}^{(1)} & \ldots & x_{t_0}^{(j)} \end{bmatrix}, \tag{3.20}$$

with j the total number of available responses. Hence, the free response matrix at the left hand side of equation (3.20) is at most rank n. If it is rank n, the extended observability matrix, and the generating states $x_{t_0}^{(1)} \ldots x_{t_0}^{(j)}$ can be obtained by performing a singular value decomposition. As was the case for the realization algorithm, the results of this operation will only be determined up to a similarity transformation.

Obviously, in most practical cases, measuring free responses is a time-consuming, if not impossible option. In such cases, the best one can hope for is to obtain a good free response matrix from input-output measurements. Obtaining a free response matrix from data will be the prime objective in subspace identification algorithms.

The primary objective of subspace identification algorithms is to obtain a free response matrix from measured data. Estimates for the extended observability matrix and the generating states can be obtained from a low rank approximation of this matrix.

3.4.2 Obtaining free responses from data

With $j \geq i \geq n \in \mathbb{Z}_0^+$ and the block Hankel matrix notation introduced in 3.2 and 3.3, it follows from (3.19) that

$$Y_f \;=\; \Gamma_i X_i + H_i U_f, \tag{3.21}$$

with $X_i = \begin{bmatrix} x_i & x_{i+1} & \ldots x_{i+j-1} \end{bmatrix}$. Hence, the future outputs are fully determined by a set of free responses and the future inputs. This situation is graphically depicted in Figure 3.1. Obtaining free responses for the system (3.18) is now a matter of removing the influence of the future inputs from (3.21). Several ways to achieve this are briefly discussed below.

The projection algorithm

An obvious way to remove the influence of the future inputs is to project (3.21) onto the orthogonal complement of the future inputs [27,38,43,97,138,148,154]. In this case:

$$Y_f/U_f^{\perp} = \Gamma_i \left(X_i/U_f^{\perp} \right), \tag{3.22}$$

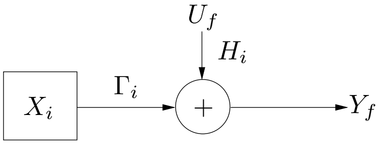


Figure 3.1: The future outputs are fully determined by a set of free responses and the future inputs.

from which an estimate for Γ_i and the part of the state in the orthogonal complement of U_f can be obtained. Although this observation is limited to the noiseless case, it can be shown that the projection presented above delivers consistent estimates if the output is perturbed by white noise. This can easily be understood from the fact that the projection (3.22) is basically a least squares problem which is known to be consistent if the output (in this case Y_f) is perturbed by white noise. However, in the presence of colored output noise and/or process noise, the projection algorithm will lead to biased results. Further improvements to this algorithm will largely focus on removing such noise contributions.

The PI-MOESP algorithm

An improvement to the projection algorithm is found by realizing that for any stable system of the form (3.18), and any $t_0 \in \mathbb{Z}$, the state x_{t_0} is uniquely determined by the system inputs up to time $t_0 - 1$. Hence, with a slight abuse of notation equation (3.21) can be rewritten as:

$$Y_f = \Gamma_i L_\infty U_{-\infty|i-1} + H_i U_f,$$

with L_∞ a linear operator so that $L_\infty U_{-\infty|i-1} = X_i$. However, in practical cases, one never has access to an infinite amount of data. In the PI-MOESP algorithm [152, 154, 155], the matrix $U_{-\infty|i-1}$ is therefore replaced by $U_{0|i-1} = U_p$, and the state sequence is replaced by its best possible estimate $\widehat{X}_i = L_p U_p$ based on the data in U_p. We have:

$$Y_f \simeq \Gamma_i \widehat{X}_i + H_i U_f = \Gamma_i L_p U_p + H_i U_f.$$

Hence, the output is approximated as the sum of two contributions. One from the past inputs, which essentially determines the state sequence estimate $\widehat{X}_i$, and one from the future inputs. This situation is graphically depicted in Figure 3.2. Mathematically, the approximation as the sum of a past and a future contribution is obtained by performing a projection of (3.21) onto the space spanned by the rows of $\begin{bmatrix} U_p^T & U_f^T \end{bmatrix}^T$ followed by a projection onto $U_f^\perp$ to remove the influence of the future inputs. It can be proven that for $j \to \infty$ and under a condition on the inputs known as persistency of excitation (see Section 3.7) we have

$$Y_f / \begin{bmatrix} U_p \\ U_f \end{bmatrix} / U_f^{\perp} = Y_f / (U_p / U_f^{\perp}) = \Gamma_i X_i / (U_p / U_f^{\perp}),$$

and this even in the presence of colored process and/or output noise. The strength of the algorithm is explained by the fact that any noise contribution is immediately removed by the initial projection on the inputs. The disadvantage of the PI-MOESP algorithm is that large parts of the dynamics are also removed in the initial projection, such as the dynamics due to noise (the so-called stochastic subsystem), and the part of the state which is not contained in $U_p/U_f^{\perp}$. Consequently, PI-MOESP results are often seen to exhibit rather large uncertainties (variances) on the obtained parameters.

The PO-MOESP algorithm

The PO-MOESP algorithm [153] is essentially the same as the PI-MOESP algorithm with the sole exception that the state is now replaced with its best possible estimate based on the past input and output data in W_p. This might seem strange, as the state vector X_i is not directly influenced by the past outputs. Nevertheless, such estimates are quite common in system theory and generally referred to as state observers. Assuming $\widehat{X}_i = L_p W_p$, we have:

$$Y_f \simeq \Gamma_i \widehat{X}_i + H_i U_f = \Gamma_i L_p W_p + H_i U_f. \tag{3.23}$$

Again, the output is approximated as the sum of a past and a future contribution. However, in contrast to the PI-MOESP approach, the PO-MOESP approach uses both the past inputs and past outputs to approximate the state which should lead to smaller variances on the obtained parameters. In line with the PI-MOESP algorithm, a free response matrix is obtained by projecting (3.21) onto the space spanned by the rows of $\begin{bmatrix} W_p^T & U_f^T \end{bmatrix}^T$ followed by a projection onto $U_f^{\perp}$. The algorithm generates consistent estimates for Γ_i in the presence of white process and/or output noise. The interpretation of the obtained state is somewhat more complicated than in the PI-MOESP case. It will be shown in Subsection 3.5.4 that the state can be seen as the result of a non-steady state Kalman filter working in parallel on the columns of W_p. For purely deterministic systems, it can be shown that $X_i = \widehat{X}_i = L_p W_p$. The basic

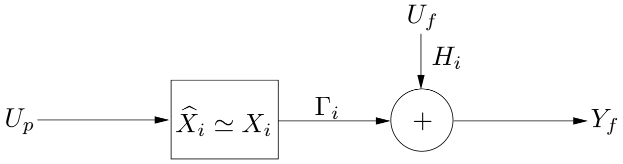


Figure 3.2: The future outputs can be decomposed along the future inputs and past inputs contained in U_p. The latter part is equal to $\Gamma_i \widehat{X}_i$ with $\widehat{X}_i$ an estimate for the state.

decomposition at the heart of the PO-MOESP algorithm is graphically depicted in Figure 3.3.

Note that the class of systems for which consistent estimates can be obtained is a subset of the class which yields consistent estimates under the PI-MOESP algorithm. However, the PO-MOESP has the advantage that most of the internal dynamics are retained in the projection onto $\begin{bmatrix} W_p^T & U_f^T \end{bmatrix}^T$. This allows for the estimation of the stochastic subsystem together with the deterministic one. Combined with the fact that a greater part of the state is conserved this generally leads to results with smaller variances on the obtained parameters.

The N4SID algorithm

The N4SID algorithm [144, 147] is based on the same decomposition as the PO-MOESP algorithm, graphically depicted in Figure 3.3. However, whereas the PO-MOESP algorithm features an orthogonal projection onto $U_f^\perp$ after the initial projection on $\begin{bmatrix} W_p^T & U_f^T \end{bmatrix}^T$, the N4SID approach simply retains only the part $\Gamma_i L_p W_p$ by performing an oblique projection $Y_f /_{U_f} W_p$. This approach yields qualitatively the same result as the PO-MOESP, namely the estimation of the $\Gamma_i \widehat{X}_i$-term in (3.23). In fact, it is easily proven that.

$$\left(Y_f /_{U_f} W_p\right) / U_f^\perp = Y_f / \begin{bmatrix} W_p \\ U_f \end{bmatrix} / U_f^\perp = Y_f / (W_p / U_f^\perp).$$

Hence, the state obtained in the PO-MOESP algorithm is identical to the state obtained in the N4SID algorithm up to a projection onto $U_f^\perp$. As in the PO-MOESP case, the algorithm provides consistent estimates for Γ_i in the presence of white process and/or output noise.

The intersection algorithms

An algorithm which clearly distinguishes itself from the algorithms so far introduced is the intersection algorithm of which the main ingredients were introduced in numerous papers [38–42, 44, 106, 108, 162–164]. The intersection algorithm is different in a sense that the states are not obtained from a projection, but as the intersection of two row-spaces. In the discussion on

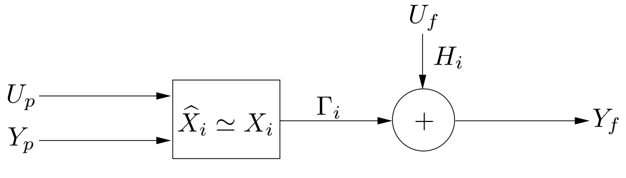


Figure 3.3: The future outputs can be decomposed along the future inputs and past measurements contained in U_p and Y_p. The latter part is equal to $\Gamma_i \widehat{X}_i$ with $\widehat{X}_i$ an estimate for the state.

the PO-MOESP and the N4SID algorithm, we have stated that there exists a matrix $L_p \in \mathbb{R}^{n\times i(m+l)}$ such that for purely deterministic systems

$$X_i = L_p W_p. \tag{3.24}$$

On the other hand, from $\Gamma_i X_i = Y_f - H_i U_f$ and assuming that Γ_i is of full rank, it easily follows that the row space of X_i is contained in the union of the row spaces of U_f and Y_f. Hence, there exists a matrix $L_f \in \mathbb{R}^{n\times i(m+l)}$ such that:

$$X_i = L_f \begin{bmatrix} Y_f \\ U_f \end{bmatrix} = L_f W_f. \tag{3.25}$$

The row-space of the state is contained in the row-space of the past, and in the row-space of the future, and is therefore contained in the intersection of the past and the future, which can for instance be calculated using a technique known as canonical correlation analysis introduced in Section 2.4. For the noiseless case, it was proven in [42] that this intersection has dimension n and indeed represents a valid state sequence. From the state sequence, matrices A, B, C and D can be obtained. The intersection algorithm generates consistent estimates for the system matrices, even if the inputs and outputs are corrupted by white noise. However, in the latter case the input and output noise must have equal covariance matrices, which severely limits the applicability of the intersection method. Intersection algorithms are therefore not very commonly used in practical applications. However, in Chapter 9 it will be shown that despite their shortcomings, intersection algorithms might be very useful in deriving subspace identification techniques for the class of Hammerstein-Wiener systems.

The CVA algorithm

Similar to the intersection method, but independently developed is the so-called CVA method, which is an abbreviation for ‘Canonical Variate Analysis’. The analysis of the CVA-algorithm is quite involved and we refer to [94, 121] for a full introduction. It suffices to say that from (3.24) and (3.25) it follows directly that in the deterministic case:

$$\begin{aligned} X_i/U_f^{\perp} &= L_p W_p/U_f^{\perp} \\ X_i/U_f^{\perp} &= L_f W_f/U_f^{\perp} = L_f Y_f/U_f^{\perp}. \end{aligned}$$

Hence, an estimate for $X_i/U_f^{\perp}$ can be obtained from the intersection of $Y_f/U_f^{\perp}$ and $W_p/U_f^{\perp}$. In the original derivation of the CVA method, this intersection is obtained using a canonical correlation analysis. The method is proven to be consistent, even in the case of white process and/or output noise.

3.5 A rigorous derivation of subspace identification

In this section, we will provide a rigorous derivation of subspace identification. Several of the earlier introduced subspace identification algorithms will be discussed in greater detail, and necessary proofs and justifications for claims made in the intuitive overview will be provided.

3.5.1 System description

As introduced in Section 3.1, this chapter is concerned with the identification of systems of the form

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_t, \\ y_t &= Cx_t + Du_t + v_t. \end{aligned} \tag{3.26}$$

Unless otherwise stated, the process noise $w_t \in \mathbb{R}^n$ and output noise $v_t \in \mathbb{R}^l$ is considered white, zero mean with second order moments

$$E\left\{ \begin{bmatrix} w_t \\ v_t \end{bmatrix} \begin{bmatrix} w_k^T & v_k^T \end{bmatrix} \right\} = \begin{bmatrix} Q & R \\ R^T & S \end{bmatrix} \delta_{tk}.$$

Furthermore, w and v are considered to be uncorrelated with the inputs:

$$E\left\{ w_t u_k^T \right\} = 0, \quad E\left\{ v_t u_k^T \right\} = 0, \quad \forall t, k.$$

$\{A, C\}$ is assumed to be observable with observable modes that can be either stable or unstable. $\{A, \begin{bmatrix} B & Q^{\frac{1}{2}} \end{bmatrix}\}$ is assumed to be controllable with controllable modes that are stable. The dimensions of all matrices and vectors appearing in (3.26) are as introduced in Section 3.1.

The system (3.26) is essentially a combination of a deterministic subsystem, driven by the input u and a stochastic subsystem influenced by w and v. Consequently, the state and output can be split up in a deterministic part and a stochastic part as $x_t = x_t^d + x_t^s$ and $y_t = y_t^d + y_t^s$. The deterministic subsystem is governed by the following state space equation:

$$\begin{aligned} x_{t+1}^d &= Ax_t^d + Bu_t, \\ y_t^d &= Cx_t^d + Du_t. \end{aligned}$$

The stochastic subsystem has the following form:

$$\begin{aligned} x_{t+1}^s &= Ax_t^s + w_t, \\ y_t^s &= Cx_t^s + v_t. \end{aligned} \tag{3.27}$$

Note that in the above, it is assumed that the deterministic and stochastic subsystem share the same dynamics governed by the matrices A and C. The corresponding model structure (3.26) is referred to as a ‘jointly-parameterized’ model or a combined stochastic-deterministic model. Such a combined model can always be obtained, even if the stochastic subsystem were governed by system matrices A^s, C^s which are different from the matrices A^d and C^d used in the deterministic system. In this case on would stack the deterministic and the stochastic state in one vector as follows:

$$\begin{aligned}\begin{bmatrix} x_{t+1}^d \\ x_{t+1}^s \end{bmatrix} &= \begin{bmatrix} A^d & 0 \\ 0 & A^s \end{bmatrix} \begin{bmatrix} x_t^d \\ x_t^s \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_t + \begin{bmatrix} 0 \\ w_t \end{bmatrix}, \\ y_t &= \begin{bmatrix} C^d & C^s \end{bmatrix} \begin{bmatrix} x_t^d \\ x_t^s \end{bmatrix} + Du_t + v_t. \end{aligned} \tag{3.28}$$

The obtained system would of course not be guaranteed to be minimal. The advantage of combining the dynamics of both subsystems in one single model structure is that common dynamics which are excited by u as well as w are naturally handled, whereas separate dynamics are easily incorporated as seen above. Furthermore, in some applications a jointly parameterized model is preferred over a separately parameterized one, for instance in modal analysis where the estimates for the system matrix A are used to extract resonances of a vibrating system. In this case it makes little sense to model the same dynamics twice. It is for all these reasons that most subspace identification approaches identify jointly parameterized models. The description of these subspace identification approaches, which include the highly popular N4SID and PO-MOESP will take up the largest part of the following discussion. We will come back to the separately parameterized models in Section 3.7.

3.5.2 Notation and input-output equations

We will largely keep the notation that was introduced at the beginning of this chapter. However where necessary superscripts $(\cdot)^d$ and $(\cdot)^s$ will be provided to make a distinction between elements of the deterministic subsystem and elements of the stochastic subsystem. The matrix Y_p^s for example is similar to the matrix Y_p, introduced in (3.16), except that the former is filled with the stochastic component y^s of the outputs y instead of the entire output. In addition we introduce:

$$\begin{aligned} X_p^d &\triangleq X_0^d \triangleq \begin{bmatrix} x_0^d & x_1^d & \dots & x_{j-1}^d \end{bmatrix}, \\ X_p^s &\triangleq X_0^s \triangleq \begin{bmatrix} x_0^s & x_1^s & \dots & x_{j-1}^s \end{bmatrix}, \\ X_f^d &\triangleq X_i^d \triangleq \begin{bmatrix} x_i^d & x_{i+1}^d & \dots & x_{i+j-1}^d \end{bmatrix}, \\ X_f^s &\triangleq X_i^s \triangleq \begin{bmatrix} x_i^s & x_{i+1}^s & \dots & x_{i+j-1}^s \end{bmatrix}, \end{aligned}$$

as the past and future deterministic and stochastic state sequences, and

$$\begin{aligned} \Delta_i^d &\triangleq \begin{bmatrix} A^{i-1}B & \dots & AB & A \end{bmatrix}, \\ \Delta_i^s &\triangleq \begin{bmatrix} A^{i-1}K & \dots & AK & A \end{bmatrix}, \end{aligned}$$

as the reversed extended controllability matrices of the deterministic and the stochastic subsystem (in forward innovation form). With these notations, the matrix input-output equations are summarized in the following theorem:

Theorem 3.1. *Combined matrix input-output equations:*

$$
\begin{aligned}
Y_p &= \Gamma_i X_p^d + H_i U_p + Y_p^s, && (3.29)\\
Y_f &= \Gamma_i X_f^d + H_i U_f + Y_f^s, && (3.30)\\
X_f^d &= A^i X_p^d + \Delta_i^d U_p. && (3.31)
\end{aligned}
$$

Proof. Easily checked by recursive substitution into the state space equation (3.26). □

Note that equation (3.30) is basically equal to (3.21), except for the additional contributions due to the stochastic subsystem stored in the Hankel matrix Y_f^s.

3.5.3 Orthogonal projection on past and future data

In Subsection 3.4.2, it was seen that both the N4SID and PO-MOESP algorithm feature an initial projection onto the space spanned by the rows of W_p and U_f. In this subsection this projection will be studied in some more detail. By defining the matrix Z_i as:

$$Z_i = Y_f / \begin{bmatrix} U_f \\ W_p \end{bmatrix},$$

the following theorem sums up the main properties of the projection:

Theorem 3.2. *Initial projection:*

- *Given the assumptions on the noise:* $E\left\{w_t u_k^T\right\} = 0,\ E\left\{v_t u_k^T\right\} = 0,\ \forall t, k,$
- *and if the input is persistently exciting of order* $2i$ *([99], p. 363), meaning that rank* $\left(\left[U_{0|2i-1}\right]\right) = 2mi$,
- *and if* $j \to \infty$, *then:*

$$Z_i = \Gamma_i \widehat{X}_i + H_i U_f,$$

with

$$\widehat{X}_i = \left[\begin{array}{c|c|c} A^i - \Omega_i \Gamma_i & \Delta_i^d - \Omega_i H_i & \Omega_i \end{array}\right] \left[\begin{array}{c} X_u^d \\ \hline U_p \\ \hline Y_p \end{array}\right], \tag{3.32}$$

whereby

$$
\begin{aligned}
\Omega_i &= \chi_i \psi_i^{-1},\\
\chi_i &= A^i (P^d - P_u^d) \Gamma_i^T + \Delta_i^s,\\
\psi_i &= \Gamma_i (P^d - P_u^d) \Gamma_i^T + L_i^s,
\end{aligned}
$$

and $X_u^d = X_p^d / \begin{bmatrix} U_p \\ U_f \end{bmatrix}$, $P^d = \frac{1}{j} X_p^d X_p^{d^T} = E\left\{x_t^d x_t^{d^T}\right\}$, $P_u^d = \frac{1}{j} X_u^d X_u^{d^T}$, $L_i^s = \frac{1}{j} Y_p^s Y_p^{s^T}$.

If the stochastic subsystem is identically zero ($L_i^s = \Delta_i^s = 0$), we have

$$\widehat{X}_i = X_i^d = \left[\begin{array}{c|c} \Delta_i^d - A^i\Gamma_i^\dagger H_i & A^i\Gamma_i^\dagger \end{array}\right] \left[\begin{array}{c} U_p \\ \hline Y_p \end{array}\right],$$

Proof. The general proof is well documented in [144, 147]. The case where the stochastic system is identically zero can directly be derived from the matrix input-output equations (3.29)-(3.31) and by setting $L_i^s = \Delta_i^s = 0$ in (3.32). ☐

Note from (3.32) that the state estimate $\widehat{X}_i$ is largely determined by the past data W_p, as desired. In case of a noiseless system, we can write $\widehat{X}_i = X_i = L_p W_p$ as predicted in the intuitive description of the PO-MOESP and the N4SID algorithm. Two properties remain to be explored if noise is present:

- It was stated in Subsection 3.4.2 that the obtained state estimate $\widehat{X}_i$ is a Kalman filter state. This property will be investigated in Subsection 3.5.4.
- X_u^d at the right hand side of (3.32) contains a contribution from the future U_f. Projecting away the future inputs as is done in the orthogonal projection on $U_f^\perp$ in PO-MOESP or the oblique projection in N4SID will remove this component. The consequences of this are explored in Subsection 3.5.5.

3.5.4 Relation to the Kalman filter

In this subsection, it will be shown that the sequence $\widehat{X}_i$ can be interpreted in terms of a bank of j non steady state Kalman filters, applied in parallel to the data. The non-steady state Kalman filter can be expressed as follows:

$$\begin{aligned}
\hat{x}_t &= A\hat{x}_{t-1} + Bu_t + K_{t-1}(y_{t-1} - C\hat{x}_{t-1} - Du_{t-1}), && (3.33)\\
K_{t-1} &= (G - AP_{t-1}C^T)(\Lambda_0 - CP_{t-1}C^T)^{-1}, && (3.34)\\
P_t &= AP_{t-1}A^T + (G - AP_{t-1}C^T) \\
&\quad (\Lambda_0 - CP_{t-1}C^T)^{-1}(G - AP_{t-1}C^T), && (3.35)
\end{aligned}$$

where $\hat{x}_t$ denotes the Kalman filter state estimate at time t, K_t is the so-called Kalman filter gain, $G = E\left\{x_{t+1}^s y_t^T\right\}$, $\Lambda_0 = E\left\{y_t^s y_t^{sT}\right\}$, $P_t = P^s - \widetilde{P}_t$ with $P^s = E\left\{x_t^s x_t^{sT}\right\}$ and $\widetilde{P}_t = E\left\{(\hat{x}_t - x_t)(\hat{x}_t - x_t)^T\right\}$ the error covariance matrix of the estimated state. Note that the set of equations (3.33-3.35) is slightly different from the classical formulation of the Kalman filter as it is for instance found in [9]. In its classical formulation, the Kalman filter is expressed in terms of the matrices $\widetilde{P}_t$ rather than $P_t = P^s - \widetilde{P}_t$. Nevertheless, the set of equations (3.33-3.35) is mathematically equivalent to the classical Kalman filter formulations (see Appendix A) and is more useful in our derivations.

The recursive set of formulas (3.33-3.35) is initialized with a certain initial state $\hat{x}_0$ and initial matrix P_0, which are in practical applications chosen by the user. The following theorem allows to link the non-steady state Kalman filter to the sequence $\widehat{X}_i$ obtained by projecting Y_f onto the space spanned by the rows of W_p and U_f.

Theorem 3.3. *Given an initial state estimate $\hat{x}_0$, an initial estimate of the matrix P_0 and input output measurements $u_0, y_0, \ldots, u_{i-1}, y_{i-1}$, then the non-steady state Kalman filter state estimate $\hat{x}_i$ at time i can be explicitly written as:*

$$\hat{x}_i = \left[\begin{array}{c|c|c} A^i - Q_i\Gamma_i & \Delta_i^d - \Omega_i H_i & Q_i \end{array}\right] \left[\begin{array}{c} \hat{x}_0 \\ \hline u_0 \\ \vdots \\ u_{i-1} \\ \hline y_0 \\ \vdots \\ y_{i-1} \end{array}\right],$$

where

$$Q_i = (\Delta_i^s - A^i P_0 \Gamma_i^T)(L_i^s - \Gamma_i P_0 \Gamma_i^T)^{-1}.$$

Proof. The proof is well documented in [144, 147]. □

Applying Theorem 3.3 in parallel to the elements of a vector $\widehat{X}_0 = \begin{bmatrix} \hat{x}_0 & \hat{x}_1 & \ldots & \hat{x}_{j-1} \end{bmatrix}$, we obtain:

$$\widehat{X}_i = \left[\begin{array}{c|c|c} A^i - \Omega_i\Gamma_i & \Delta_i^d - \Omega_i H_i & \Omega_i \end{array}\right] \left[\begin{array}{c} \widehat{X}_0 \\ \hline U_p \\ \hline Y_p \end{array}\right], \tag{3.36}$$

Equation (3.36) is essentially the same as (3.32) with the following substitutions:

$$\widehat{X}_0 = X_u^d, \tag{3.37}$$

$$P_0 = P_u^d - P^d. \tag{3.38}$$

Hence, $\widehat{X}_i$ in (3.32) can indeed be seen as the result of a non-steady state Kalman filter with initialization given by (3.37-3.38). We write $\widehat{X}_i = \widehat{X}_{i[X_0,P_0]} = \widehat{X}_{i[X_u^d, P_u^d - P^d]}$. The situation is graphically depicted in Figure 3.4.

3.5.5 A unifying theorem

From the discussion in Subsection 3.5.3 and Subsection 3.5.4 it can now be seen that under the assumptions of Theorem 3.2,

$$\begin{aligned} Y_f/_{U_f} W_p &\underset{j\to\infty}{=} \Gamma_i \widehat{X}_{i[X_p^d/_{U_f} U_p, P_u^d - P^d]} \\ Y_f/(W_p/U_f^\perp) &\underset{j\to\infty}{=} Z_i/U_f^\perp = \Gamma_i \widehat{X}_{i[X_p^d/(U_p/U_f^\perp), P_u^d - P^d]}. \end{aligned}$$

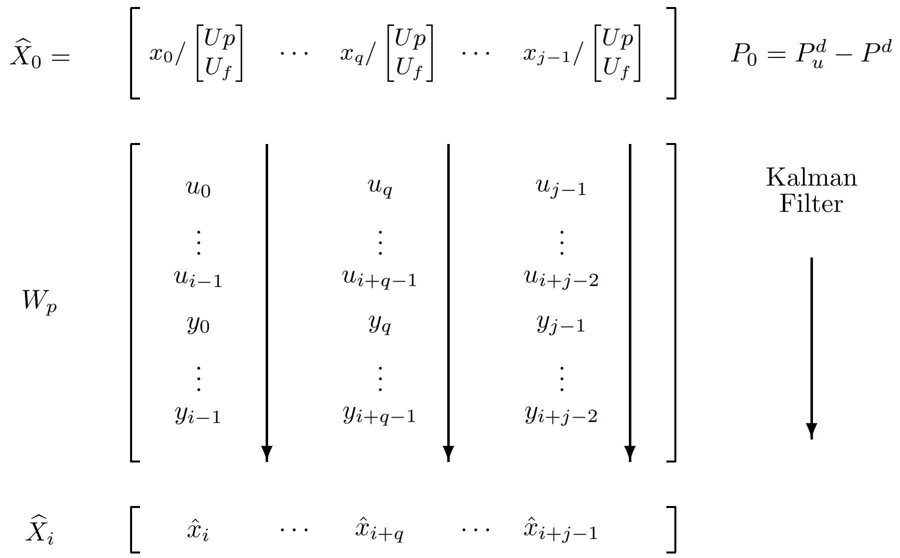

$\widehat{X}_0 =$
$x_0/\begin{bmatrix}Up\\U_f\end{bmatrix} \cdots x_q/\begin{bmatrix}Up\\U_f\end{bmatrix} \cdots x_{j-1}/\begin{bmatrix}Up\\U_f\end{bmatrix}$
$P_0 = P_u^d - P^d$
W_p
u_0 $\vdots$ u_{i-1} y_0 $\vdots$ y_{i-1}
u_q $\vdots$ u_{i+q-1} y_q $\vdots$ y_{i+q-1}
u_{j-1} $\vdots$ u_{i+j-2} y_{j-1} $\vdots$ y_{i+j-2}
Kalman Filter
$\widehat{X}_i$
$\hat{x}_i \cdots \hat{x}_{i+q} \cdots \hat{x}_{i+j-1}$

Figure 3.4: Interpretation of $\widehat{X}_i$ as a sequence of non-steady state Kalman filter state estimates based upon i input-output observation pairs u_t, y_t. The Kalman filter is initialized with $\widehat{X}_0 = X_u^d$ and $P_0 = P_u^d - P^d$.

Defining $\widetilde{X}_i = \widehat{X}_{i[X_p^d/_{U_f}U_p, P_u^d - P^d]}$ and recalling the fact that $Y_f/(W_p/U_f^\perp) = \left(Y_f/_{U_f}W_p\right)/U_f^\perp$ we obtain

$$Y_f/_{U_f}W_p \underset{j\to\infty}{=} \Gamma_i \widetilde{X}_i, \tag{3.39}$$

$$Y_f/(W_p/U_f^\perp) \underset{j\to\infty}{=} \Gamma_i \widetilde{X}_i/U_f^\perp. \tag{3.40}$$

This leads us to the following theorem, unifying a large set of existing subspace identification algorithms in one single framework by introducing two weighting matrices W_1 and W_2.

Theorem 3.4. *Under the assumptions that:*

1. *The deterministic input u_t is uncorrelated with the process noise w_t and the measurement noise v_t.*
2. *The input u_t is persistently exciting of order $2i$.*
3. *The number of measurements goes to infinity $j \to \infty$.*
4. *The process noise w_t and the measurement noise v_t are not identically zero.*
5. *Two user defined weighting matrices $W_1 \in \mathbb{R}^{li\times li}$ and $W_2 \in \mathbb{R}^{j\times j}$ are such that W_1 is of full rank and W_2 obeys: rank$(W_p) =$ rank(W_pW_2).*

And with $\mathcal{O}_i$ defined as the oblique projection:

$$\mathcal{O}_i \triangleq Y_f /_{U_f} W_p, \tag{3.41}$$

and the singular value decomposition:

$$W_1 \mathcal{O}_i W_2 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 S_1 V_1^T, \tag{3.42}$$

we have:

1. *The matrix $\mathcal{O}_i$ is given as $\mathcal{O}_i = \Gamma_i \widetilde{X}_i$ with $\widetilde{X}_i = \widehat{X}_{i[X_p^d/_{U_f}U_p, P_u^d - P^d]}$.*

2. *The order of the system (3.1) is equal to the number of singular values in equation (3.42) different from zero.*

3. *The extended observability matrix Γ_i is equal to:*

$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2} T,$$

   with T a similarity transformation to indicate that the system is only determined up to a change in basis (see equation (3.4)).

4. *The state $\widetilde{X}_i$ is equal to:*

$$\widetilde{X}_i = \Gamma_i^{\dagger} \mathcal{O}_i$$

Proof. Trivial from the discussion in Subsections 3.5.3 and 3.5.4. ☐

In (3.42) it is assumed that for $j \to \infty$, S_1 contains the non-zero singular values. Hence, for $j \to \infty$, the minimal order of the system can be determined from the number of singular values different from zero. In practical cases, and for finite data-samples, the smallest singular values will in general not be zero. In this case, the order is estimated by looking for a 'gap' in the singular value spectrum. The resulting low rank decomposition is in this case not exact and can be shown to correspond to a form of frequency balanced model reduction in the sense of Enns [145, 147]. Hereby, the weighting matrices W_1 and W_2 in the SVD determine the particular frequency weighting that is used and thereby have an impact on the state space basis in which the final model is returned. A full analysis of the impact of the choice of W_1 and W_2 is outside the scope of this thesis, and we refer the interested reader to [145, 147]. Nevertheless, some particular choices for W_1 and W_2 deserve attention as they allow us to fit most subspace identification algorithms in one framework. This is not only true for the N4SID and the PO-MOESP algorithm (see (3.39) and (3.40)) but it can also be shown that the CVA method can be included in this framework [146, 147]. Some weights W_1 and W_2 which are needed to obtain different kinds of subspace algorithms are displayed in Table 3.1.

Method	W_1	W_2
N4SID	I_{il}	I_j
MOESP	I_{il}	$\Pi_{U_f^\perp}$
CVA	$\left(\frac{1}{j} Y_f/U_f^\perp \left(Y_f/U_f^\perp\right)^T\right)^{-1/2}$	$\Pi_{U_f^\perp}$

Table 3.1: Different combined stochastic-deterministic subspace identification algorithms and their equivalent weighting matrices in the unifying Theorem 3.4.

3.5.6 Extracting the system matrices using the extended observability matrix

Up till now, the discussion largely focused on obtaining an estimate for the extended observability matrix and the Kalman filter state sequence $\widetilde{X}_i$ based on input-output data. The problem that remains to be solved is that of finding the state-space matrices A, B, C, D, Q, R, S from Γ_i and/or $\widetilde{X}_i$. Two classes of solutions for this problem exist. A first class extracts the matrices A and C from Γ_i, as it is done in realization theory (see Sections 3.2 and 3.3). Estimates for B and D are subsequently obtained by going back to the data and solving a least-squares problem. This class of solutions is mostly found in the literature on MOESP type of algorithms [152–155] but can equally well be used in other algorithms. It is further discussed in the text below. A second class of solutions extracts A, B, C and D using the obtained system states. This class of solutions is mostly found in the literature on N4SID and CVA and is further discussed in Subsection 3.5.7.

As mentioned above, in the first class of solutions, A and C are calculated from Γ_i after which B and D can be found by solving a linear least squares problem in the original data block-Hankel matrices. The advantage of this type of approaches is that the procedure is very transparent, as will be shown shortly. The disadvantage is that the second step, the estimation of B and D takes a considerable amount of time, as a rather large least-squares problem needs to be solved.

Determination of A and C

The matrix C can easily be extracted as the first l rows of Γ_i. For the matrix A, we use the following shift invariance property [92]:

$$\underline{\Gamma_i}A = \overline{\Gamma_i},$$

where $\underline{\Gamma_i}$ and $\overline{\Gamma_i}$ symbolize Γ_i with respectively the last l and first l rows removed. This is exactly the same approach as used in the derivation of deterministic and stochastic realization in Sections 3.2 and 3.3.

Determination of B and D

Many approaches exist to estimate B and D once A and C are known, and a full overview would be beyond the scope of this thesis. It suffices to say that once A and C are known, the transfer function of the deterministic system:

$$H(z) = C(zI - A)^{-1}B + D,$$

is linear in the matrices B and D. Most existing approaches exploit this fact and estimate B and D using a least squares algorithm in available input-output data. We introduce one possible approach which will be used in this thesis, and refer to [147] for further reading.

- **Simulation error** A simulated output $\hat{y}_t$ can be determined as:

$$\begin{aligned}\hat{y}_t &= Du_t + \sum_{r=0}^{t-1} CA^{t-r-1}Bu_r \\ &= [u_t^T \otimes I_l] \cdot \text{vec}(D) + \left(\sum_{r=0}^{t-1} u_r^T \otimes CA^{t-r-1}\right) \cdot \text{vec}(B).\end{aligned}$$

  With this last equation, B and D can be found by minimizing the following criterion:

$$\sum_{t=0}^{s} \left[y_t - [u_t^T \otimes I_l] \cdot \text{vec}(D) - \left(\sum_{r=0}^{t-1} u_r^T \otimes CA^{t-r-1}\right) \cdot \text{vec}(B) \right]^2,$$

  which is linear in vec(D) and vec(B).

Determination of Q, R and S

The determination of Q, R and S has never been the main concern in the original papers about MOESP-like approaches such as the ones presented in [152,154,155]. In contrast to the state based approaches which will be outlined in 3.5.7, no commonly accepted techniques to estimate Q, R and S are therefore available. Nevertheless, once the system matrices A, B, C and D are obtained several approaches exist to obtain estimates for Q, R and S. we discuss two possibilities:

- **Subtracting the deterministic outputs:** Once A, B, C and D are known, the stochastic part of the output can easily be removed as:

$$y_t^s = y_t - \hat{y}_t^d, \ \ \forall t, \tag{3.43}$$

  with $\hat{y}_t^d$ an estimate for the deterministic output following from the identified deterministic model. A so-called stochastic identification algorithm (see also Section 3.6) can thereafter be performed on the outputs y_t^s to determine the properties of the noise-model. One possibility is to estimate G from (3.10) using Γ_i and the output covariance matrices $\Lambda_i = E\left\{y_{t+i}^T y_t\right\}$. Once G is known, Q, R and S can be obtained using the equations (3.14-3.15).

- **Estimating the state:** Once A, B, C and D are known, we can obtain an estimate for the system states and the process and measurement noise sequences $\rho_w = \begin{bmatrix} w_i & \dots & w_{i+j-1} \end{bmatrix}$ and $\rho_v = \begin{bmatrix} v_i & \dots & v_{i+j-1} \end{bmatrix}$ from:

  $$\begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \widetilde{X}_i \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} \widehat{\rho}_w \\ \widehat{\rho}_v \end{bmatrix}, \tag{3.44}$$

  which is linear in the unknown states. An estimate for the noise model is now obtained as:

  $$\begin{bmatrix} \widehat{Q} & \widehat{R} \\ \widehat{R}^T & \widehat{S} \end{bmatrix} = \frac{1}{j} \begin{bmatrix} \widehat{\rho}_w \widehat{\rho}_w^T & \widehat{\rho}_w \widehat{\rho}_v^T \\ \widehat{\rho}_v \widehat{\rho}_w^T & \widehat{\rho}_v \widehat{\rho}_v^T \end{bmatrix}. \tag{3.45}$$

  The obtained states from (3.44) will converge to the true states for $i \to \infty$. For finite i, the obtained states, and hence the noise model will in general not be consistently estimated using this procedure.

A practical algorithm

A practical identification algorithm using the extended observability matrix is found in Figure 3.5. This algorithm will be used in many examples and derivations in this thesis.

3.5.7 Extracting the system matrices using the states

In Subsections 3.5.3, and 3.5.4 it was observed that

$$Z_i = Y_f / \begin{bmatrix} U_f \\ W_p \end{bmatrix} = \Gamma_i \widehat{X}_{i_{[X_u^d, P_u^d - P^d]}} + H_i U_f. \tag{3.46}$$

We introduce $U_f^- = U_{i+1|2i-1}$, $Y_f^- = Y_{i+1|2i-1}$, $U_p^+ = U_{0|i}$, $Y_p^+ = Y_{0|i}$ and $W_p^+ = \begin{bmatrix} U_p^+ \\ Y_p^+ \end{bmatrix}$, which are essentially the same as U_f, Y_f, U_p, Y_p and W_p, except for the fact that the border between "past" and "future" was shifted one instance in time. With these definitions, it can easily be proven that (see [144, 147])

$$Z_{i+1} = Y_f^- / \begin{bmatrix} U_f^- \\ W_p^+ \end{bmatrix} = \Gamma_{i-1} \widehat{X}_{i+1_{[X_u^d, P_u^d - P^d]}} + H_{i-1} U_f^-. \tag{3.47}$$

Hence, the state sequence $\widehat{X}_{i+1}$ is the result of a set of Kalman filters working in parallel on the columns of W_p^+, with the same initial conditions as the Kalman filter that generates $\widehat{X}_i$. In other words, $\widehat{X}_{i+1}$ is the result of just one further iteration of the Kalman filter procedure that leads to $\widehat{X}_i$. This situation is graphically depicted in Figure 3.6. Removing the components along U_f from (3.46) and (3.47), it is now obvious that:

$$\mathcal{O}_i \quad = \quad Y_f /_{U_f} W_p = \Gamma_i \widehat{X}_{i_{[X_p^d /_{U_f} U_p, P_u^d - P^d]}} = \Gamma_i \widetilde{X}_i, \tag{3.48}$$

A practical algorithm using Γ_i

1. Calculate the oblique projection:

$$\mathcal{O}_i = Y_f /_{U_f} W_p.$$

2. Calculate the SVD of the weighted oblique projection:

$$W_1 \mathcal{O}_i W_2 = U S V^T.$$

3. Determine the order by inspecting the singular values in S and partition the SVD accordingly to obtain U_1, U_2 and S_1.
4. Determine Γ_i and $\Gamma_i^\perp$ as:

$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2}, \qquad \Gamma_i^\perp = U_2^T W_1.$$

5. Determine A from Γ_i as $A = \underline{\Gamma_i}^\dagger \overline{\Gamma_i}$.
6. Estimate B and D from

$$\begin{aligned}(\widehat{B}, \widehat{D}) &= \arg\min_{B,D} \sum_{t=0}^{s} \Big[y_t - [u_t^T \otimes I_l] \cdot \mathrm{vec}(D) \\ &- \left(\sum_{r=0}^{t-1} u_r^T \otimes C A^{t-r-1} \right) \cdot \mathrm{vec}(B) \Big]^2 ,\end{aligned}$$

7. Subtract the estimated deterministic output

$$y_t^s = y_t - \hat{y}_t^d,$$

and perform stochastic identification on the stochastic output sequence y_t^s.

Figure 3.5: A practical subspace identification algorithm using the extended observability matrix.

$$\mathcal{O}_{i+1} \quad = \quad Y_f^- / _{U_f^-} W_p^+ = \Gamma_i \widehat{X}_{i_{[X_p^d / _{U_f^-} U_p^+ , P_u^d - P^d]}} = \Gamma_i \widetilde{X}_{i+1}. \tag{3.49}$$

Note that based on the two state sequences $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$, estimates for the system matrices can be obtained as follows:

$$\begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \widetilde{X}_i \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} \widehat{\rho}_w \\ \widehat{\rho}_v \end{bmatrix}, \tag{3.50}$$

which can be solved in a least-squares sense for A, B, C and D.

Care should be taken about the fact that both state sequences $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ should be calculated in the same state-space basis. By simply calculating $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ from the SVD of $\mathcal{O}_i$ and $\mathcal{O}_{i+1}$, this is not necessarily the case. However, $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ can be forced into the same state-space by first determining Γ_i and $\widetilde{X}_i$ from (3.48) and calculating $\widetilde{X}_{i+1}$ as:

$$\widetilde{X}_{i+1} = \underline{\Gamma_i}^{\dagger} \mathcal{O}_{i+1}.$$

Another remaining problem, as can clearly be observed from (3.48) and (3.49), is that the obtained state sequence estimates $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ are the results of two separate Kalman filters with initial conditions $X_p^d / _{U_f} U_p$ and $X_p^d / _{U_f^-} U_p^+$ respectively. This has some serious consequences, as the set of equations (3.50) are not entirely consistent due to the different initial conditions [144, 147]. However, it can be proven [144, 147] that A, B, C and D, are consistently estimated if at least one of the following conditions is satisfied:

- $i \to \infty$,
- The system is purely deterministic, i.e. $v_t = w_t = 0$, $\forall t$,
- The deterministic input u_t is white noise.

If none of the above conditions is satisfied, one obtains biased estimates for A, B, C and D using the algorithm described above.

Once A, B, C and D are obtained, estimates for Q, R and S can trivially be obtained from the estimated residuals of (3.50) as in (3.45).

Remark on unbiased state-based algorithms

Unlike the method described above, state-based algorithms that do return consistent estimates exist. However, since these algorithms are quite involved and will not be used in this thesis for reasons of clarity, we refer to [144, 147] for further reading on this topic. It suffices to say that a particular advantage of state-based algorithms is that the noise-model is straightforwardly obtained, without the need for a second stochastic identification step as is the case in algorithms based on the extended observability matrix. Hence, even though the unbiased state-based algorithms presented in [144, 147] are theoretically quite involved, in practical applications they do have their advantages over the approaches so far discussed. If one is only interested in the deterministic model, an approach based on the extended observability matrix is probably preferable.

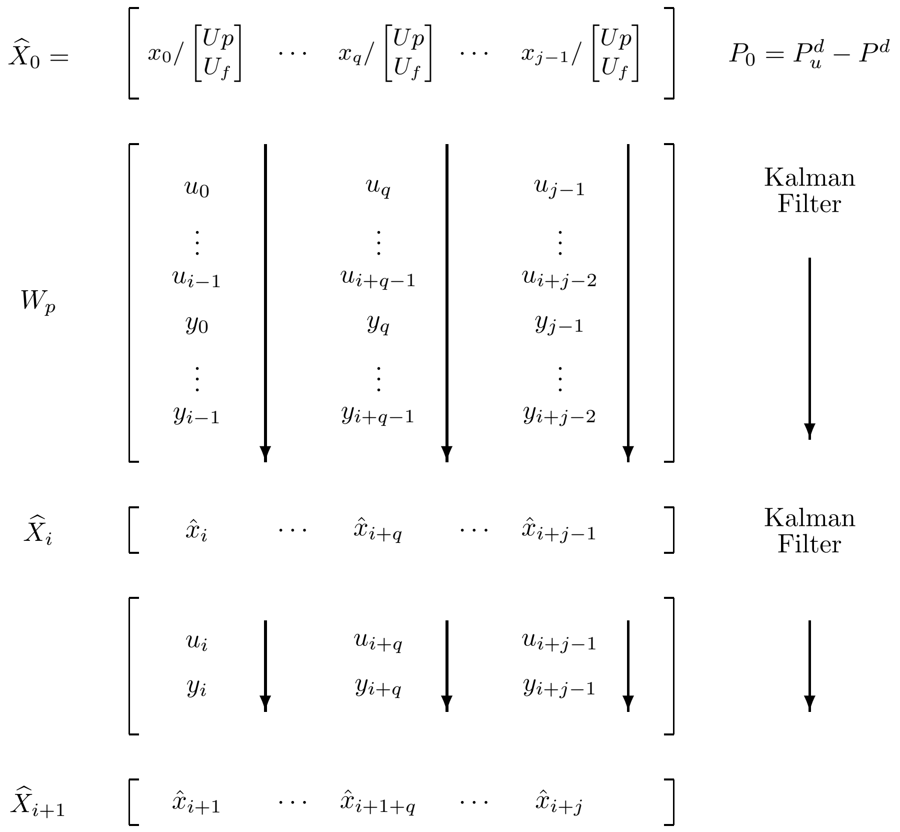

$\widehat{X}_0 =$
$x_0 / \begin{bmatrix} Up \\ U_f \end{bmatrix} \cdots x_q / \begin{bmatrix} Up \\ U_f \end{bmatrix} \cdots x_{j-1} / \begin{bmatrix} Up \\ U_f \end{bmatrix}$
$P_0 = P_u^d - P^d$
W_p
$u_0 \; \vdots \; u_{i-1} \; y_0 \; \vdots \; y_{i-1}$
$u_q \; \vdots \; u_{i+q-1} \; y_q \; \vdots \; y_{i+q-1}$
$u_{j-1} \; \vdots \; u_{i+j-2} \; y_{j-1} \; \vdots \; y_{i+j-2}$
Kalman Filter
$\widehat{X}_i$
$\hat{x}_i \cdots \hat{x}_{i+q} \cdots \hat{x}_{i+j-1}$
Kalman Filter
$u_i \; y_i$
$u_{i+q} \; y_{i+q}$
$u_{i+j-1} \; y_{i+j-1}$
$\widehat{X}_{i+1}$
$\hat{x}_{i+1} \cdots \hat{x}_{i+1+q} \cdots \hat{x}_{i+j}$

Figure 3.6: Interpretation of $\widehat{X}_i$ and $\widehat{X}_{i+1}$ as two consecutive estimates from a non-steady state Kalman filter state estimates based upon $i+1$ input-output observation pairs u_t, y_t. The Kalman filter is initialized with $\widehat{X}_0 = X_u^d$ and $P_0 = P_u^d - P^d$.

A practical algorithm

A practical identification algorithm using the states is found in Figure 3.7. This algorithm will be used as a basis for the derivation of Hammerstein identification methods in Chapter 8.

3.6 Stochastic subspace identification

3.6.1 Problem definition

The term stochastic subspace identification is used to describe the process of using subspace identification to obtain a stochastic model if no known input measurements are available. In that, the aim of stochastic subspace

A practical algorithm using the states (biased)

1. Calculate the oblique projections:

$$\begin{aligned}\mathcal{O}_i &= Y_f /_{U_f} W_p, \\ \mathcal{O}_{i+1} &= Y_f^- /_{U_f^-} W_p^+.\end{aligned}$$

2. Calculate the SVD of the weighted oblique projection:

$$W_1 \mathcal{O}_i W_2 = U S V^T.$$

3. Determine the order by inspecting the singular values in S and partition the SVD accordingly to obtain U_1 and S_1.
4. Determine Γ_i as:

$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2}.$$

5. Determine the state sequences:

$$\begin{aligned}\widetilde{X}_i &= \Gamma_i^\dagger \mathcal{O}_i \\ \widetilde{X}_{i+1} &= \underline{\Gamma_i}^\dagger \mathcal{O}_{i+1}.\end{aligned}$$

6. Solve the set of linear equations for A, B, C and D:

$$\begin{bmatrix}\widetilde{X}_{i+1} \\ Y_{i|i}\end{bmatrix} = \begin{bmatrix}A & B \\ C & D\end{bmatrix}\begin{bmatrix}\widetilde{X}_i \\ U_{i|i}\end{bmatrix} + \begin{bmatrix}\rho_w \\ \rho_v\end{bmatrix},$$

7. Determine Q, R and S from the residuals as

$$\begin{bmatrix}Q & R \\ R^T & S\end{bmatrix} = \frac{1}{j}\begin{bmatrix}\rho_w \rho_w^T & \rho_w \rho_v^T \\ \rho_v \rho_w^T & \rho_v \rho_v^T\end{bmatrix}.$$

Figure 3.7: A practical subspace identification algorithm using the states. For general systems, the algorithm returns biased estimates for A, B, C and D. Unbiased identification algorithms using the states exist and are described in [144, 147].

identification is the same as that of stochastic realization theory. The only difference is that stochastic subspace algorithms start from data-measurements in structured block Hankel matrices, while stochastic realization algorithms start from estimates of the output covariance matrices. Despite this difference, in this section it will be seen that most existing stochastic realization algorithms can easily be understood in the stochastic subspace identification framework.

The problem considered in stochastic subspace identification is the following: given a system of the form

$$\begin{aligned} x_{t+1} &= Ax_t + w_t, \\ y_t &= Cx_t + v_t, \end{aligned}$$

with w_t and v_t white noise sequences with second order moments

$$E\left\{ \begin{bmatrix} w_t \\ v_t \end{bmatrix} \begin{bmatrix} w_k^T & v_k^T \end{bmatrix} \right\} = \begin{bmatrix} Q & R \\ R^T & S \end{bmatrix} \delta_{tk},$$

and given a set of output measurements $\{y_t\}, t = 0, \ldots, N-1$, find the minimal system order n, the system matrices A and C up to within a similarity transformation, and the noise covariance matrices Q, R and S.

3.6.2 A unifying framework

A stochastic subspace identification algorithm can directly be obtained by setting the inputs to zero in the analysis of the combined stochastic-deterministic case. Doing this, the oblique projection $Y_f/_{U_f}W_p$ reduces to an orthogonal projection Y_f/Y_p. It can be shown that with a proper weighting of this projection, most existing stochastic realization algorithms can be treated in a subspace framework. This observation is summarized in the following theorem [144, 147]:

Theorem 3.5. *Under the assumptions that:*

1. *The process noise w_t and the measurement noise v_t are not identically zero.*
2. *The number of measurements goes to infinity $j \to \infty$.*
3. *Two user defined weighting matrices $W_1 \in \mathbb{R}^{li \times li}$ and $W_2 \in \mathbb{R}^{j \times j}$ are such that W_1 is of full rank and W_2 obeys: rank$(Y_p) =$ rank$(Y_p W_2)$.*

And with $\mathcal{O}_i$ defined as the orthogonal projection:

$$\mathcal{O}_i \triangleq Y_f/Y_p, \tag{3.51}$$

and the singular value decomposition:

$$W_1 \mathcal{O}_i W_2 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 S_1 V_1^T, \tag{3.52}$$

we have:

1. *The matrix* $\mathcal{O}_i$ *is given as* $\mathcal{O}_i = \Gamma_i \widetilde{X}_i$ *with* $\widetilde{X}_i = \widehat{X}_{i[0,0]}$ *the result of a non-steady state Kalman filter, initialized at* 0, *and applied in parallel to the columns of* Y_p.

2. *The order of the system is equal to the number of singular values in equation (3.52) different from zero.*

3. *The extended observability matrix* Γ_i, *and the associated extended controllability matrix of the covariance model* $\Delta_i^c = \begin{bmatrix} A^{i-1}G & A^{i-2}G & \dots & G \end{bmatrix}$ *are equal to (up to a change in basis):*

$$\begin{aligned}\Gamma_i &= W_1^{-1} U_1 S_1^{1/2} T, \\ \Delta_i^c &= \frac{1}{j} \Gamma_i^{\dagger} Y_f Y_p^T\end{aligned}$$

with T *a similarity transformation to indicate that the system is only determined up to a change in basis (see equation (3.4)).*

4. *A realization* $\widetilde{X}_i$ *for the state sequence is given as:*

$$\widetilde{X}_i = \Gamma_i^{\dagger} \mathcal{O}_i .$$

Proof. The proof follows almost immediately by setting the inputs to zero in Theorem 3.4. The expression for the inverted extended observability matrix Δ_i^c follows from (3.17). For a detailed proof of the theorem, we refer the reader to [143, 147]. □

3.6.3 Relation to existing stochastic subspace identification algorithms

Stochastic subspace identification is very closely related to stochastic realization. This is apparent from $Y_f / Y_p = Y_f Y_p^T (Y_p Y_p^T)^{-1} = \Gamma_i \Delta_i^c (\frac{1}{j} Y_p Y_p^T)^{-1}$. Hence for $j \to \infty$ and Y_p of full rank, the extended observability matrix returned by stochastic realization and the extended observability matrix returned by stochastic subspace identification are the same. By properly choosing the weights W_1 and W_2 in Theorem 3.5, several variants of the basic stochastic realization algorithm can be recovered, such as the principal component algorithm (PC [6, 8]), the unweighted principal component algorithm (UPC [8]) and the canonical variate algorithm (CVA [3, 4, 8]). Without going into further detail about these variants (the interested reader is referred to [147] and the references therein) an overview of the appropriate weights is given in Table 3.2. Note how the weights for the CVA case correspond to the weights for the CVA case in the combined deterministic-stochastic case, with the inputs set to zero in the latter algorithm. Also note how the UPC case corresponds to N4SID, again with the inputs set to zero.

Method	W_1	W_2
PC	I_{il}	$\frac{1}{\sqrt{j}} Y_p^T (Y_p Y_p^T)^{-1/2} Y_p$
UPC	I_{il}	I_j
CVA	$\left(\frac{1}{j} Y_f Y_f^T\right)^{-1/2}$	I_j

Table 3.2: Different stochastic subspace identification algorithms and their equivalent weighting matrices in the unifying Theorem 3.5

3.6.4 Extracting the system matrices

Two ways exist to estimate the system matrices A, C, Q, R and S in stochastic subspace identification algorithms. A first possibility is to estimate a covariance model A, G, C, Λ_0, using results from Theorem 3.5. In a second step, Q, R and S can then be obtained using a similar procedure as found in stochastic realization algorithms (see 3.3). A second possibility is to obtain A, C from a least-squares algorithm in a first step and estimate Q, R and S from the residuals of this least squares problem. Both possibilities will briefly be outlined below.

Using the covariance model

As outlined in Section 3.3, the matrices A, G, C, Λ_0 can be viewed as forming a deterministic system of which the Markov parameters are the output covariance matrices Λ_t, $t \geq 0$ of the stochastic system. An estimate for Λ_0 can directly be obtained from the data as $\Lambda_0 = \frac{1}{j} Y_{i|i} Y_{i|i}^T$. G is obtained as the last l columns of Δ_i^c, where Δ_i^c following directly from Theorem 3.5. As in the combined stochastic-deterministic case, the system matrices A and C can be extracted using the shift-invariance property of Γ_i, or using the estimated state sequences $\widetilde{X}_i, \widetilde{X}_{i+1}$, where $\widetilde{X}_{i+1} = \underline{\Gamma}_i{}^{\dagger} \mathcal{O}_{i-1}$ with $\mathcal{O}_{i-1} = Y_f^- / Y_p^+$. We have

$$\begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A \\ C \end{bmatrix} \widetilde{X}_i + \begin{bmatrix} \widehat{\rho}_w \\ \widehat{\rho}_v \end{bmatrix},$$

where ρ_w and ρ_v can be shown to be uncorrelated with $\widetilde{X}_i$ [144,147]. Unlike in the combined deterministic stochastic case, the steady state Kalman filter banks for $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ have the same initialization, namely zero. Hence, in contrast to the combined case, A and C can consistently be estimated as

$$\begin{bmatrix} A \\ C \end{bmatrix} = \begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} \widetilde{X}_i^{\dagger}.$$

Consistent estimates for Q, R and S can now be obtained in a second step by solving the set of equations (3.14-3.15), similar as in the stochastic realization algorithm. It is this approach that is usually followed for the stochastic subspace identification problem.

Using least-squares residuals

As was seen above, one possibility to estimate the system matrices A and C is to use the Kalman filter state sequences $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$. In a second step estimates for Q, R and S can readily be obtained as

$$\begin{bmatrix} \widehat{Q} & \widehat{R} \\ \widehat{R}^T & \widehat{S} \end{bmatrix} = \frac{1}{j} \begin{bmatrix} \widehat{\rho}_w \widehat{\rho}_w^T & \widehat{\rho}_w \widehat{\rho}_v^T \\ \widehat{\rho}_v \widehat{\rho}_w^T & \widehat{\rho}_v \widehat{\rho}_v^T \end{bmatrix}$$

Similarly as in the combined case, these estimates will in general be biased, unless $i \rightarrow \infty$. Nevertheless this approach will be seen to be useful in some specific cases in Chapter 4.

A practical algorithm

A practical stochastic subspace identification algorithm is found in Figure 3.8.

3.7 Subspace identification for separately parameterized models

The combined stochastic-deterministic subspace identification algorithm studied in Section 3.5 identifies jointly parameterized models. As outlined in 3.5.1, this means that the dynamics of the deterministic and stochastic subsystem are considered to be equal. If this is not the case, a jointly parameterized model can always be obtained by increasing the system order (see equation (3.28)). However, increasing the system order increases the number of parameters to be estimated, and hence also the variance of the uncertainty on the estimated parameters. In recent years, a lot of attention has been drawn to this problem, and it has been shown [23, 26, 90] that, although in general, combined subspace identification approaches are still believed to outperform methods based on a separate parameterization, for some specific cases using a separately parameterized approach will lead to better results. A full analysis of this issue will be given in Chapter 5. In the following we will restrict ourselves to a basic introduction into subspace identification methods for separately parameterized models.

3.7.1 Decomposition of the system in a deterministic and a stochastic subsystem

At the heart of the separately parameterized approach is the so-called data orthogonalization step, where the future outputs Y_f are split up into an input-dependent part along the past and future inputs:

$$\mathcal{Y}_f^d = Y_f / \begin{bmatrix} U_p \\ U_f \end{bmatrix},$$

A practical stochastic subspace identification algorithm

1. Calculate the orthogonal projections:
$$\begin{aligned}\mathcal{O}_i &= Y_f/Y_p,\\ \mathcal{O}_{i+1} &= Y_f^-/Y_p^+.\end{aligned}$$
2. Calculate the SVD of the weighted orthogonal projection:
$$W_1\mathcal{O}_iW_2 = USV^T.$$
3. Determine the order by inspecting the singular values in S and partition the SVD accordingly to obtain U_1 and S_1.
4. Determine Γ_i as:
$$\Gamma_i = W_1^{-1}U_1S_1^{1/2}.$$
5. Determine the state sequences:
$$\begin{aligned}\widetilde{X}_i &= \Gamma_i^\dagger\mathcal{O}_i\\ \widetilde{X}_{i+1} &= \underline{\Gamma_i}^\dagger\mathcal{O}_{i+1}.\end{aligned}$$
6. Solve the set of linear equations for A and C:
$$\begin{bmatrix}\widetilde{X}_{i+1}\\ Y_{i|i}\end{bmatrix} = \begin{bmatrix}A\\ C\end{bmatrix}\widetilde{X}_i + \begin{bmatrix}\rho_w\\ \rho_v\end{bmatrix}.$$
7. Determine G as the last l columns of $\Delta_i^c = \frac{1}{j}\Gamma_i^\dagger Y_fY_p^T$ and Λ_0 as $\frac{1}{j}Y_{i|i}Y_{i|i}^T$.
8. Determine P from
$$P = APA^T + (G - APC^T)(\Lambda_0 - CPC^T)^{-1}(G - APC^T).$$
and Q, R and S from
$$\begin{aligned}&\Sigma = (\Lambda_0 - CPC^T)^{-1},\;\; K = (G - APC^T)\Sigma^{-1},\\ &Q = K\Sigma K^T,\;\; R = K\Sigma,\;\; S = \Sigma.\end{aligned}$$

Figure 3.8: A practical stochastic subspace identification algorithm. The algorithm returns consistent estimates for A, C, Q, R and S.

and a part uncorrelated with the inputs:

$$\mathcal{Y}_f^s = Y_f / \begin{bmatrix} U_p \\ U_f \end{bmatrix}^{\perp}.$$

Following this initial projection, an input-output subspace identification algorithm (assuming no noise) is performed on $\mathcal{Y}_f^d$, and a stochastic subspace identification algorithm is performed on $\mathcal{Y}_f^s$. Note that the no-noise assumption on the deterministic data is justified in the limit for $j \to \infty$ as all noise-sources are considered to be uncorrelated with the inputs, and therefore removed in the initial projection.

3.7.2 Identification of the deterministic subsystem using PI-MOESP

For the deterministic system, we know from Subsection 3.4.2 that an effective way to obtain an estimate for the extended observability matrix and the state is to project the future deterministic outputs onto the orthogonal complement of the inputs. The corresponding projection

$$\mathcal{O}_i = \mathcal{Y}_f^d / U_f^{\perp} = Y_f / \begin{bmatrix} U_p \\ U_f \end{bmatrix} / U_f^{\perp} = Y_f / (U_p / U_f^{\perp})$$

corresponds to the projection introduced in the PI-MOESP algorithm [152]. For the PI-MOESP algorithm, it was proven that a consistent estimate for the extended observability matrix can be obtained from the following relations

$$\begin{aligned} Y_f &= \Gamma_i X_i^d + H_i U_f + Y_f^s, \\ Y_f / (U_p / U_f^{\perp}) &\underset{j\to\infty}{=} \Gamma_i X_i^d / (U_p / U_f^{\perp}). \end{aligned}$$

Once the extended observability matrix is obtained, A^d, B, C^d and D can be calculated as in the combined stochastic deterministic algorithms. Note that in principle an algorithm based on the states is also possible. A biased version, similar to the biased state-based algorithm for the combined case has been described in [123]. However, as far as the PI-MOESP algorithm is concerned, we will limit ourselves to algorithms based on the extended observability matrix in this thesis.

3.7.3 Identification of the stochastic subsystem

The identification of the stochastic subsystem is more problematic than that of the deterministic subsystem. From

$$\begin{aligned} Y_f &= \Gamma_i X_i^d + H_i U_f + Y_f^s, \\ \mathcal{Y}_f^s = Y_f / \begin{bmatrix} U_p \\ U_f \end{bmatrix}^{\perp} &\underset{j\to\infty}{=} \Gamma_i X_i^d / \begin{bmatrix} U_p \\ U_f \end{bmatrix}^{\perp} + Y_f^s, \end{aligned}$$

it is clear that $\mathcal{Y}_f^s$ will still contain some elements of the deterministic subsystem, namely those parts of the deterministic state X_i^d which are uncorrelated with the past and future inputs. In [123], this difference between the estimated stochastic outputs $\mathcal{Y}_f^s$ and the true stochastic outputs Y_f^s is called a 'smoothing error'. Ways to deal with this error are discussed below.

- **Neglect the smoothing error:** The most obvious solution is to neglect the smoothing error. Nevertheless, it was shown in [123] that, if neglected, this error can lead to biased results, and a much higher estimated order for the stochastic model than the true one.

- **Simulate the deterministic output contribution with the model obtained using PI-MOESP:** A possibility is to use the PI-MOESP model to get estimates $\hat{y}_t^d$ for the outputs y_t^d based on available input measurements. An estimate for the stochastic outputs $\hat{y}_t^s$ is then obtained as $\hat{y}_t^s = y_t - \hat{y}_t^d$ similar as in (3.43). The sequence $\{\hat{y}^s\}$ can be used to start a stochastic identification algorithm.

- **Use a prefiltering technique to remove the smoothing error:** In [122], a prefiltering technique is described to remove the smoothing error. However, the analysis of this technique is quite involved and beyond the scope of this thesis. We refer the reader to [122] for further reading.

It should be noted at this point that the stochastic identification algorithms so far discussed do not only need the future outputs Y_f^s, but also the past outputs Y_p^s. However, for the past outputs, a decomposition by projecting on past and future inputs and their complement is even more problematic than for the future outputs. However, the data in Y_f^s can conveniently be redistributed over a new set of past and future block Hankel matrices. The length of the rows will thereby decrease from j to $j - i$. In most practical cases, $j \gg i$ and this decrease is negligible.

3.8 A practical application

To emphasize the main advantages of subspace identification algorithms for practical applications, in this section we will consider the application of subspace identification to the practical example of a glass oven. The description here is a summary of the more elaborate discussion regarding this example in [11,144].

The dataset is a part of the online database DaISy (Database for the Identification of Systems) [45]. It consists of 3 inputs (2 burners and 1 ventilator) and 6 outputs (temperatures measured at different locations). The data have been pre-processed using detrending, peak shaving, delay estimation and normalization (see [11]). A total of 1247 data-points are available of which 700 are used to identify a linear model. Two different strategies are used:

1. The biased subspace identification algorithm presented in Figure 3.7 with the number of block-rows set to $i = 10$ and the order $n = 5$.

2. A predictor error method using the system identification toolbox in matlab [100]. The predictor error method is initialized with the results of the subspace identification algorithm.

On a Pentium-IV, 2 MHz, the subspace algorithm took 240 msec to complete whereas the predictor error method needed 20.899 seconds. However, the extra time needed hardly resulted in better estimates for the predictor error method. For each output channel k in the validation data $\{y_t(k)\}$, $t = 701, \ldots, 1247$, the relative prediction error for channel k is calculated as

$$\epsilon_k = 100\sqrt{\frac{\sum_{t=701}^{1247} \left(y_t(k) - y_t^s(k)\right)^2}{\sum_{t=701}^{1247} y_t^2(k)}},$$

with y_t^s the one-step ahead prediction for the output y_t based on the obtained models. The results are displayed in Table 3.3. Note that the estimates obtained using the subspace identification algorithm are of the same quality as those of the predictor error method but, as mentioned earlier, the subspace identification algorithm is about a 1000 times faster. Furthermore, in contrast to the predictor error method, the subspace algorithm does not rely on a proper initialization. More examples of the use of subspace identification algorithms on practical datasets are for instance found in [52, 147].

	1	2	3	4	5	6
Subspace	64.74%	66.67%	65.35%	33.45%	30.04%	57.49%
PEM	71.83%	72.67%	71.07%	27.71%	30.15%	56.96%

Table 3.3: Performance of a subspace identification algorithm and a predictor error method on a dataset originating from measurements on a glass oven, and for each of the 6 outputs separately. The performance displayed is the relative prediction error as given by the formula (3.8). Note that the data has been detrended which in part explains the rather large relative prediction errors obtained.

3.9 Summary

In this chapter, an overview of subspace identification algorithms was presented. Three groups of subspace identification algorithms were discussed. The first group contains the combined stochastic-deterministic algorithms where a deterministic and a stochastic subsystem are estimated in one single orthogonal or oblique projection. The second group contains the stochastic algorithms, identifying systems without observed inputs. A third and last group contains algorithms which estimate separately parameterized systems. A schematic overview of the discussed methods, and their place in the spectrum of subspace identification techniques is given in Figure 3.9.

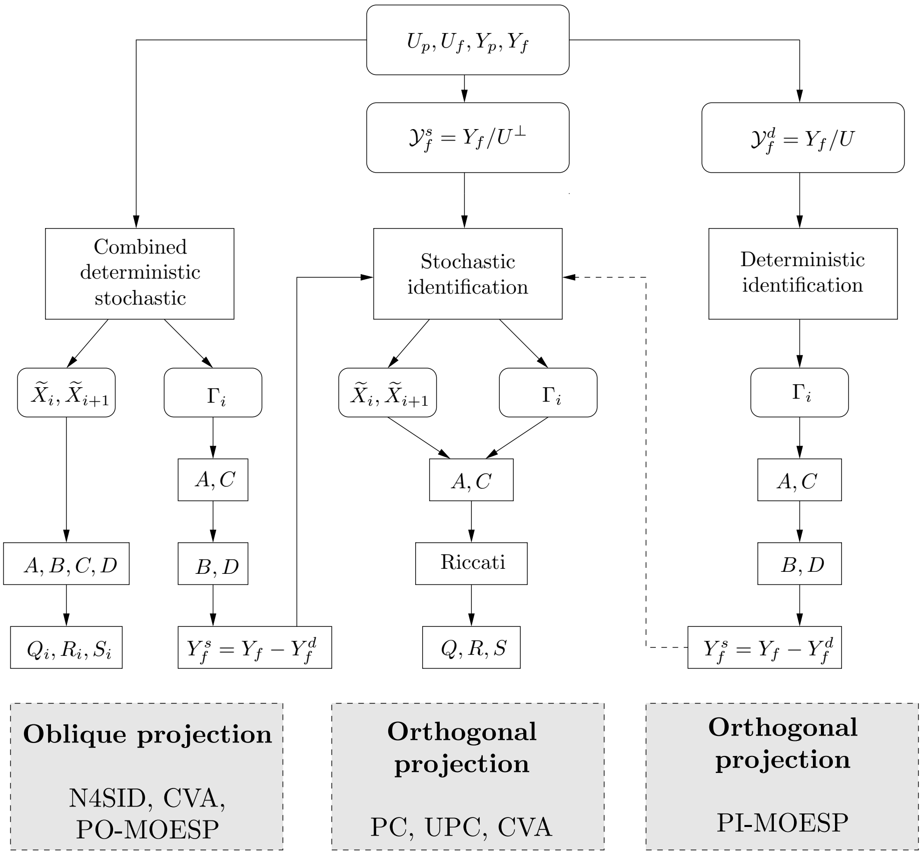


Figure 3.9: Summary of subspace identification algorithms. Subspace identification algorithms can be split into two classes, based on whether the deterministic and stochastic system are assumed to have the same dynamics (the combined stochastic-deterministic scheme for jointly parameterized models), or different dynamics (separately parameterized models). In the latter case, two identification algorithms have to be used, a deterministic one and a stochastic one. Unified frameworks exist for the combined and the stochastic methods. The symbols Q_i, R_i and S_i are used to denote that the obtained noise model is only consistent for $i \to \infty$.

Chapter 4

The positive realness problem

The most cited advantage of subspace identification techniques over classical predictor error methods is that the former only make use of numerically robust geometrical operations such as projections and the singular value decomposition. Hence, in principle a subspace identification algorithm will always produce a model provided some assumptions on the data are met. Nevertheless, in some specific cases, stochastic subspace identification algorithms are known to break down due to the so-called lack of positive realness. In this chapter, the concept of positive realness will be introduced in a realization framework. The consequences of a lack of positive realness for stochastic subspace identification algorithms will be explored, and it will be shown that positive realness can be imposed by using Tikhonov regularization.

4.1 Problem setting

In this section, the problem of positive realness will be introduced in a subspace context. It will be shown that stochastic realization and stochastic subspace identification algorithms break down if an intermediate result, the covariance model, is not positive real.

4.1.1 The covariance model

We will largely follow the same notations as used in the introduction of stochastic realization and stochastic subspace identification in Chapter 3. More specifically, we will consider stochastic systems and models of the form:

$$\begin{aligned} x_{t+1} &= Ax_t + w_t, \\ y_t &= Cx_t + v_t, \end{aligned} \tag{4.1}$$

with

$$E\left\{\begin{bmatrix} w_t \\ v_t \end{bmatrix}\begin{bmatrix} w_k^T & v_k^T \end{bmatrix}\right\} = \begin{bmatrix} Q & R \\ R^T & S \end{bmatrix}\delta_{tk}.$$

Denoting the output covariance matrices as $\Lambda_k = E\left\{y_{t+k}y_t^T\right\}$, and the cross-covariance matrix between the states and the observations as $G = E\left\{x_{t+1}y_t^T\right\}$, we have that

$$\Lambda_k = CA^{k-1}G, \;\; \Lambda_{-k} = \Lambda_k^T, \;\; k \geq 1. \tag{4.2}$$

As mentioned in the discussion on stochastic realization in Section 3.3, the output covariances can be considered as Markov parameters of a deterministic linear time invariant system with system matrices (A, G, C, Λ_0). Throughout this chapter, we will refer to (A, G, C, Λ_0) as the 'covariance model'. In stochastic realization and stochastic subspace identification, the covariance model is estimated from data, after which the following algebraic Riccati equation is solved for P (see Section 3.3):

$$P = APA^T + (G - APC^T)(\Lambda_0 - CPC^T)^{-1}(G - APC^T)^T. \tag{4.3}$$

Estimates for the noise covariance matrices Q, R and S are then obtained through

$$\begin{aligned} \Sigma &= \Lambda_0 - CPC^T, \;\; K = (G - APC^T)\Sigma^{-1}, \\ Q &= K\Sigma K^T, \;\; R = K\Sigma, \;\; S = \Sigma. \end{aligned}$$

4.1.2 Positive realness of a covariance model

While any stochastic system of the form (4.1) can be described by a covariance model (A, G, C, Λ_0), it was shown by Faurre et al. [50] that the opposite is only true under some rather stringent conditions. These are expressed by the positive real lemma [49,50], which states that a covariance model (A, G, C, Λ_0) describes a stochastic process if and only if the following matrix inequality is satisfied for at least one positive definite matrix $P = P^T > 0$:

$$\left[\begin{array}{cc} Q & R \\ R^T & S \end{array}\right] = \left[\begin{array}{cc} P & G \\ G^T & D + D^T \end{array}\right] - \left[\begin{array}{cc} APA^T & APC^T \\ CPA^T & CPC^T \end{array}\right] \geq 0. \tag{4.4}$$

In such cases, the covariance sequence (4.2) is called positive and the model (A, G, C, Λ_0) positive real. Notice that a positive real model needs to be stable to satisfy the Lyapunov equation in the upper left block of (4.4). The positivity condition can be expressed in many forms. It can be shown that, among others, the following equivalences hold, provided that A has no eigenvalues outside the unit-circle [50]:

- The covariance sequence (4.2) is positive.
- The spectral density matrix $\Phi(z)$ of the system (4.1) is positive semi-definite for all z on the unit circle: $S_z(z) + S_z^T\left(z^{-1}\right) \geq 0$ for $z = e^{j\omega}$, where $S_z(z) = \frac{\Lambda_0}{2} + C(zI_n - A)^{-1}G$.

- The spectral density $S_z(z) + S_z^T\left(z^{-1}\right)$ can be factorized as $S_z(z) + S_z^T\left(z^{-1}\right) = H(z)\Sigma H^T\left(z^{-1}\right)$, with $H(z)$ the transfer function of the forward innovation model $(\widehat{A}, \widehat{K}, \widehat{C}, I_l)$. This is the so-called spectral factorization.

- The algebraic Riccati equation (4.3) has a positive definite solution P which is the minimal solution of (4.4) and $\Lambda_0 - CPC^T \geq 0$.

Note from the last equivalence that if a covariance model is not positive real, no positive definite solution for the Riccati equation (4.3) exists and hence no physically meaningful noise matrices Q, R and S can be obtained. Since the covariance models identified as an intermediate step in stochastic realization and stochastic subspace identification are estimated using a finite amount of data, they are subject to various modeling errors. Hence, there is no immediate guarantee that these intermediate covariance models will be positive real. It is shown in [33] that this happens rather frequently in stochastic subspace identification algorithms, causing the breakdown of the algorithm and leaving the user without a noise model. In this chapter, we will explore several methods to impose positive realness on a covariance model obtained as an intermediate step in stochastic subspace identification algorithms. As for the stochastic realization algorithm we can state that it fits in the unified stochastic subspace framework (see Subsection 3.6.2), and is therefore implicitly treated in this text.

4.1.3 Causes of positive realness problems

The problem of positive realness may appear in practical applications. Even when the true system is a valid linear stochastic system, its spectral density $S_z(z) + S_z^T\left(z^{-1}\right)$ may have eigenvalues that are near zero at some points. The modeled spectral density risks being negative for these points due to various reasons, which is of course physically impossible. The covariance model $\widehat{A}, \widehat{G}, \widehat{C}, \widehat{\Lambda}_0$, for example, is built on a finite number of observed covariances $\{\widehat{\Lambda}_k\}_{k=0}^{2i-1}$. Even if these were exact ($j \to \infty$), the realization algorithm does not ensure that the infinite covariance sequence $\{\widetilde{\Lambda}_k\}_{k=0}^{\infty} = \widehat{C}\widehat{A}^{k-1}\widehat{G}$ derived from the covariance model, is positive. The smaller the number of initial covariances used to estimate the covariance model, the more likely this problem is to occur. Hence the choice of i has a direct influence on the possible occurrence of positivity problems [96, 111]. Secondly, for j finite, the observed covariances are subject to statistical errors that may increase the probability for positive realness problems to occur. Finally the ability of $(\widehat{A}, \widehat{G}, \widehat{C}, \widehat{\Lambda}_0)$ to model the observed covariance sequence is clearly dependent on the choice of the model order n. The influence of the parameters i, j and n will be illustrated by means of some examples in Section 4.4. For a further theoretical description, the reader is referred to [96].

4.2 Classical approaches to solve the positive realness problem

Few remedies are given for the cases where the procedure for obtaining a physically meaningful noise model breaks down due to positivity problems. In most cases, the covariance model has to be discarded and a new model must somehow be obtained (e.g. by using a different model order or by changing the dimensions of the block Hankel matrix (i and j)). In order to avoid such remodeling some proposals have been made by various authors [101, 120, 139, 147]. Usually they consist of algorithms to alter one or more matrices out of the identified set $\widehat{A}, \widehat{G}, \widehat{C}, \widehat{\Lambda}_0$ in order to make the covariance model positive real. A drawback is that in many cases one ends up with a biased model. Furthermore, most proposed algorithms only work if the identified system matrix $\widehat{A}$ is stable, a condition that is not necessarily satisfied for high order models identified on a finite amount of data [142]. In the following, we will briefly review existing approaches to solve the positive realness problem. In Section 4.3, a new method based on Tikhonov regularization will be proposed. In Section 4.4 it will be shown that the newly proposed method outperforms the existing approaches on a set of examples.

4.2.1 Altering $\widehat{\Lambda}_0$

If the estimated covariance model is stable, the most obvious way to impose positive realness is to lift the spectral density

$$\widehat{\Phi}(z) = \widehat{\Lambda}_0 + \widehat{C}(zI_n - \widehat{A})^{-1}\widehat{G} + \widehat{G}^T(z^{-1}I_n - \widehat{A}^T)^{-1}\widehat{C}^T$$

by replacing $\widehat{\Lambda}_0$ with $\widetilde{\Lambda}_0 = \widehat{\Lambda}_0 + cI_l$, $c > 0$ such that the spectral density is positive for all z on the unit circle. This method, was proposed by Peternell in [120]. It is obvious that sufficiently increasing $\widehat{\Lambda}_0$ will always result in a positive real covariance model $\widehat{A}, \widehat{G}, \widehat{C}, \widetilde{\Lambda}_0$. Several ways exist to estimate the parameter c. The most obvious approaches are a grid search with a progressively refined grid or a bisection algorithm [7]. In every iteration, positive realness can be evaluated by solving the Riccati equation (4.3) and checking whether $P > 0$. The biggest advantage of this technique is its simplicity and intuitiveness. The biggest disadvantage is that, as stated above, changing $\widehat{\Lambda}_0$ amounts to lifting the entire spectral density, which is quite a drastic measure. Furthermore, since $\widehat{\Lambda}_0$ is obtained from data in a very direct way, one would expect it to be more reliably estimated than $\widehat{A}$,$\widehat{G}$ and $\widehat{C}$. Hence, it seems more logical to adapt $\widehat{A}$,$\widehat{G}$ and $\widehat{C}$ in stead of $\widehat{\Lambda}_0$. Another disadvantage is found in the fact that the approach presented here can only be applied to stable models.

Example 4.1 An output sequence was created by filtering a zero mean white Gaussian noise sequence with length 1000 through the following system:

$$H(z) = \frac{(z - 0.99e^{\pm 2j})(z - 0.98e^{\pm 1.4j})(z - 0.99e^{\pm 0.6j})(z - 0.9)(z + 0.9)}{(z - 0.8e^{\pm 2.1j})(z - 0.8e^{\pm j})(z - 0.8e^{\pm 1.7j})(z - 0.8e^{0.8j})}$$

The stochastic subspace identification algorithm presented in Figure 3.8 was thereafter used to generate a covariance model $\widehat{A}, \widehat{G}, \widehat{C}, \widehat{\Lambda}_0$. The spectral density of the obtained covariance model is plotted as a solid line in Figure 4.1. Note that the spectral density is negative near the arrow in the figure, reaching a minimum of -0.3494. As a result, the Riccatti equation in step 8 of the algorithm in Figure 3.8 can not be solved for P. If $\widehat{\Lambda}_0$ is increased by 0.3494 the spectral density is positive (dashed line in the Figure), and the forward innovation model can be obtained. A more detailed analysis of this system including a full Monte-Carlo analysis and plots for the forward innovation model is given in Section 4.5.

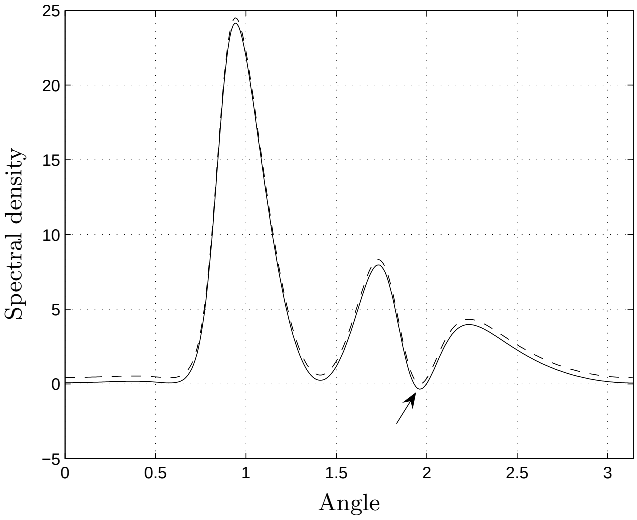


Figure 4.1: The idea of the method presented by Peternell and explored in Subsection 4.2.1 is to increase $\widehat{\Lambda}_0$ so that the spectral density $\widehat{\Phi}(z)$ is positive for all z on the unit circle. The spectral density of Example 4.1 is shown in the figure (solid line) and clearly negative in the neighborhood of the arrow in the figure. The lifted spectral density (dashed line) is positive over the entire frequency range. In Subsection 4.2.2 an improved version of this algorithm is discussed which changes $\widehat{G}$ rather than $\widehat{\Lambda}_0$ but has largely the same effect on the spectral density. A full Monte-Carlo analysis on the system in Example 4.1 is given in Section 4.5.

4.2.2 Altering $\widehat{G}$

A slightly adapted version of the former approach is proposed in [139]. Again, a matrix $\widetilde{\Lambda}_0 = \widehat{\Lambda}_0 + cI_l$, $c > 0$ is sought to make the spectral density positive for all z on the unit circle. However, in a second step the change cI_l to $\widehat{\Lambda}_0$ is transferred to the matrix $\widehat{G}$ and the original $\widehat{\Lambda}_0$ is used in the covariance model. Practically, the method amounts to finding a solution $\widehat{P}$ to the Riccati equation

$$\widehat{P} = \widehat{A}\widehat{P}\widehat{A}^T + (\widehat{G} - \widehat{A}\widehat{P}\widehat{C}^T)(\widetilde{\Lambda}_0 - \widehat{C}\widehat{P}\widehat{C}^T)^{-1}(\widehat{G} - \widehat{A}\widehat{P}\widehat{C}^T)^T,$$

where-after a $\widetilde{G}$ is sought such that

$$\widehat{P} = \widehat{A}\widehat{P}\widehat{A}^T + (\widetilde{G} - \widehat{A}\widehat{P}\widehat{C}^T)(\widehat{\Lambda}_0 - \widehat{C}\widehat{P}\widehat{C}^T)^{-1}(\widetilde{G} - \widehat{A}\widehat{P}\widehat{C}^T)^T. \tag{4.5}$$

Note that in the latter equation, $\widetilde{\Lambda}_0$ was again replaced by $\widehat{\Lambda}_0$. The equation is quadratic in the unknown $\widetilde{G}$ and can easily be solved. The procedure proposed in [139] determines Cholesky factors L_1 and L_2 such that

$$L_1^T L_1 = \widehat{\Lambda}_0 - \widehat{C}\widehat{P}\widehat{C}^T, \;\; L_2^T L_2 = \widetilde{\Lambda}_0 - \widehat{C}\widehat{P}\widehat{C}^T,$$

and calculates $\widetilde{G}$ as

$$\widetilde{G} = \widehat{G} L_2^{-1} L_1 + \widehat{A}\widehat{P}\widehat{C}^T (I_l - L_2^{-1} L_1).$$

It can easily be checked that this is indeed a solution to (4.5). The resulting model $(\widehat{A}, \widetilde{G}, \widehat{C}, \widehat{\Lambda}_0)$ is positive real. It is obvious that this method only works for stable models, which can be considered to be its biggest disadvantage.

4.2.3 Constrained optimization using Semi Definite Programming

In [102] a proposal was made for a new identification scheme based on existing stochastic subspace methods and Semi Definite Programming (SDP). A stable $\widetilde{A}$ is obtained by solving:

$$(\widetilde{A}, \widehat{P}) = \arg\min_{A,P} \|(A - \widehat{A})P\|_2 \qquad \text{s.t.} \left\{ \begin{array}{l} P > 0 \\ P - APA^T > 0 \end{array} \right.$$

Positive realness is thereafter imposed by solving a similar SDP-problem involving vectors of stacked covariance sequences. The algorithm is quite involved and a full analysis is beyond the scope of this thesis. The interested reader is referred to [102]. Note that this algorithm is applicable to stable as well as unstable models and therefore more widely usable than the other methods so far discussed.

4.2.4 Using the residuals in a least-squares approach

As outlined in 3.6.4, estimates for the matrices A and C in stochastic subspace identification can be obtained using the shift-invariance property of the extended observability matrix Γ_i, or using the steady state Kalman filter state sequences $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ through the equation

$$\begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A \\ C \end{bmatrix} \widetilde{X}_i + \begin{bmatrix} \widehat{\rho}_w \\ \widehat{\rho}_v \end{bmatrix}. \tag{4.6}$$

In line with what is done in the combined stochastic deterministic case, the residuals $\widehat{\rho}_w$ and $\widehat{\rho}_v$ in (4.6) can in principle be used to obtain estimates for the noise matrices Q, R and S as

$$\begin{bmatrix} \widehat{Q} & \widehat{R} \\ \widehat{R}^T & \widehat{S} \end{bmatrix} = E\left\{ \frac{1}{j} \begin{bmatrix} \widehat{\rho}_w \\ \widehat{\rho}_v \end{bmatrix} \begin{bmatrix} \widehat{\rho}_w^T & \widehat{\rho}_v^T \end{bmatrix} \right\}, \tag{4.7}$$

which does not involve an implicit calculation of the covariance model and avoids possible positive realness problems. A drawback of this approach is that the noise matrices are only consistently estimated for $i \to \infty$. Hence the reason why the approach involving the Riccati equation is much more commonly used in stochastic subspace identification. However, in case of positive realness problems it is proposed in [147] to bypass the Riccati equation and use (4.7), ignoring the bias that will be introduced. An advantage of this approach is that it is intuitive and perfectly in line with what is done in the stochastic-deterministic case. Again, the biggest disadvantage is that this approach is limited to stable covariance models.

4.3 Imposing positive realness using Tikhonov regularization

In this section, a new method will be proposed to impose positive realness on a covariance model $\widehat{A}, \widehat{G}, \widehat{C}, \widehat{\Lambda}_0$ using the concept of weighted Tikhonov regularization on A and C. Tikhonov regularization, which is discussed in more detail in Appendix C is an effective way to deal with ill-conditioned least-squares problems or impose certain conditions on the solutions of such problems. In this section it will be shown that using Tikhonov regularization, positive-realness can be imposed on the covariance model obtained in stochastic subspace identification methods. The presented method works for stable as well as unstable models and will be shown to outperform existing approaches in Section 4.4.

4.3.1 Regularization on $\widehat{A}$ and $\widehat{C}$

We will assume that estimates $\widehat{A}$ and $\widehat{C}$ for A and C are obtained using the state estimates $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ through (4.6). Formally, this is written as

$$(\widehat{A}, \widehat{C}) = \arg\min_{A,C} J_1(A, C), \tag{4.8}$$

with

$$J_1(A, C) = \left\| \begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} A \\ C \end{bmatrix} \cdot \widetilde{X}_i \right\|_F^2 .$$

To impose positive realness, we will add a regularization term to the cost function $J_1(A, C)$ from (4.8):

$$(\widetilde{A}_c, \widetilde{C}_c) = \arg\min_{A,C} \left(J_1(A, C) + cJ_2(A, C) \right), \tag{4.9}$$

with

$$J_2(A, C) = \mathrm{Tr}\left(\begin{bmatrix} A \\ C \end{bmatrix} W \begin{bmatrix} A \\ C \end{bmatrix}^T\right),$$

where $c \geq 0$ is a positive real scalar and W a positive definite matrix of appropriate dimensions that satisfies $W - \widehat{G}\widehat{\Lambda}_0^{-1}\widehat{G}^T \geq 0$. A typical choice is the identity matrix, which is motivated by [74].

In order to get a feel for the effects of the addition of the regularization term $cJ_2(A, C)$ we note that a similar regularization term $c\mathrm{Tr}\left(AWA^T\right)$, involving only the system matrix A was described in [142], and was shown to impose stability on a model. This can intuitively be understood from the fact that with W the identity matrix, the term $\mathrm{Tr}\left(AWA^T\right)$ is nothing else than the sum of the squared poles of the system. Hence, adding an extra term $c\mathrm{Tr}\left(AWA^T\right)$ to the costfunction effectively drags the poles inside the unit circle as c is increased. Although for the term $cJ_2(A, C)$ such an intuitive reasoning is not straightforwardly available, we will prove that by the choice of the regularization term $cJ_2(A, C)$ the covariance model can not only be made stable, but also positive real, provided the regularization coefficient c is chosen sufficiently large. A further advantage of the regularization term is that the problem (4.9) remains quadratic and that the optimal solution follows from a linear set of equations:

$$\begin{bmatrix} \widetilde{A}_c \\ \widetilde{C}_c \end{bmatrix} = \begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} \cdot \widetilde{X}_i^T \cdot \left[\widetilde{X}_i\widetilde{X}_i^T + cW\right]^{-1} = \begin{bmatrix} \widehat{A} \\ \widehat{C} \end{bmatrix} \widetilde{X}_i\widetilde{X}_i^T \left[\widetilde{X}_i\widetilde{X}_i^T + cW\right]^{-1}, \quad (4.10)$$

where we remind the reader that $\widehat{A}$ and $\widehat{C}$ are the unregularized estimates for A and C based on a finite set of output covariances. From the optimality of the least squares estimate (4.10), it follows that for $c_1, c_2 \geq 0$

$$J_1(\widetilde{A}_{c_2}, \widetilde{C}_{c_2}) + c_1 J_2(\widetilde{A}_{c_2}, \widetilde{C}_{c_2}) \geq J_1(\widetilde{A}_{c_1}, \widetilde{C}_{c_1}) + c_1 J_2(\widetilde{A}_{c_1}, \widetilde{C}_{c_1}), \quad (4.11)$$

$$J_1(\widetilde{A}_{c_1}, \widetilde{C}_{c_1}) + c_2 J_2(\widetilde{A}_{c_1}, \widetilde{C}_{c_1}) \geq J_1(\widetilde{A}_{c_2}, \widetilde{C}_{c_2}) + c_2 J_2(\widetilde{A}_{c_2}, \widetilde{C}_{c_2}), \quad (4.12)$$

where (4.12) can be rewritten as:

$$\begin{gathered} J_1(\widetilde{A}_{c_1}, \widetilde{C}_{c_1}) + c_1 J_2(\widetilde{A}_{c_1}, \widetilde{C}_{c_1}) + (\Delta c) J_2(\widetilde{A}_{c_1}, \widetilde{C}_{c_1}) \geq \\ J_1(\widetilde{A}_{c_2}, \widetilde{C}_{c_2}) + c_1 J_2(\widetilde{A}_{c_2}, \widetilde{C}_{c_2}) + (\Delta c) J_2(\widetilde{A}_{c_2}, \widetilde{C}_{c_2}), \end{gathered} \quad (4.13)$$

with $\Delta c = c_2 - c_1$. Combining (4.11) and (4.13) it is easily seen that the regularization term $J_2(\widetilde{A}_c, \widetilde{C}_c)$ is a non-increasing function of c.

The idea of using regularization to deal with undesirable properties of an estimator is by no means new. The concept of regularization amounts to reducing the variance of an estimator at the expense of introducing a hopefully small bias, the so-called bias-variance trade-off. In function approximation, for instance, regularization is used to impose a certain amount of smoothness and deal with the well known problem of over-fitting [55]. Other applications are found in areas as neural networks [18], Support Vector Machines [132], and system identification [130]. Furthermore, some known techniques can be rewritten in a regularization context. The technique described in [142] to impose stability on a model using regularization, for instance, is essentially equivalent to a technique described in [28], provided a certain choice for the weighting matrices is made in the former reference. In a sense, most classical algorithms outlined in Section 4.2 can also be considered as a form of regularization, in that an undesirable property of the covariance model is removed at the expense of the introduction of a small bias. As mentioned before the type of regularization that is used in this chapter is known as weighted Tikhonov regularization and is briefly explained in Appendix C

4.3.2 Choosing the regularization parameter

It will be shown in the following lemma that, by using the regularization term introduced in (4.9), positive realness can always be imposed provided the regularization coefficient c is chosen sufficiently large [65].

Lemma 4.1. Regularization on $\widehat{A}$ and $\widehat{C}$: *Let $\widehat{G}$, $\widehat{\Lambda}_0, \widehat{A}, \widehat{C}$ be given. Let $W \geq 0$ be chosen such that $W - \widehat{G}\widehat{\Lambda}_0\widehat{G}^T \geq 0$, and define $\widehat{\Sigma} = \widetilde{X}_i\widetilde{X}_i^T$, $\mathcal{L} = \widehat{\Sigma}\begin{bmatrix}\widehat{A}^T & \widehat{C}^T\end{bmatrix}\begin{bmatrix}W & \widehat{G} \\ \widehat{G}^T & \widehat{\Lambda}_0\end{bmatrix}^{-1}\begin{bmatrix}\widehat{A} \\ \widehat{C}\end{bmatrix}\widehat{\Sigma}$, $P_0 = \widehat{\Sigma}W^{-1}\widehat{\Sigma} - \mathcal{L}$. Suppose the covariance model $(\widehat{A}, \widehat{G}, \widehat{C}, \widehat{\Lambda}_0)$ is not positive real. Then there exists a c^* such that the system $(\widetilde{A}_c$, $\widehat{G}$, $\widetilde{C}_c$, $\widehat{\Lambda}_0)$, with $\widetilde{A}_c$ and $\widetilde{C}_c$ as in (4.10), is positive real for $c \geq c^*$, with $c^* = max_{i|\theta_i \in \mathbb{R}^+}\theta_i$, and $\{\theta_i\}_{i=1}^{2n}$ the set of generalized eigenvalues of the following eigenvalue problem (with n the model order):*

$$\theta = \lambda\left(\begin{bmatrix}0_n & -I_n \\ P_0 & 2\widehat{\Sigma}\end{bmatrix}, -\begin{bmatrix}I_n & 0_n \\ 0_n & W\end{bmatrix}\right).$$

Proof. We show that (4.4), with A, C, G and P replaced by $\widetilde{A}_c$, $\widetilde{C}_c$, $\widehat{G}$ and $\widehat{P}$, holds under the assumptions of the lemma for $\widehat{P} = W$. This means that

$$\begin{bmatrix}W & \widehat{G} \\ \widehat{G}^T & \widehat{\Lambda}_0\end{bmatrix} - \begin{bmatrix}\widetilde{A}_cW\widetilde{A}_c^T & \widetilde{A}_cW\widetilde{C}_c^T \\ \widetilde{C}_cW\widetilde{A}_c^T & \widetilde{C}_cW\widetilde{C}_c^T\end{bmatrix} \geq 0, \tag{4.14}$$

where the first term is positive semidefinite since $W \geq 0$, $W - \widehat{G}\widehat{\Lambda}_0^{-1}\widehat{G}^T \geq 0$, and $\widetilde{A}_c$ and $\widetilde{C}_c$ are as defined in (4.10). Notice that the left hand side of (4.14) can be seen as the Schur complement of the following matrix (see Appendix B for an introduction into the Schur complement and its relation to positive definiteness of a matrix):

$$\left[\begin{array}{cc|c} W & \widehat{G} & \widehat{A}\widehat{\Sigma} \\ \widehat{G}^T & \widehat{\Lambda}_0 & \widehat{C}\widehat{\Sigma} \\ \hline \widehat{\Sigma}\widehat{A}^T & \widehat{\Sigma}\widehat{C}^T & (\widehat{\Sigma} + cW)W^{-1}(\widehat{\Sigma} + cW)\end{array}\right] \geq 0, \tag{4.15}$$

which must be positive semi-definite since (4.14) holds and $(\widehat{\Sigma}+cW)W^{-1}(\widehat{\Sigma}+cW)$ is positive semi-definite by construction. Again taking the Schur complement of (4.15), with $W-\widehat{G}\widehat{\Lambda}_0^{-1}\widehat{G}^T \geq 0$, we obtain

$$(\widehat{\Sigma}+cW)W^{-1}(\widehat{\Sigma}+cW)-\mathcal{L} \geq 0.$$

This can also be written as

$$c^2W+2c\widehat{\Sigma}+\widehat{\Sigma}W^{-1}\widehat{\Sigma}-\mathcal{L} \geq 0. \tag{4.16}$$

Equation (4.16) is clearly satisfied for $c \to \infty$. The exact lower bound c^* for (4.16) to hold is given by the largest non-negative root of

$$\det\left(c^2W+2c\widehat{\Sigma}+\widehat{\Sigma}W^{-1}\widehat{\Sigma}-\mathcal{L}\right)=0.$$

Using the definition of P_0, this reduces to

$$\begin{aligned}\det\left(c^2W+2c\widehat{\Sigma}+P_0\right)=0 \quad &\Longleftrightarrow \quad \det\left(c\left(cW+2\widehat{\Sigma}\right)+P_0\right)=0\\ &\Longleftrightarrow \quad \det\left(c\begin{bmatrix}I_n & 0_n\\ 0_n & W\end{bmatrix}+\begin{bmatrix}0_n & -I_n\\ P_0 & 2\widehat{\Sigma}\end{bmatrix}\right)=0.\end{aligned}$$

□

From Lemma 4.1 it follows that a positive real model is always obtained for $c \geq c^*$, and in particular for $c = c^*$. Furthermore, since any positive real model is necessarily stable (which follows immediately from the upper left part of (4.4)), stability is automatically guaranteed. However, c^* can be a too conservative estimate. In general it seems reasonable to keep the amount of regularization as low as possible. Hence one should search for the smallest possible $c \leq c^*$ for which a positive real model is found. A lower bound c_s for c can be found from a theorem presented in [142], where c_s follows from a generalized eigenvalue problem and is shown to be the smallest c imposing stability on the estimated covariance model. As shown in Figure 4.2, a minimal c imposing positive realness will always satisfy $c_s \leq c \leq c^*$. When the realization $(\widetilde{A}_{c_s}, \widehat{G}, \widetilde{C}_{c_s}, \widehat{\Lambda}_0)$ is not yet positive real, i.e., $\Phi(z) < 0$ for a certain $z = e^{j\theta}$, we can find a $c \geq c_s$ imposing positive realness, for instance by applying a bisection algorithm or an iterative search with progressively refined grid on the interval $c_s \leq c \leq c^*$.

4.4 Comparing the presented algorithm to existing techniques

In this section we will compare the presented algorithm to the existing techniques discussed in the beginning of this chapter. This comparison will be performed by hand of a set of Monte-Carlo simulations.

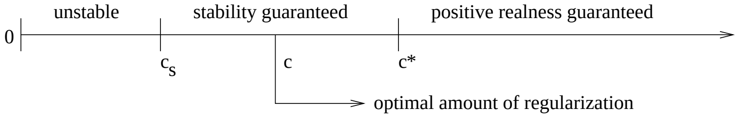

unstable
stability guaranteed
positive realness guaranteed
0
c
s
c
c*
optimal amount of regularization

Figure 4.2: The optimal amount of regularization c is certainly larger than the amount of regularization that is needed to make the covariance model stable c_s. Furthermore it is certainly smaller or equal to c^* which was shown to impose positive realness in Lemma 4.1.

4.4.1 Examples

Gaussian, zero mean, unit variance, white noise sequences where fed into two known systems to create a set of output sequences. For each output sequence the stochastic subspace identification algorithm presented in Figure 3.8 was used in combination with techniques to impose positive realness where necessary. The systems that were used for the simulation are the following:

$$\begin{aligned} H_1(z) &= \frac{(z-0.99e^{\pm 2j})(z-0.98e^{\pm 1.4j})(z-0.99e^{\pm 0.6j})(z-0.9)(z+0.9)}{(z-0.8e^{\pm 2.1j})(z-0.8e^{\pm j})(z-0.8e^{\pm 1.7j})(z-0.8e^{0.8j})} \\ H_2(z) &= \frac{(z-0.85e^{\pm 2.3562j})(z-0.8999e^{\pm 0.7853j})(z-0.9802)}{(z-0.9e^{\pm 3j})(z-0.9196e^{\pm 0.1998j})(z-0.8507)}, \end{aligned}$$

where the latter is an example that was previously used in [102] to study the performance of the SDP-technique. Results for these two systems are reported in Table 4.1 for $H_1(z)$ and Table 4.2 for $H_2(z)$. The abbreviations for the different techniques used in the tables are the following:

- REG$_{\widehat{\Lambda}_0}$: Adapting $\widehat{\Lambda}_0$ such as explained in 4.2.1. A bisection algorithm was used to determine the optimal change to $\widehat{\Lambda}_0$.
- REG$_{\widehat{G}}$: Adapting $\widehat{G}$ such as explained in 4.2.2. Again a bisection algorithm was used in the implementation.
- SDP: A technique based on semi definite programming problems explained in 4.2.3. The performance of the SDP-technique was evaluated using software written by the authors and published on their website.
- RES: Estimating the noise model using the residuals such as explained in 4.2.4.
- REG$_{\widehat{A},\widehat{C}}$: Regularization on $\widehat{A}$ and $\widehat{C}$.

Each table contains the results of 4 different experiments, each with a different choice of the parameters n (order of the model), i (number of block-rows), and N (number of observations). For each experiment, 1000 Gaussian white

$n = 8, i = 16, N = 500$			Not positive real 528/1000				Unstable 0/1000
	Stable models					Unstable models	
	$\text{REG}_{\widehat{A},\widehat{C}}$	$\text{REG}_{\widehat{G}}$	$\text{REG}_{\widehat{\Lambda}_0}$	RES	SDP	$\text{REG}_{\widehat{A},\widehat{C}}$	SDP
Mean(d_∞)	1.6	2.05	2.24	1.46	9.54	-	-
Var(d_∞)	0.324	0.666	0.624	0.239	504	-	-
Mean(d_2)	0.571	0.695	0.771	0.549	1.93	-	-
Var(d_2)	0.0181	0.0573	0.0566	0.0146	3.41	-	-
Mean(d_1)	1.35	1.76	1.82	1.32	3.47	-	-
Var(d_1)	0.0813	0.459	0.31	0.0673	3.38	-	-

$n = 8, i = 12, N = 500$			Not positive real 794/1000				Unstable 4/1000
	Stable models					Unstable models	
	$\text{REG}_{\widehat{A},\widehat{C}}$	$\text{REG}_{\widehat{G}}$	$\text{REG}_{\widehat{\Lambda}_0}$	RES	SDP	$\text{REG}_{\widehat{A},\widehat{C}}$	SDP
Mean(d_∞)	1.55	2.19	2.42	1.48	3.59	2.48	8.01
Var(d_∞)	0.253	0.684	0.665	0.518	37.2	0.0523	68.4
Mean(d_2)	0.577	0.75	0.846	0.549	1.12	1	1.62
Var(d_2)	0.0171	0.0662	0.0716	0.0159	0.274	0.00826	0.8
Mean(d_1)	1.37	1.87	2.02	1.29	2.54	2.47	3.16
Var(d_1)	0.0881	0.475	0.426	0.0662	0.843	0.0872	1

$n = 8, i = 16, N = 1000$			Not positive real 544/1000				Unstable 1/1000
	Stable models					Unstable models	
	$\text{REG}_{\widehat{A},\widehat{C}}$	$\text{REG}_{\widehat{G}}$	$\text{REG}_{\widehat{\Lambda}_0}$	RES	SDP	$\text{REG}_{\widehat{A},\widehat{C}}$	SDP
Mean(d_∞)	1.15	1.58	1.75	1.05	8	1.46	41.9
Var(d_∞)	0.157	0.495	0.445	0.0977	1.84e+03	-	-
Mean(d_2)	0.413	0.533	0.602	0.418	1.48	0.55	8.05
Var(d_2)	0.00939	0.0456	0.0407	0.00594	4.41	-	-
Mean(d_1)	0.972	1.5	1.41	1.03	2.74	1.44	10.3
Var(d_1)	0.0431	0.716	0.208	0.0275	2.43	-	-

$n = 10, i = 16, N = 500$			Not positive real 727/1000				Unstable 182/1000
	Stable models					Unstable models	
	$\text{REG}_{\widehat{A},\widehat{C}}$	$\text{REG}_{\widehat{G}}$	$\text{REG}_{\widehat{\Lambda}_0}$	RES	SDP	$\text{REG}_{\widehat{A},\widehat{C}}$	SDP
Mean(d_∞)	1.7	2.45	2.63	2.19	18.3	2.48	15.1
Var(d_∞)	0.488	1.57	1.04	6.02	4e+03	2.31	3.76e+03
Mean(d_2)	0.579	0.784	0.889	0.591	2.46	0.696	2.06
Var(d_2)	0.0172	0.0907	0.117	0.0294	8.59	0.0298	9.65
Mean(d_1)	1.36	1.98	2.12	1.35	4.08	1.65	3.47
Var(d_1)	0.0709	0.643	0.76	0.0655	6.7	0.174	6.82

Table 4.1: Performance for various techniques over a Monte-Carlo simulation with 1000 datasets generated using the linear system $(H_1(z))$ with model orders $n = 8$ or $n = 10$, block-rows $i = 12$ or $i = 16$ and number of data-points $N = 500$ or $N = 1000$. The number of covariance models that needed corrections for stability and/or positive realness are given for every case. Mean and standard deviations of the distance measures d_∞, d_2 and d_1 over the 1000 datasets are also given. Note that these measures are consistently lower for $\text{REG}_{\widehat{A},\widehat{C}}$ and RES with respect to other techniques. However, a disadvantage with RES is that its applicability is limited to stable models. Another interesting observation from the table is that the number of unstable models increases as the model order increases and that the number of non-positive real covariance models increases with decreasing i (see also Section 4.4.2).

$n = 5, i = 10, N = 500$				Not positive real 419/1000			Unstable 39/1000
	Stable models					Unstable models	
	$\text{REG}_{\widehat{A},\widehat{C}}$	$\text{REG}_{\widehat{G}}$	$\text{REG}_{\widehat{\Lambda}_0}$	RES	SDP	$\text{REG}_{\widehat{A},\widehat{C}}$	SDP
Mean(d_∞)	5.07	10.1	11.1	9.6	49.1	6.71	27.7
Var(d_∞)	4.45	15.2	19.5	836	5.19e+04	36.4	5.23e+03
Mean(d_2)	1.31	2.95	3.31	1.41	4.76	1.75	3.72
Var(d_2)	0.175	1.5	1.96	0.508	60.4	0.324	32
Mean(d_1)	2.15	5.19	5.92	2.01	5.3	2.95	4.55
Var(d_1)	0.356	4.65	6.54	0.602	25.5	0.84	18.6

$n = 5, i = 8, N = 500$				Not positive real 425/1000			Unstable 35/1000
	Stable models					Unstable models	
	$\text{REG}_{\widehat{A},\widehat{C}}$	$\text{REG}_{\widehat{G}}$	$\text{REG}_{\widehat{\Lambda}_0}$	RES	SDP	$\text{REG}_{\widehat{A},\widehat{C}}$	SDP
Mean(d_∞)	5.02	9.28	10.1	6.18	12.5	7.3	9.21
Var(d_∞)	4.09	17	22.1	35	532	45.9	390
Mean(d_2)	1.31	2.67	2.95	1.37	2.52	1.79	1.9
Var(d_2)	0.167	1.66	2.24	0.335	7.57	0.601	1.88
Mean(d_1)	2.14	4.66	5.24	1.96	3.42	2.95	2.95
Var(d_1)	0.345	5.17	7.54	0.485	7.26	1.85	1.32

$n = 5, i = 10, N = 1000$				Not positive real 399/1000			Unstable 4/1000
	Stable models					Unstable models	
	$\text{REG}_{\widehat{A},\widehat{C}}$	$\text{REG}_{\widehat{G}}$	$\text{REG}_{\widehat{\Lambda}_0}$	RES	SDP	$\text{REG}_{\widehat{A},\widehat{C}}$	SDP
Mean(d_∞)	3.84	10.8	12.1	6.26	31	5.22	4.54
Var(d_∞)	2.18	12.6	14.6	67.2	6.47e+03	1.71	1.74
Mean(d_2)	1.02	3.24	3.66	1.14	4.16	1.56	1.41
Var(d_2)	0.102	1.2	1.46	0.302	26.3	0.146	0.214
Mean(d_1)	1.67	5.65	6.55	1.62	4.78	2.6	2.49
Var(d_1)	0.214	3.9	5.01	0.389	16.6	0.404	0.57

$n = 8, i = 10, N = 500$				Not positive real 632/1000			Unstable 427/1000
	Stable models					Unstable models	
	$\text{REG}_{\widehat{A},\widehat{C}}$	$\text{REG}_{\widehat{G}}$	$\text{REG}_{\widehat{\Lambda}_0}$	RES	SDP	$\text{REG}_{\widehat{A},\widehat{C}}$	SDP
Mean(d_∞)	5.87	9.93	10.8	7.81	21.7	8.61	33.6
Var(d_∞)	7.83	16.1	19.9	66.7	9.27e+03	24.4	7.83e+04
Mean(d_2)	1.38	2.74	3.06	1.47	2.53	1.79	2.81
Var(d_2)	0.19	1.54	2.04	0.318	23.1	0.513	35.9
Mean(d_1)	2.17	4.77	5.37	2.07	3.07	2.88	3.38
Var(d_1)	0.317	5.14	7.33	0.371	8.42	1.5	19.6

Table 4.2: Performance for various techniques over a Monte-Carlo simulation with 1000 datasets generated using the linear system ($H_2(z)$) with model orders $n = 5$ or $n = 8$, block-rows $i = 8$ or $i = 10$ and number of data-points $N = 500$ or $N = 1000$. The number of covariance models that needed corrections for stability and/or positive realness are given for every case. Mean and standard deviations of the distance measures d_∞, d_2 and d_1 over the 1000 datasets are also given. Note that these measures are consistently lower for $\text{REG}_{\widehat{A},\widehat{C}}$ and RES with respect to other techniques. However, a disadvantage with RES is that its applicability is limited to stable models. Another interesting observation from the table is that the number of unstable models increases as the model order increases.

noise-sequences with zero mean and unit variance were generated with the desired length N, and an equal number of covariance models were produced. The number of covariance models that needed corrections for stability and/or positive realness are displayed in the table. Remembering that unstable models are always non-positive real the latter number will always be greater than the former. Below this information, the performance of each technique on these non-positive real models is given. The performance on all non-positive real, but stable models is given at the left. The results for the unstable models are given at the right for those methods which can deal with unstable covariance models. The performance measures d_∞, d_2, d_1 used in the tables are norms of the differences between the transfer functions of the simulated and the identified stochastic models in forward innovation form:

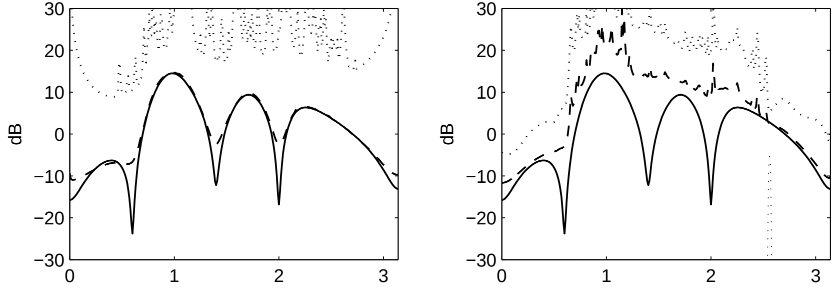
$$d_p = \left\| H(z) - \widehat{H}(z)_{(\widehat{A},\widehat{K},\widehat{C},1)} \right\|_p, \quad p = 1, 2, \infty.$$

4.4.2 Discussion

Two techniques, RES and REG$_{\widehat{A},\widehat{C}}$ clearly outperform the others. For some experiments the former results in slightly better estimates. However, problems with this method might occur as the system order is increased. To visualize this, in Figure 4.3 the estimated spectral density $\widehat{H}_1(z)\widehat{H}_1(z)^T$ for the first example ($H_1(z)$), fourth experiment ($n = 10$, $i = 16$, $N = 500$) averaged over all 818 stable runs (including the ones which did not need correction) are given, together with the spectral density of the original model and a 2σ error bound on the latter. Hereby, σ is the standard deviation on the spectral density as obtained from the Monte-Carlo simulation. It is clear that the uncertainty on the obtained spectral density is much higher when using the RES technique than when using the REG$_{\widehat{A},\widehat{C}}$ technique. As for the complexity, all algorithms discussed in this chapter are roughly $\mathcal{O}(qn^3)$, with q the number of iterations necessary to find a regularization constant c or to solve an SDP problem. For RES, $q = 1$ as no optimization is performed.

Apart from the performance of the different techniques, it is also interesting to have a look into the influence of the parameters n, i and N on the occurrence of positive realness problems. In Table 4.1, decreasing i from 16 to 12 clearly resulted in a much higher number of non positive real models. In Table 4.2, however, the number of non positive real models remained largely the same when decreasing i. It is well known that when the modeling order n increases, the probability to obtain unstable models increases considerably (see also [142]). This can also be observed in Tables 4.1 and 4.2. Finally, it can be observed that for the examples described in this chapter the influence of N on the occurrence of positivity problems is relatively low compared to that of n and i.

(a) RES (b) SDP



(c) $\mathrm{REG}_{\widehat{G}}$ (d) $\mathrm{REG}_{\widehat{A},\widehat{C}}$

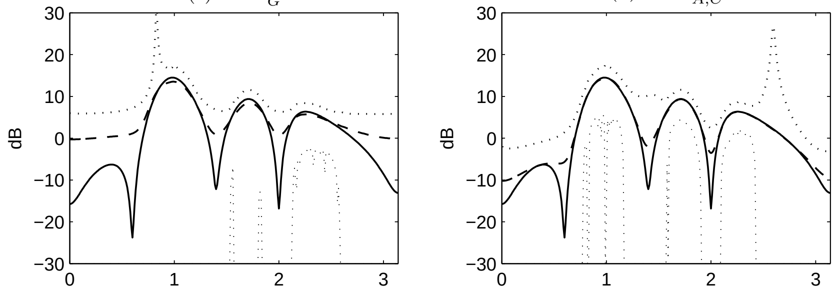


Figure 4.3: Averaged Spectral density over 1000 runs for the example ($H_1(z)$) with $n = 10$, $i = 16$, $N = 500$ (dashed line) with 2σ error region (dotted line). The solid line is the spectral density of the original model used for simulation. Two techniques, RES and $\mathrm{REG}_{\widehat{A},\widehat{C}}$ clearly outperform the others.

4.5 A practical application

The regularization procedure described in this chapter was used to identify a stochastic subspace model from 2 minutes of measurements on a steel transmitter mast for cellular phone networks [115], which is displayed in Figure 4.4. Nine accelerometers were placed on the mast and the mast's response on the wind turbulence was measured with a sampling rate of 100Hz for about 5 minutes. Thereafter, the data was downsampled by a factor of 8. A 16^{th}-order stochastic SISO subspace model was created based on data from one of the accelerometers and using subspace identification with i, the number of block rows, set to 32. For this set of parameters a stable, but non positive real covariance model was obtained, where-after the different regularization techniques described in this chapter were used to obtain positive real models. The original measurement spectrum and the modeled spectra resulting from the two best performing techniques in the simulations of section 4.4, namely RES and $\mathrm{REG}_{\widehat{A},\widehat{C}}$ are displayed in Figure 4.5, together with the absolute values of the differences between them. Note that all the spectra are strictly positive. Also note that the RES technique performs better in the regions between the

Figure 4.4: Vibration measurements on a steel transmitter mast were used to evaluate the performance of the algorithm proposed in this chapter. The transmitter mast is located in Antwerp, Belgium and is part of a cellular phone network.

peaks, while $\mathrm{REG}_{\widehat{A},\widehat{C}}$ is seen to fit the peaks themselves better. For comparison, the variances of the model fit errors for $\mathrm{REG}_{\widehat{G}}$ and SDP are given below the figure.

4.6 Conclusions

Stochastic subspace methods for the identification of linear time-invariant systems are known to be asymptotically unbiased [143]. However, if a finite amount of data is used, the procedure might break down due to positive realness problems. In this chapter a regularization approach was proposed to impose positive realness on a formerly identified covariance model. It was shown that, if an adequate amount of regularization is used, a positive real model can always be obtained. The simulation results indicate that this new approach in general yields better models than other existing techniques. Similarly, the approach was seen to be useful for the analysis of practical datasets as in the area of structural identification and vibration analysis.

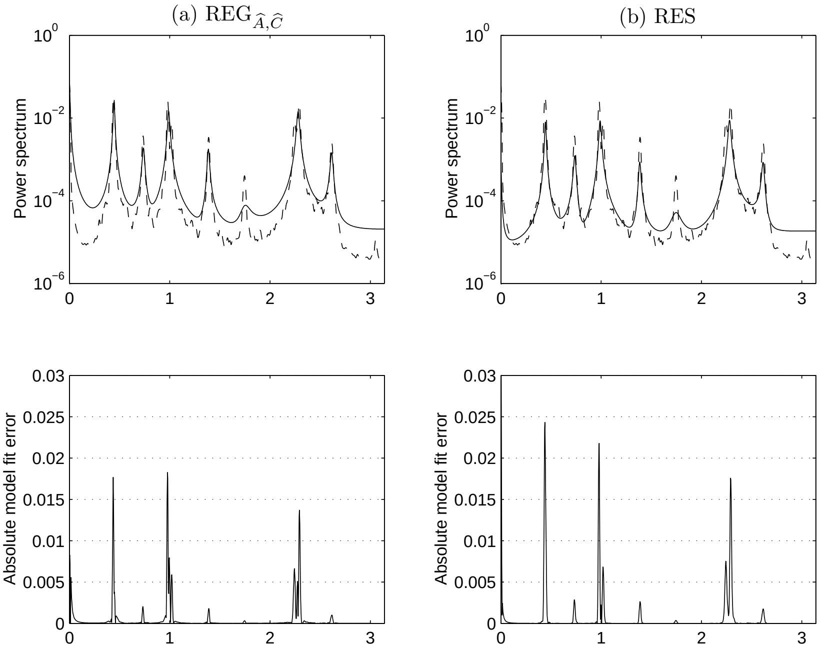


Figure 4.5: Output spectra of one of the accelerometers on a steel mast (dashed lines), together with the estimated spectra using REG$_{\widehat{A},\widehat{C}}$ and RES (full line). The absolute differences between the spectra in the uppermost two figures are depicted in the figures at the bottom. The variances of the differences are $3.71 \cdot 10^{-6}$ for the REG$_{\widehat{A},\widehat{C}}$ case and $11.05 \cdot 10^{-6}$ for the RES case. In similar experiments the variances for the REG$_{\widehat{G}}$ and SDP techniques were found to be $10.47 \cdot 10^{-6}$ and $15.05 \cdot 10^{-6}$ respectively.

Chapter 5

Ill-conditioning in subspace identification

Over the last few years, experimental evidence has been mounting that in certain experimental conditions, combined subspace identification algorithms, and especially the N4SID algorithm, may run into ill-conditioning and lead to ambiguous results. Various reasons for this phenomenon will be explored in this chapter and a recently proposed solution, the so-called orthogonal decomposition method, will be discussed. The orthogonal decomposition method will be seen to differ from N4SID by the replacement of the oblique projection with an orthogonal projection, and by a preliminary decomposition of the system in a deterministic and a stochastic component. An improved version of N4SID will thereafter be proposed which copies the idea of the decomposition in a deterministic and a stochastic part but maintains the oblique projection. In order to avoid conditioning problems, regularization will be performed on the latter. It will be shown that an improved regularized N4SID-algorithm is obtained which can compete with the orthogonal decomposition method, even under difficult experimental conditions.

5.1 Introduction

The majority of linear subspace identification algorithms with known inputs are of the combined stochastic-deterministic type. The reason for the popularity of these methods is that the estimated state sequence $\widetilde{X}_i$, which is expressed as a function of the past input- and output-data contained in W_p, can conveniently be interpreted as the result of a non-steady state Kalman filter (see Subsections 3.5.4 and 3.5.5). This in contrast to approaches based on separately parameterized stochastic and deterministic subsystems, introduced in Section 3.7, where the estimated state is solely expressed in terms of the past inputs U_p. Hence, combined stochastic-deterministic subspace approaches can somewhat loosely be seen as the subspace equivalent of ARX modeling [84, 85], whereas an algorithm as the PI-MOESP is more closely related to the basic ideas behind FIR modeling [116]. Especially for a small number of block-rows i in U_p, combined algorithms can be expected to outperform algorithms based on separately parameterized models.

Nevertheless, it was observed in several papers [21, 25, 26, 89] that in certain experimental conditions the standard combined stochastic-deterministic subspace identification methods, and most notably the N4SID, may yield unreliable results. It was argued in these publications that this behavior can be explained in terms of an ill-conditioning of the multiple regression problem underlying the oblique projection $Y_f /_{U_f} W_p$ and/or a correlation between the stochastic contribution to the outputs of the system and the system inputs. In this chapter we will discuss both sources of ill-conditioning and show that they are to a large extent caused by the same phenomenon, namely a high coloring in the systems inputs. Existing methods to deal with the bad conditioning of the oblique projection by replacing it by an orthogonal projection will be discussed and analyzed. We will argue that combined methods such as PO-MOESP and CVA, where the oblique projection is replaced by an orthogonal projection, have some specific advantages over N4SID when faced with conditioning problems. Nevertheless, in the presence of strong stochastic resonances in frequency bands where the input power is low, any existing combined method will be seen to yield unreliable results. Using separately parameterized model structures, based on

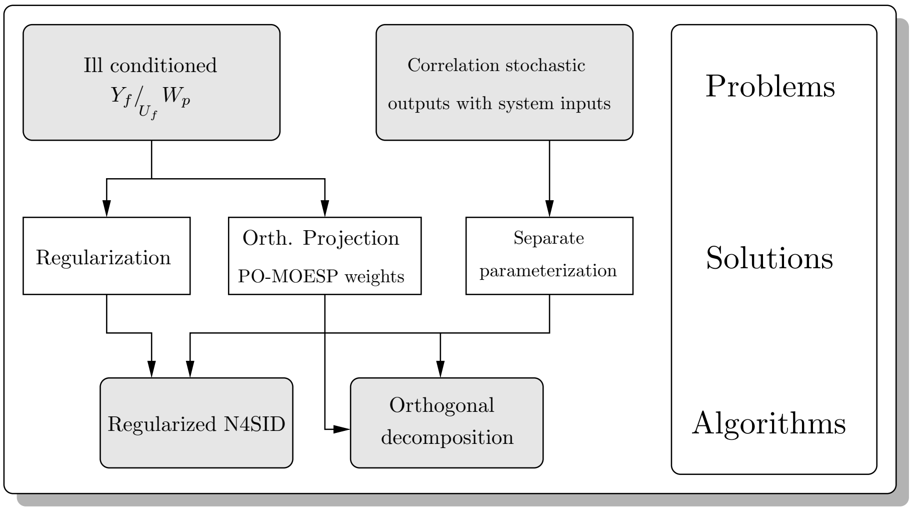


Figure 5.1: Short overview of two types of problems discussed in this chapter (first line), possible solutions (second line), and algorithmic implementations of these solutions (third line).

the work in [21, 26] will be introduced as a remedy against this phenomenon. Special attention will thereby go to the orthogonal decomposition method which combines the advantages of PO-MOESP with a separate parameterization and is introduced in [23] as the most likely candidate to replace classical combined approaches when faced with conditioning problems.

The most important contribution of this chapter will be the introduction of an alternative to the orthogonal decomposition algorithm which copies the idea of the separate parameterization but reinstates the oblique projection $Y_f /_{U_f} W_p$ as the main geometrical operation to estimate the system state. An extra regularization term will be added to the oblique projection to avoid possible cases of ill-conditioning. Numerical results will be provided to prove that under optimal as well as difficult experimental conditions, the newly proposed regularized N4SID algorithm is competitive with the orthogonal decomposition method. The aim of this chapter is thereby not the introduction itself of yet another subspace method, but to demonstrate that regularization can play an important role in the initial steps of subspace identification.

This chapter is organized as follows. In Section 5.2 possible causes of ill-conditioning will be investigated. In Section 5.3 the orthogonal decomposition method will be introduced as a solution to ill-conditioning, based on the work presented in [21, 22, 26]. In Section 5.4 a new algorithm will be proposed based on weighted regularization in the oblique projection $Y_f /_{U_f} W_p$. In Section 5.5, finally, the performance of the newly proposed algorithm will be investigated on a number of examples. A graphical overview of the problems, solutions and resulting algorithms presented in this chapter is given in Figure 5.1.

5.2 An analysis of ill-conditioning in subspace identification

5.2.1 Reasons for ill-conditioning

Two reasons are often cited in the literature [22, 25] for the occurrence of conditioning problems in combined stochastic-deterministic subspace identification algorithms. The first involves a strong correlation between the rows of the past and future block-Hankel matrices W_p and U_f and centers around the oblique projection which is implicitly or explicitly found in all combined subspace algorithms. Its influence is mostly felt in the estimates for the system matrices A and C. The second relates to stochastic contributions to the outputs which are correlated with the system inputs, and leads to unreliable estimates for B and D. Both cases will be discussed below.

The oblique projection as a source of conditioning problems

In the unifying Theorem 3.4 for combined stochastic-deterministic subspace identification methods it was seen that a key component of these algorithms is the oblique projection of the future outputs Y_f along the future inputs U_f onto the past W_p. Following the notation in Chapter 3, we have:

$$\mathcal{O}_i = Y_f /_{U_f} W_p \underset{j\to\infty}{=} \Gamma_i \widetilde{X}_i,$$

from which estimates for the extended observability matrix Γ_i and a non-steady state Kalman filter sequence $\widetilde{X}_i$ can be obtained. In essence this oblique projection can be understood in a least squares regression framework where matrices $\widehat{L}_1$ and $\widehat{L}_2$ are calculated as:

$$(\widehat{L}_1, \widehat{L}_2) = \arg\min_{L_1, L_2} \left\| Y_f - \begin{bmatrix} L_1 & L_2 \end{bmatrix} \begin{bmatrix} U_f \\ W_p \end{bmatrix} \right\|_F^2 \tag{5.1}$$

and the oblique projection is given as $Y_f /_{U_f} W_p = \widehat{L}_2 W_p$. It is well known that the least squares problem (5.1) is ill-conditioned if the condition number of the regression matrix $\begin{bmatrix} W_p^T & U_f^T \end{bmatrix}^T$ is large. This typically happens when one or more rows in the regression matrix are nearly parallel. An important measure is the smallest principal angle between the space spanned by the rows of W_p and the space spanned by the rows of U_f. Denoting this angle by $\theta_{\min}$ and following the derivation in 2.5 for the condition number of a general oblique projection we have:

$$\mathrm{Cond}_L\left(\mathcal{P}^T_{\{W_p^T | U_f^T\}}\right) = \frac{1}{\sin(\theta_{\min})}. \tag{5.2}$$

As mentioned in the introduction, it is reported in various articles [21, 26, 89] that combined stochastic-deterministic subspace identification algorithms, and especially the N4SID, tend to give bad results if the condition number (5.2) is high, a situation that is graphically illustrated in Figure 5.2. This situation will typically occur when strongly colored inputs are applied to the system. In [35–37] it was proven that there is a strong correlation between the amount of coloring in the input signal u and the principal angles between the row spaces formed by the block Hankel matrices U_p and U_f. More specifically, the principal angles between U_p and U_f are known to decrease with increasing coloredness of

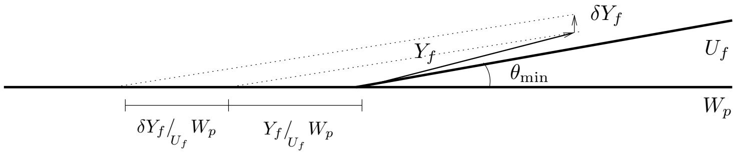


Figure 5.2: Possible ill-conditioning of the oblique projection due to a near parallelism between W_p and U_f. If W_p and U_f are nearly parallel, a small variation on Y_f can have a relatively large effect on the oblique projection in the direction of the principal directions of W_p corresponding to small principal angles.

the input. Furthermore, if the colored inputs are generated by filtering a white noise sequence through a known linear system, the expected principal angles can be calculated exactly if the amount of measurements goes to infinity ($j \to \infty$). From the definition of the principal angle between two spaces in Section 2.4 and the fact that W_p contains the matrix U_p it follows directly that $\theta_{\min}$ will be smaller or equal to the smallest principal angle between U_p and U_f. Hence, a large amount of coloring in the input signal will give rise to a high condition number (5.2) and an unreliable estimate for Γ_i and $\widetilde{X}_i$. This will in turn lead to unreliable system matrix estimates, especially for the matrices A and C [25]. However, even if the principal angles between U_p and U_f are relatively large, the condition number of the oblique projection can still be high since the past data matrix Y_p can in principle also be correlated with U_f. In [26] it is shown that for a given linear system, a set of probing inputs can always be designed which generate data with minimal principal angles between W_p and U_f.

Correlations between stochastic contributions to the outputs and the system inputs

Another possible source of ill-conditioning arises from correlations between stochastic contributions to the outputs and the system inputs. Following the notation from Subsection 3.5.1 and Subsection 3.5.2 it is easily verified that for a combined deterministic system of the form (3.26), we have

$$y_t^d = Du_t + \sum_{i=0}^{\infty} CA^i Bu_{t-i}, \qquad y_t^s = v_t + \sum_{i=0}^{\infty} CA^i w_{t-i}, \qquad \forall t.$$

As v_k and w_k are uncorrelated with u_l for all k, l, the following properties result:

$$E\left\{y_k^s u_l^T\right\} = 0_{l \times m}, \qquad E\left\{y_k^s {y_l^d}^T\right\} = 0_{l \times n}, \qquad \forall k, l. \tag{5.3}$$

Equation (5.3) and especially the fact that the stochastic outputs y_k^s are orthogonal to the system inputs is what ultimately allows to separate the stochastic and deterministic contributions to the outputs in system identification procedures. In combined stochastic-deterministic subspace identification algorithms the separation of the stochastic and the deterministic contributions to the outputs is performed after estimation of the joint state sequences $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ and the extended observability matrix Γ_i using one of the procedures described in Subsections 3.5.6 and 3.5.7, which typically amount to a regression in the system inputs, whereby the residuals take up the role of the process- and measurement noise. When using an unbiased algorithm to estimate the system matrices, the estimated model is known to converge to the true system as the amount of data goes to infinity.

Nevertheless, when dealing with finite amounts of data in an identification context, block Hankel matrices filled with stochastic outputs such as Y_f^s and Y_p^s are usually not entirely uncorrelated with input block Hankel matrices as U_p and U_f. As will be seen in the examples throughout this chapter, this regularly results in stochastic contributions to the measured outputs being incorrectly introduced into the deterministic subsystem. This is known to result in bad estimates for the system matrices B and D and the deterministic transfer function as a whole, specifically in regions of the frequency band where the input power is low [25].

It should be noted that this problem is in essence not limited to subspace identification algorithms. In fact, it is found in one form or another in any linear system identification approach, including non-parametric estimates for the frequency response function (see Example 5.1). In the absence of any known excitation for certain frequency bands, it is impossible to obtain a good estimate for the deterministic transfer function in these frequency bands, as can for instance be observed from the Cramer-Rao bound [26,32,126]. In this case, even weak correlations between the stochastic contribution to the outputs and the system inputs can lead to stochastic resonances being incorrectly attributed to the deterministic subsystem.

Since this thesis is specifically oriented towards subspace system identification algorithms, the discussion in this chapter will mainly focus on the subspace specific first problem, namely the ill-conditioning in the oblique-projection. Nevertheless, in Subsection 5.3.2 it will be shown that by actively removing the stochastic components of the measured data, the estimate for the deterministic subsystem can considerably be improved.

Example 5.1 The example that will be considered here was introduced in [24] as a typical case of ill-conditioning in subspace identification algorithms. The example consists of a deterministic system with transfer function

$$H_d(z) = \frac{(z+0.1-0.8i)(z+0.1+0.8i)(z-0.5)}{(z-0.75-0.55i)(z-0.75+0.55i)(z-0.9)},$$

driven by a colored input which is obtained by filtering a zero mean, unit variance, white Gaussian noise sequence with length 500 through the system

$$H_u(z) = \frac{(z+0.6\pm 0.6i)(z+0.1\pm 0.8i)(z-0.7)}{(z-0.7\pm 0.4i)(z-0.2\pm 0.7i)(z-0.85)}.$$

10% of colored output noise is added, obtained by filtering a zero mean white Gaussian noise sequence through the system

$$H_s(z) = \frac{(z-0.5)(z-0.7)}{(z+0.2-0.6i)(z+0.2+0.6i)}.$$

The transfer functions of $H_d(z)$, $H_u(z)$ and $H_s(z)$ are displayed in Figure 5.3. The input and output spectra, and the experimental frequency response function obtained from the constructed input and output sequence are displayed in Figure 5.4 and reveal possible difficulties with the estimation of the deterministic transfer function in the high frequency regions. Results for the transfer function estimates of two

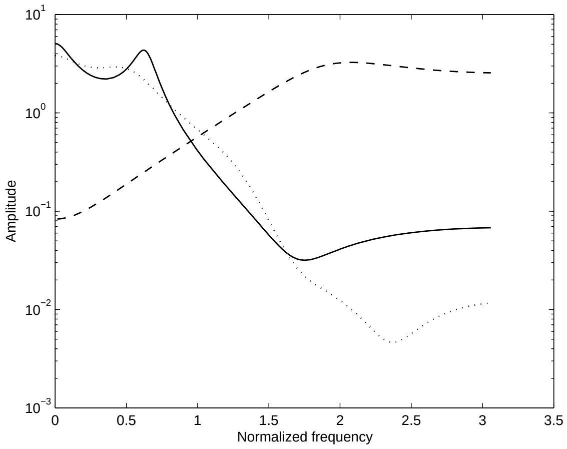


Figure 5.3: Transfer function of the true deterministic subsystem $H_d(z)$ (solid), the stochastic subsystem $H_s(z)$ (dashed) and the system generating the colored input signal $H_u(z)$ (dotted) in Example 5.1.

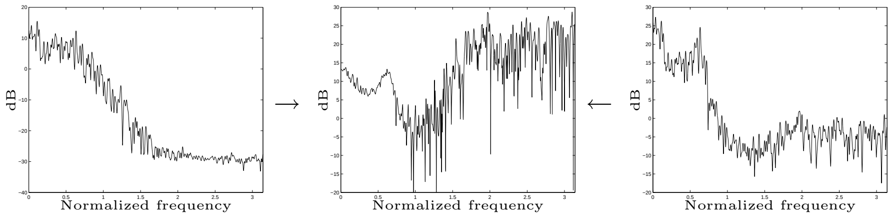


Figure 5.4: Experimental input power spectrum (left), output power spectrum (right) and the frequency response function (middle) in Example 5.1. A naïve calculation of the experimental frequency response function yields large peaks in certain regions of the spectrum were stochastic resonances are found together with small inputs. Although subspace identification methods have an entirely different nature than this naïve calculation, it is known that they too tend to have difficulties in separating the stochastic and the deterministic subsystem, specifically when the inputs are highly colored.

subspace identification algorithms, N4SID and PO-MOESP, performed on these measured input and output samples are displayed in Figure 5.5. During the identification, the order was fixed to 5, and the number of block-rows, i in the Hankel matrices to 10. The algorithm that was used is an unbiased combined state-based algorithm reported in [147]. It is obvious from the figure that both algorithms were unable to yield perfect models. This is due to the presence of a stochastic resonance in the higher frequency regions where virtually no inputs are present, and the strong coloring of the input signal resulting in $\theta_{\min} = 0.98$ degrees as the smallest principal angle between W_p and U_f for the given example. Note however that the PO-MOESP algorithm performs better than the N4SID algorithm. This phenomenon will be discussed in Subsection 5.3.1.

5.2.2 Some parallels with bad conditioning in ARX identification

It was shown in [98, 125] that the problem of ill-conditioning of a regression problem in system identification is not limited to subspace algorithms. In fact, the classical ARX identification approach is known to suffer from problems of a similar nature. The polynomial coefficients $a_k, k = 1, \dots, n$ and $b_l, l = 1, \dots, m$ of a SISO ARX model

$$y_t = \sum_{k=1}^{n} a_k y_{t-k} + \sum_{l=1}^{m} b_l u_{t-l},$$

are often obtained by solving a least squares problem of the following form:

$$(\hat{a}_k, \hat{b}_l) = \arg\min_{a_k, b_l} \left\| Y_{n|n} - \left[\begin{array}{ccc|ccc} a_n & \dots & a_1 & b_m & \dots & b_1 \end{array} \right] \left[\begin{array}{c} Y_{0|n-1} \\ \hline U_{0|m-1} \end{array} \right] \right\|_F^2, \tag{5.4}$$

which is ill-conditioned in case of a near parallelism of rows in the regression matrix. As was seen in [98], this will typically be the case when inputs and/or outputs are limited to a small portion of the frequency band. This can easily be understood from the following lemma.

Lemma 5.1. Condition number and principal angles: *Consider a matrix $A \in \mathbb{R}^{i \times N}$ with $i \leq N$ and two index vectors $I_1 \in \mathbb{R}^{i_1}, I_2 \in \mathbb{R}^{i_2}$ so that $i_1 + i_2 \leq i$ and I_1 and I_2 completely disjunct. Then*

$$Cond(A) \geq \frac{1}{\sin\left(\theta_{\min}\left(A(I_1,:) \sphericalangle A(I_2,:)\right)\right)},$$

where $\theta_{\min}\left(A(I_1,:) \sphericalangle A(I_2,:)\right)$ denotes the smallest principal angle between the row spaces of $A(I_1,:)$ and $A(I_2,:)$.

Proof. See Appendix D.1. □

Taking into account the relations between subspace angles and signal coloring in [35–37], the link between ill-conditioning and coloring of the inputs and/or outputs is now obvious.

The above also serves to highlight differences and similarities between ARX- and subspace-identification algorithms. Both are to a certain extent built

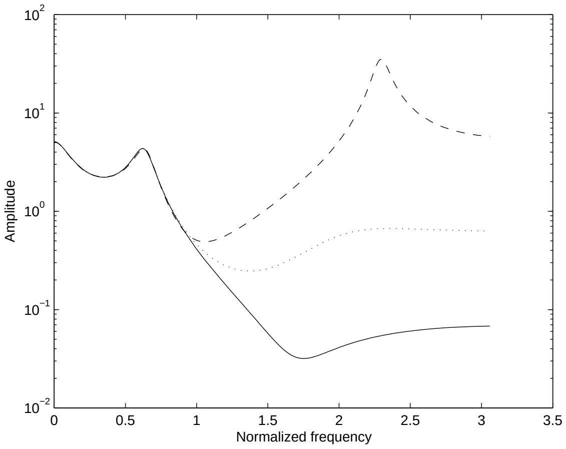


Figure 5.5: Estimated deterministic transfer function of the N4SID model (dashed) and the PO-MOESP model (dotted) compared to the true transfer function (solid). Both combined stochastic-deterministic subspace identification algorithms are seen to suffer from ill-conditioning. The estimate of the N4SID model is particularly bad in the high frequency regions.

around least-squares problems and therefore subject to possible bad conditioning in the regression matrix. On the other hand, in subspace identification the result $\widehat{L}_2$ of the least squares estimate (5.1) is again multiplied with the past data in W_p to obtain $\mathcal{O} = \widehat{L}_2 W_p$. Hence, a possible near parallelism of rows within W_p does not pose any problems. As a result subspace identification algorithms are much more robust to highly colored outputs, but do suffer from coloring in the input-signal.

Example 5.2 We consider the following SISO system:

$$A(z)y = B(z)u + e, \tag{5.5}$$

with A and B polynomials in the forward shift operator z where $B(z) = z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3$ and $A(z) = (z - 0.98e^{\pm i})(z - 0.99e^{\pm 1.6i})(z - 0.97e^{\pm 0.4i})$. A dataset is generated from this system with u and e zero mean white Gaussian noise sequences of length 1000 with standard deviation 2 and 0.1 respectively. ARX identification using the least-squares approach given in (5.4), with $n = 6$ and $m = 4$, was used to obtain estimates for the polynomials $A(z)$ and $B(z)$. The true transfer function together with the estimated one is shown in Figure 5.6.a. In a second experiment, 1% of output noise was added to the outputs and again estimates for $A(z)$ and $B(z)$ are obtained using (5.4). The result is displayed in Figure 5.6.b. Note that the transfer function estimates have been gravely affected by the addition of the small amount of output-noise. The reason for this is found in the condition number of the output regression matrix at the right hand side of (5.4) which turns out to be quite high for this example due to the fact that the output spectrum is concentrated in the lower frequency bands of the spectrum. As a result, the least-squares problem (5.4) is ill-conditioned. An N4SID and a PO-MOESP subspace estimate based on the noisy dataset are displayed in Figures 5.6.c and 5.6.d. The subspace identification algorithm that was used for these estimates is the biased combined state-based algorithm shown in Figure 3.7 with the proper N4SID and PO-MOESP weighting. The order was chosen equal to 6 and the number of block rows equal to 10. From the figure, it is clear that subspace identification algorithms are much more robust against a strong coloring of the output spectrum.

5.3 The orthogonal decomposition method

The two main components of the orthogonal decomposition method as proposed in [21, 26] are the replacement of the oblique projection by an orthogonal projection, and the use of a separately parameterized deterministic and stochastic model. Both changes will be discussed in Subsections 5.3.1 and 5.3.2, respectively. In Subsection 5.3.2 it will also be seen that the orthogonal decomposition method is very closely related to the existing PI-MOESP algorithm described in Section 3.7.

5.3.1 Replacement of the oblique projection by an orthogonal projection

It was seen in Section 5.2 that under the presence of highly colored inputs, the oblique projection, which is implicitly or explicitly present in combined subspace identification algorithms is potentially ill-conditioned. However, it has been known for some time [1,52] that the quality of combined subspace estimates is strongly influenced by the choice of the weighting matrices W_1 and W_2 in the unifying Theorem 3.4. In general, and especially under the presence of colored inputs, algorithms employing the PO-MOESP or CVA weighting scheme are known to outperform the N4SID on a large variety of examples [52, 147]. This is largely due to the right multiplication of the oblique projection by the matrix $W_2 = \Pi_{U_f{}^\perp}$ which removes the correlations with U_f from the past data W_p and essentially replaces the oblique projection $Y_f /_{U_f} W_p$ by an orthogonal projection $Y_f/(W_p/U_f^\perp)$ as also seen in Table 3.1. The orthogonal decomposition method as proposed in [26] employs a PO-MOESP weighting scheme (see also Figure 5.1).

As will be shown in Example 5.3 the replacement of the oblique projection by an orthogonal projection leads to a considerable improvement in the estimates

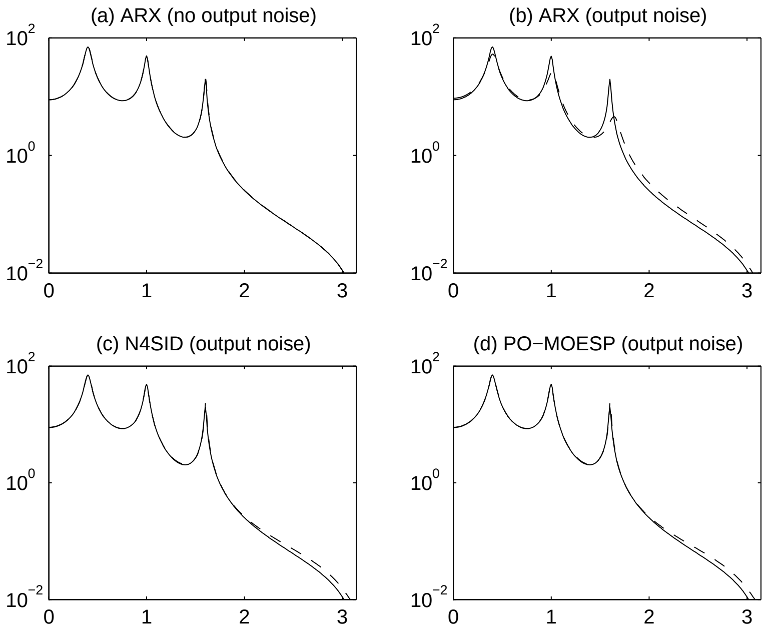


Figure 5.6: True (solid) and estimated (dashed) transfer functions for Example 5.2 using different identification methods and different amounts of output noise. In sub-figure (a), ARX was used on a dataset without output noise. In sub-figure (b), ARX was used on a dataset with 1% of output noise. In subfigures (c) and (d), N4SID and PO-MOESP were respectively used on the same dataset with 1% of output noise. The ARX method is seen to be very sensitive to the output noise due to the concentration of the output signal in the lower frequency range. Subspace methods are seen to be more robust against this type of output induced ill-conditioning.

of the system matrices A and C. In [145, 147] the superior behavior of the PO-MOESP and CVA algorithm was explained by noting that right multiplication of the oblique projection by $W_2 = \Pi_{U_f^\perp}$ amounts to a form of frequency weighted balancing where the highest weight is given to those regions in the frequency spectrum where the input power is high. Nevertheless, the replacement of the oblique projection by an orthogonal projection does not solve the problem of stochastic components of the system which are incorrectly attributed to the inputs. Hence, even when using an orthogonal projection, the estimates of B and D are still known to be bad in certain experimental conditions.

Example 5.3 The presence of a stochastic resonance, combined with a large amount of coloring in the inputs, was seen to lead to ill-conditioning in the earlier discussed Example 5.1. Following up on this example, we generate 100 datasets with the same statistics as the dataset used to obtain the results for Example 5.1. For each of these datasets, the poles of the deterministic subsystem were estimated from the fifth order extended observability matrix Γ_i following from the projection $Y_f \big/_{U_f} W_p \underset{j\to\infty}{=} \Gamma_i \widetilde{X}_i$ and $Y_f/(W_p/U_f^{\perp}) \underset{j\to\infty}{=} \Gamma_i \widetilde{X}_i/U_f^{\perp}$ respectively. The results for both cases, which correspond to the N4SID and the PO-MOESP algorithm, are shown in Figure 5.7. Clearly, the replacement of the oblique projection $Y_f \big/_{U_f} W_p$ by an orthogonal projection $Y_f/(W_p/U_f^{\perp})$ leads to much better estimates for the extended observability matrix, and consequently the system poles. This largely explains the better overall performance for the PO-MOESP algorithm in Example 5.1. Nevertheless, as was seen in the latter example, even the PO-MOESP algorithm yields a suboptimal estimate for the transfer function in the high frequency range due to the occurrence of stochastic resonances in regions of the frequency spectrum where the input power is low. Although the estimates for A and C using the PO-MOESP algorithm are reliable, these resonances result in bad estimates for the matrices B and D. In Subsection 5.3.2, it will be seen that this remaining problem can be dealt with using separately parameterized model structures.

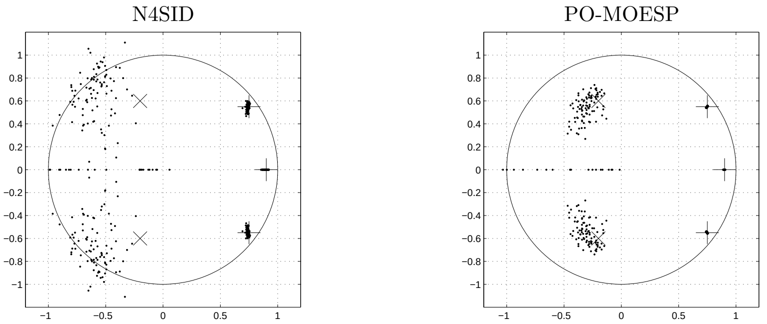


Figure 5.7: Estimates N4SID and PO-MOESP poles (dots) for 100 trials using datasets generated according to Example 5.1. The true poles are displayed using a large '+' for poles of the deterministic subsystem and a 'X' for poles of the stochastic subsystem. It is clear that the PO-MOESP approach yields far better estimates for the system poles due to the replacement of the oblique projection by an orthogonal projection.

5.3.2 A separately parameterized model structure

Although most known subspace approaches are of the combined stochastic-deterministic type, recently some renewed interest has emerged in subspace identification algorithms that identify separately parameterized models (see Section 3.7 for a description of this type of model structure). Arguments in favor of the use of separately parameterized models largely center around the claim that in most practical cases the stochastic and deterministic system have little common dynamics anyway [26]. Furthermore, by decoupling the identification of the deterministic and the stochastic subsystem, different state-space bases can be chosen for the deterministic and the stochastic state. One could for instance choose to identify the deterministic system in a deterministically balanced basis and the stochastic system in a stochastically balanced one. A last advantage of the decoupling of the identification of the deterministic and the stochastic system is an increased robustness against stochastic system components ending up in the deterministic model as will be seen below.

Practical implementation

The first step in generating seperately parameterized models is the decoupling of the future output matrix Y_f in a deterministic and a stochastic component, which is performed as follows:

$$Y_f^d = Y_f / \begin{bmatrix} U_p \\ U_f \end{bmatrix}, \quad Y_f^s = Y_f / \begin{bmatrix} U_p \\ U_f \end{bmatrix}^{\perp}.$$

A seperate identification procedure is thereafter used for the deterministic subsystem and the stochastic subsystem. As we will mainly be interested in the performance of the deterministic model, we will limit ourselves in this chapter to the analysis of Y_f^d. For further reading on the identification of the stochastic subsystem we refer the reader to [21,26]. When using the PO-MOESP weighting scheme as it is present in the orthogonal decomposition method, estimates for the extended observability matrix and the state of the deterministic subsystem are obtained as follows:

$$Y_f^d/(W_p/U_f^{\perp}) \underset{j\to\infty}{=} \Gamma_i \widetilde{X}_i / U_f^{\perp}, \tag{5.6}$$

where-after A, B, C and D are estimated using one of the algorithms presented in Subsection 3.5.6.

Relation to the PI-MOESP algorithm

As reported in [26], the projection (5.6) is nothing else than the basic projection behind the PI-MOESP identification algorithm introduced in Section 3.7. This follows immediately from the following lemma.

Lemma 5.2. *With* $U = \begin{bmatrix} U_p^T & U_f^T \end{bmatrix}^T$ *and following the basic notation for subspace block Hankel matrices introduced in Chapter 3, the following relations hold:*

$$Y_f^d/(W_p/U_f^\perp) = Y_f/(U_p/U_f^\perp), \tag{5.7}$$

$$Y_f^d/(W_p/U_f^\perp) = Y_f/(W_p/U_f^\perp)/U. \tag{5.8}$$

Proof. See appendix D.2. □

Equation (5.8) offers another view on the PI-MOESP algorithm. Namely that of a PO-MOESP algorithm where the result of the orthogonal projection is again projected on the system inputs. Note that this removes most of the stochastic contributions from $\mathcal{O}_i$. Stochastic contributions which survive the projection due to weak correlations with the system inputs (see Subsection 5.2.1) are removed in the SVD step where the system order is chosen equal to the order of the deterministic subsystem instead of that of the combined stochastic-deterministic system. This is also seen in Example 5.4.

Example 5.4 The PI-MOESP method with $n = 3$ and $i = 10$ is applied to the dataset that was used in Example 5.1. The resulting transfer function estimate together with that of the N4SID and the PO-MOESP method is displayed in Figure 5.8. Where the PO-MOESP was earlier seen to yield better results than the N4SID due to the replacement of the oblique

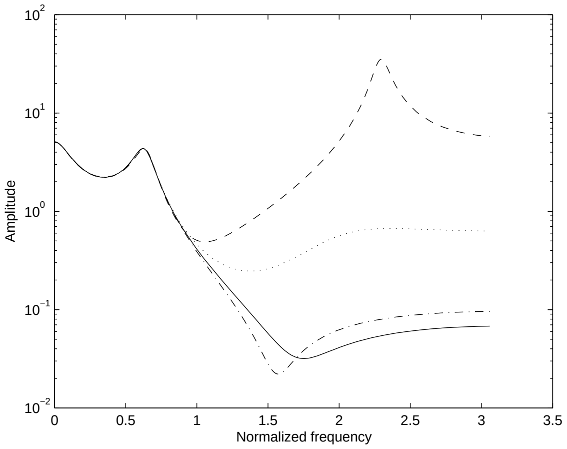


Figure 5.8: Estimated deterministic transfer function of the N4SID model (dashed), the PO-MOESP model (dotted) and the PI-MOESP model (dash-dotted) compared to the true transfer function (solid). The PI-MOESP method clearly outperforms the N4SID and the PO-MOESP method on this dataset due to the use of a separate parameterization.

> projection by an orthogonal projection, using a separate parameterization as is done in the PI-MOESP clearly leads to a further improvement in the estimate of the transfer function on this dataset. A full statistical analysis of the performance of different subspace identification methods using Monte-Carlo simulations will be presented in Section 5.5.

In Example 5.4 it is observed that the PI-MOESP algorithm leads to a much better estimate for the deterministic component of the system introduced in Example 5.1, where a highly colored input was present. Other examples reported in [26] yield similar results. Furthermore, an asymptotic variance analysis performed in [26] theoretically confirms that in the case of highly colored inputs, the PI-MOESP algorithm tends to outperform combined approaches. and that the difference with respect to combined approaches with an orthogonal projection (such as the PO-MOESP) is largely found in the accuracy of the estimates for B and D.

Note that the PI-MOESP algorithm does not fit in the unifying Theorem 3.4, since the projection on U does not necessarily conserve the order of the system.

5.4 A new algorithm based on regularization

In Section 5.2 it was seen that the N4SID method performs very badly in the presence of highly colored inputs. This is mostly due to strong correlations between the past W_p and the future U_f, leading to an ill-conditioned oblique projection $Y_f /_{U_f} W_p$. In Subsection 5.3.1 it was argued that by removing the influence of the future inputs in U_f from the past by changing the oblique projection in an orthogonal projection, the conditioning of subspace identification algorithms, and especially the estimation accuracy of A and C can be improved. In this section it will be seen that instead of the somewhat drastic removal of the entire influence of the future inputs from W_p, combining the oblique projection with a more subtle regularization approach leads to estimates for A and C of the same or better quality as those obtained using the PO-MOESP. The result of the oblique projection can thereafter be projected on the inputs to avoid stochastic contributions ending up in the state and obtain reliable estimates for B and D. The algorithm so obtained is a possible alternative to the PI-MOESP algorithm. Its performance will be studied in Section 5.5.

5.4.1 A regularized oblique projection

The oblique projection in the regularized N4SID algorithm is obtained by solving the following regularized least-squares problem:

$$(\widehat{L}_1^\gamma, \widehat{L}_2^\gamma) = \arg\min_{L_1,L_2} \left(\left\| Y_f - \begin{bmatrix} L_1 & L_2 \end{bmatrix} \begin{bmatrix} U_f \\ W_p \end{bmatrix} \right\|_F^2 + \gamma \|L_2 W_p\|_F^2 \right),$$

with γ a positive regularization constant. The regularization term $\|L_2 W_p\|_F^2$ is a non-increasing function of γ and serves to keep the norm of the obtained projection low. As will be seen shortly, the regularization term $\|L_2 W_p\|_F^2$ will mainly influence the result of the projection along the principal directons of Row(W_p) corresponding to small principal angles between U_f and W_p. This reduces the variance on the obtained estimates of Γ_i and $\widetilde{X}_i$ at the expense of the introduction of a small bias. The regularized oblique projection $\mathcal{O}_i$ is found as

$$\begin{aligned} \begin{bmatrix} \widehat{L}_1^\gamma & \widehat{L}_2^\gamma \end{bmatrix} &= Y_f \begin{bmatrix} U_f^T & W_p^T \end{bmatrix} \left(\begin{bmatrix} U_f U_f^T & U_f W_p^T \\ W_p U_f^T & (1+\gamma) W_p W_p^T \end{bmatrix} \right)^{-1}, \\ \mathcal{O}_i &= \widehat{L}_2^\gamma W_p. \end{aligned} \tag{5.9}$$

Noticing that a change in basis in W_p and U_f will not influence the calculation (5.4.1), we assume without loss of generality that W_p and U_f are formed by a set of orthonormal basis vectors for Row(U_f) and Row(W_p) such that

$$U_f U_f^T = I_{im}, \;\; W_p W_p^T = I_{i(m+l)}, \;\; U_f W_p^T = \begin{bmatrix} \Lambda & 0_{im \times il} \end{bmatrix},$$

with $\Lambda \in \mathbb{R}^{im}$ a diagonal matrix containing the cosines of the principal angles between W_p and U_f. With this choice of basis, (5.9) can be rewritten as:

$$\begin{aligned} \begin{bmatrix} \widehat{L}_1^\gamma & \widehat{L}_2^\gamma \end{bmatrix} &= Y_f \left[\begin{array}{c|c} U_f^T & W_p^T \end{array} \right] \left[\begin{array}{c|cc} I_{im} & \Lambda & 0 \\ \hline \Lambda & (1+\gamma) I_{im} & 0 \\ 0 & 0 & (1+\gamma) I_{il} \end{array} \right]^{-1}, \\ \mathcal{O}_i &= \widehat{L}_2^\gamma W_p. \end{aligned}$$

Furthermore, for any row t in L_1^γ and L_2^γ, we have:

$$\begin{aligned} \|\widehat{L}_1^\gamma(t,:)\|_F &= \|\widehat{L}_1^\gamma(t,:) U_f\|_F, \\ \|\widehat{L}_2^\gamma(t,:)\|_F &= \|\widehat{L}_2^\gamma(t,:) W_p\|_F, \end{aligned}$$

which enables us to study the properties of the regularized oblique projection through an analysis of the estimates $\widehat{L}_1^\gamma$ and $\widehat{L}_2^\gamma$.

5.4.2 Influence on the obtained projection

It is instructive to study the influence of the introduction of the regularization term $\|L_2 W_p\|_F^2$ on the obtained parameters $\widehat{L}_1^\gamma$ and $\widehat{L}_2^\gamma$. This is done in the following lemma:

Lemma 5.3 (Influence of regularization on $\widehat{L}_1^\gamma$ and $\widehat{L}_2^\gamma$). *Adopting the working assumption of exact knowledge of Y_p and U_f and a zero mean temporary and stationary white noise perturbation δY_f on the data Y_f such that $E\left\{(\delta Y_f)^T \delta Y_f\right\} = \sigma_y^2 I_j$, and defining*

$$\begin{bmatrix} \delta\widehat{L}_1^\gamma & \delta\widehat{L}_2^\gamma \end{bmatrix} = \delta Y_f \left[\begin{array}{c|c} U_f^T & W_p^T \end{array} \right] \left(\left[\begin{array}{c|cc} I_{im} & \Lambda & 0 \\ \hline \Lambda & (1+\gamma) I_{im} & 0 \\ 0 & 0 & (1+\gamma) I_{il} \end{array} \right] \right)^{-1}, \tag{5.10}$$

we have for any $t = 1, \ldots, il$*:*

$$E\left\{\left(\left[\delta\widehat{L}_1^\gamma(t,:) \quad \delta\widehat{L}_2^\gamma(t,:)\right]\right)^T \left(\left[\delta\widehat{L}_1^\gamma(t,:) \quad \delta\widehat{L}_2^\gamma(t,:)\right]\right)\right\}$$

$$= \sigma_y^2 \begin{bmatrix} \frac{(1+\gamma)^2 I_{im} - (1+2\gamma)\Lambda^2}{((1+\gamma)I_{im}-\Lambda^2)^2} & \frac{\Lambda^3-\Lambda}{((1+\gamma)I_{im}-\Lambda^2)^2} & 0 \\ \frac{\Lambda^3-\Lambda}{((1+\gamma)I_{im}-\Lambda^2)^2} & \frac{I_{im}-\Lambda^2}{((1+\gamma)I_{im}-\Lambda^2)^2} & 0 \\ 0 & 0 & \frac{1}{(1+\gamma)^2} I_{il} \end{bmatrix}, \tag{5.11}$$

Proof. See Appendix D.3. □

Under the assumptions stipulated in Lemma 5.3, it is easily seen that

$$E\left\{\left(\delta\widehat{L}_2^\gamma(t,:)\right)^T \left(\delta\widehat{L}_2^\gamma(t,:)\right)\right\} = \sigma_y^2 \begin{bmatrix} \frac{I_{im}-\Lambda^2}{((1+\gamma)I_{im}-\Lambda^2)^2} & 0 \\ 0 & \frac{1}{(1+\gamma)^2} I_{il} \end{bmatrix}. \tag{5.12}$$

Since we are only interested in the row-space of the oblique projection, (5.12) can be scaled by a factor $(1+\gamma)^2$ without affecting the qualitative interpretation of the results. We have

$$(1+\gamma)^2 E\left\{\left(\delta\widehat{L}_2^\gamma(t,:)\right)^T \left(\delta\widehat{L}_2^\gamma(t,:)\right)\right\} = \begin{bmatrix} \frac{I_{im}-\Lambda^2}{\left(I_{im}-\frac{\Lambda^2}{1+\gamma}\right)^2} & 0 \\ 0 & I_{il} \end{bmatrix}, \tag{5.13}$$

It follows directly from (5.13) that

$$(1+\gamma)^2 \sqrt{E\left\{\left(\delta\widehat{L}_2^\gamma(t,k)\right)^T \left(\delta\widehat{L}_2^\gamma(t,k)\right)\right\}} = \sqrt{\frac{1-\cos^2(\theta_k)}{\left(1-\frac{\cos^2(\theta_k)}{1+\gamma}\right)^2}}, \tag{5.14}$$

which decreases with increasing γ, especially when θ_k is small. This is also seen in Figure 5.9, where (5.14) is depicted as a function of γ and θ_k. Hence, based on this analysis we can conclude that using regularization allows to reduce the variance on the obtained estimate for the oblique projection in the N4SID algorithm. However, we remind the reader that the variance analysis of the regularized oblique projection are only valid under the assumption of temporary and stationary white noise on the data in Y_f and exact knowledge of W_p and U_f. The same goes for the definition of the unregularized condition number (2.15), which is derived from a similar assumption. Although for U_f this assumption is reasonable, perturbations on Y_p need to be taken into account in principle. This is certainly true given the fact that the statistical properties of any perturbation on Y_p are in essence the same as those of perturbations on Y_f.

A variance analysis for the oblique projection with perturbations in Y_p could be derived, e.g. starting from expressions used in the derivation of a condition number for least squares estimators with errors in the variables such as reported in [46, 76, 133, 140, 141, 158]. However, the obtained expressions would be significantly more complicated than the ones so far obtained. This

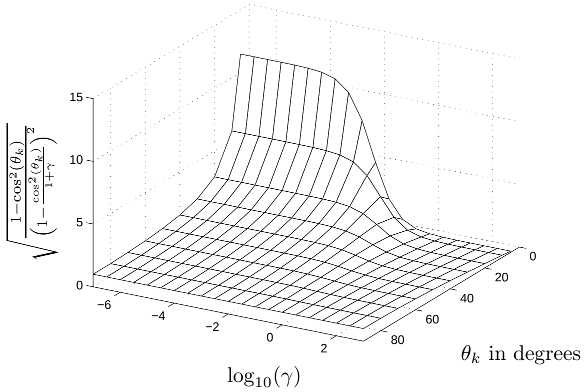


Figure 5.9: Graphical interpretation of equation (5.14) as a function of $\cos(\theta_k)$ and γ.

while the simple expression (5.14) has the advantage that the expected variance on a given estimate $L_2^{\gamma}(t,k)$ is determined by exactly one principal angle, which enables a quick interpretation of the effect of the regularization term on a given component.

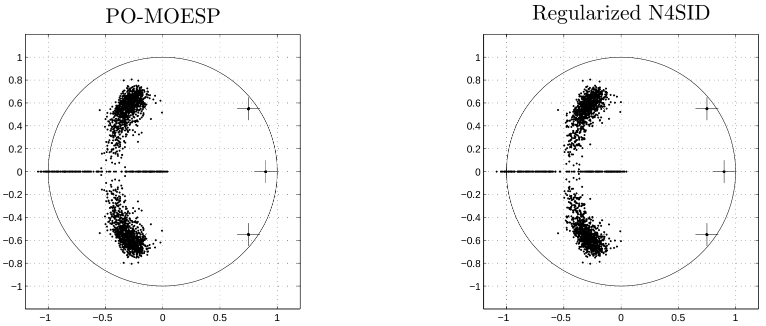
In the following section, the performance of the regularized N4SID algorithm and the decrease in the variance of the estimates will be evaluated using a set of Monte-Carlo simulations.

5.5 Performance of the regularized N4SID

5.5.1 Influence of regularization

We study the performance of the regularized N4SID algorithm using a Monte-Carlo analysis on the system and data introduced in Example 5.1 and further analyzed in Examples 5.3 and 5.4. For the entire analysis, subspace estimates will be obtained using $n = 5$, and $i = 10$.

In a first step, 1000 datasets are generated with the same statistics as the datasets used to obtain the results of Example 5.1. The estimated system poles (deterministic and stochastic) using a PO-MOESP algorithm and the regularized N4SID algorithm are displayed in Figure 5.10. The regularization constant γ was chosen equal to 10, a choice which will be justified shortly. Note that the pole estimates using N4SID have significantly been improved with respect to the unregularized pole-estimates in Figure 5.7, to the extent that hardly any difference in performance with respect to the PO-MOESP is seen in Figure 5.10. A more quantitative comparison of the accuracy of the pole-estimates using both techniques is found in Table 5.1, where the euclidean distances between the estimated and the true poles averaged over the 1000


PO-MOESP

Regularized N4SID

Figure 5.10: Estimated poles (dots) using PO-MOESP and regularized N4SID ($\gamma = 10$) for 1000 trials using datasets generated according to Example 5.1. The true poles are displayed using a large '+' for poles of the deterministic subsystem and an 'X' for poles of the stochastic subsystem. In the latter case, the 'X' is hidden by the actual estimates. Note that the deterministic poles are perfectly estimated using both techniques. The variance on the estimates for the stochastic poles is high, but comparable for both techniques. A more quantitative comparison is given in Table 5.2.

datasets are given for the classical N4SID, the PO-MOESP, and the regularized N4SID. From the table it is clear that the regularization approach yields slightly better pole-estimates than the PO-MOESP.

Poles	$0.75 \pm 0.55i$	0.9	$-0.2 \pm 0.6i$
Mean(d_{N4SID})	0.0071	0.0043	0.4680
Std(d_{N4SID})	0.0042	0.0028	0.1180
Mean(d_{MOESP})	0.0014	0.0009	0.2108
Std(d_{MOESP})	0.0009	0.0007	0.1700
Mean(d_{REG})	0.0011	0.0006	0.2100
Std(d_{REG})	0.0007	0.0005	0.1770

Table 5.1: Euclidean distances between the estimated and the true poles averaged over 1000 datasets generated according to Example 5.1 for the classical N4SID (d_{N4SID}), the PO-MOESP (d_{MOESP}), and the regularized N4SID (d_{REG}) with $\gamma = 10$.

In a second step, we study the effect on the estimated transfer function. Defining d_2 and d_∞ as the 2-norm and infinity-norm of the difference between the estimated and the true transfer function, the mean and standard deviations on d_2 and d_∞ over all 1000 datasets, and for the classical N4SID, the PO-MOESP and the regularized N4SID with various regularization constants are given in Table 5.2. From the table it is seen that regularization significantly improves

	# Stable	Mean(d_2)	Std(d_2)	Mean(d_∞)	Std(d_∞)
N4SID	772	3.272	2.1294	16.97	65.44
PO-MOESP	987	0.4154	0.4351	1.127	6.305
REG ($\gamma = 0.001$)	777	3.134	1.931	14.07	30.50
REG ($\gamma = 0.01$)	984	0.7009	0.6761	1.785	4.185
REG ($\gamma = 0.1$)	986	0.4677	0.4119	1.037	1.674
REG ($\gamma = 1$)	986	0.3667	0.3976	0.9588	2.779
REG ($\gamma = 10$)	986	0.3378	0.3876	0.8766	2.437
REG ($\gamma = 100$)	986	0.3343	0.3855	0.8670	2.396
REG ($\gamma = 500$)	986	0.3340	0.3852	0.8661	2.393
REG ($\gamma = 1000$)	986	0.3339	0.3852	0.8660	2.392
REG ($\gamma = 1500$)	986	0.3339	0.3852	0.8659	2.392
REG ($\gamma = 3000$)	986	0.3339	0.3852	0.8659	2.392
REG ($\gamma = 10000$)	986	0.3339	0.3852	0.8659	2.392

Table 5.2: Mean and standard deviation of the distances between the true and the estimated transfer functions over all 1000 datasets, and this for the classical N4SID, the PO-MOESP and the regularized N4SID with various regularization constants. Distances were only averaged over stable models. The number of stable models is displayed in the table.

the quality of the N4SID subspace estimate. Furthermore, the regularized N4SID is seen to outperform the PO-MOESP for a wide range of regularization parameters. The table also provides justification for the choice of $\gamma = 10$ which was earlier made when reporting the performance on the pole-estimates, although any γ in the range $1 \ldots 10000$ would have been appropriate. We note that in practical applications, comparison with the true system is not possible and γ needs to be tuned, for instance using validation on an independent dataset. The average estimated transfer function and a 95% error region for the three techniques mentioned in Table 5.2 are displayed in Figure 5.11. Again, the PO-MOESP and the regularized N4SID are seen to outperform the classical N4SID, with a slight advantage of the regularized N4SID over the PO-MOESP. Nevertheless, as was also seen in Example 5.4, in the high frequency regions, the estimates for the transfer function are still far from optimal. This issue will be dealt with in the next section, where we will use a projection on the input space in line with Subsection 5.3.2.

5.5.2 Projection on the input-space

In Subsection 5.3.2 it was seen that in the case of highly colored inputs, a separate parameterization of the deterministic and the stochastic subsystem can be preferable. Such a separate parameterization was obtained by projecting $\mathcal{O}_i$ orthogonally onto $\begin{bmatrix} U_p^T & U_f^T \end{bmatrix}^T$ before calculating Γ_i and $\widetilde{X}_i$. The same approach is followed to obtain separately parameterized estimates for the classical N4SID and the regularized N4SID. Denoting the classical N4SID and the regularized

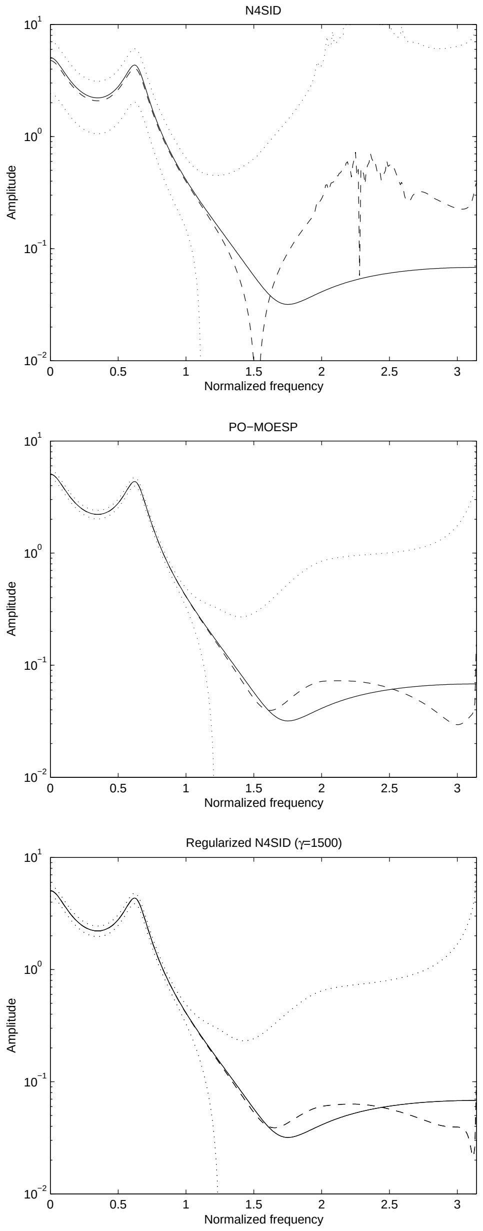


Figure 5.11: Average transfer function (dashed) for the N4SID, the PO-MOESP, and the regularized N4SID ($\gamma = 10$) estimates with 95% error region (dotted). The solid line is the transfer function of the deterministic subsystem.

N4SID, both with projection on the inputs, by PI-N4SID and PI-REG, the resulting transfer functions and the difference between the estimated and the true transfer functions are displayed in Table 5.3 and Figure 5.12. It is important to note here that the estimation order in all subspace algorithms was set to $n = 3$, the dimension of the deterministic subsystem. Note from the table and the figure that the transfer function estimate of the PO-MOESP and the regularized N4SID has significantly improved by projecting onto the input space. For the classical N4SID, such a projection does not lead to any meaningful results.

A surprising observation in Table 5.3 is that the regularized algorithm performs very well for high regularization constants, even for $\gamma = 10000$, in which case one would expect all dynamics to be destroyed. This phenomenon will be discussed in 5.6, where it will be seen that the term $\gamma\|L_2W_p\|_F^2$ with $\gamma \to \infty$ corresponds to a relatively moderate weighted regularization on a certain subspace of W_p. Hence, even for $\gamma \to \infty$, the dynamics of the estimated model will not be completely eliminated. In this sense, regularization on the result of an oblique projection exhibits a completely different behavior as classical Tikhonov regularization for least-squares problems where the dynamics are indeed destroyed for $\gamma \to \infty$.

5.6 Weighted regularization

We prove that the regularization term $\gamma\|L_2W_p\|_F^2$ corresponds to a weighted regularization on a certain subspace of W_p.

Lemma 5.4. *Assume that U_f and W_p are formed by orthonormal basisses for Row(U_f) and Row(W_p) such that*

$$U_fU_f^T = I_{im}, \;\; W_pW_p^T = I_{i(m+l)}, \;\; U_fW_p^T = \begin{bmatrix}\Lambda & 0_{im\times il}\end{bmatrix},$$

with $\Lambda \in \mathbb{R}^{im}$ a diagonal matrix containing the cosines of the principal angles between W_p and U_f. Under these assumptions, the spaces spanned by the rows and columns of $\mathcal{O}_i$, obtained from

$$\begin{aligned}(\widehat{L}_1^\gamma, \widehat{L}_2^\gamma) &= \arg\min_{L_1,L_2}\left(\left\|Y_f - \begin{bmatrix}L_1 & L_2\end{bmatrix}\begin{bmatrix}U_f\\ W_p\end{bmatrix}\right\|_F^2 + \gamma\|L_2W_p\|_F^2\right),\\ \mathcal{O}_i &= \widehat{L}_2^\gamma W_p,\end{aligned}$$

are equal to those spanned by the rows and columns of $\mathcal{P}_i$, obtained from

$$\begin{aligned}(\widetilde{L}_1, \widetilde{L}_2) &= \arg\min_{L_1,L_2}\left(\left\|Y_f - \begin{bmatrix}L_1 & L_2\end{bmatrix}\begin{bmatrix}U_f\\ W_p\end{bmatrix}\right\|_F^2 + \tilde{\gamma}\|L_2SW_p\|_F^2\right),\\ \mathcal{P}_i &= \widetilde{L}_2W_p,\end{aligned}$$

with $\tilde{\gamma} = \frac{\gamma}{1+\gamma}$ and $S = \begin{bmatrix}\Lambda & 0\\ 0 & 0\end{bmatrix}$.

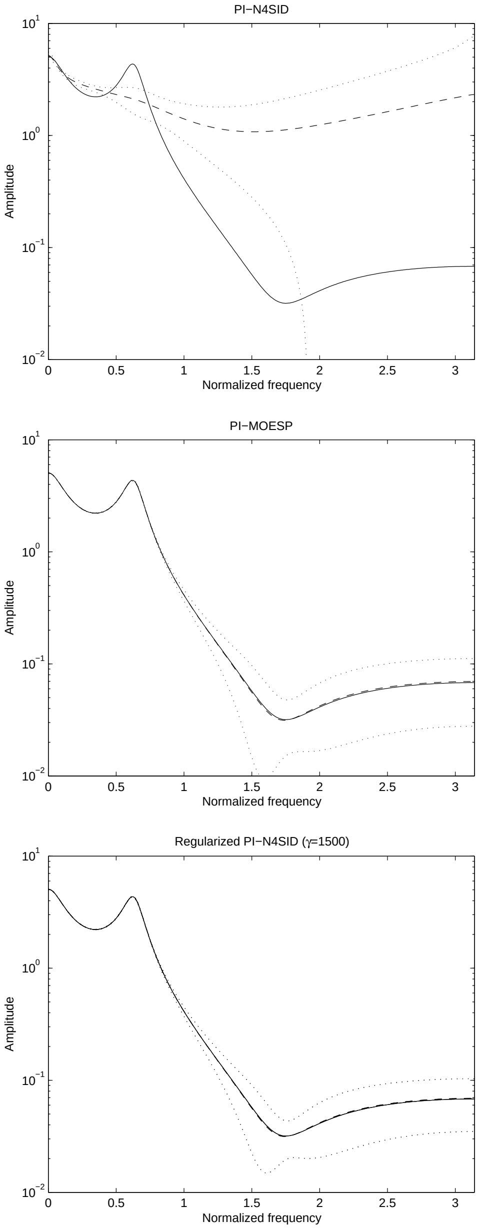


Figure 5.12: Average transfer function (dashed) for the PI-N4SID, the PI-MOESP, and the regularized PI-N4SID ($\gamma = 10$) estimates with 95% error region (dotted). The solid line is the transfer function of the deterministic subsystem.

	Mean(d_2)	Std(d_2)	Mean(d_∞)	Std(d_∞)
PI-N4SID	1.638	2.129	3.120	2.3776
PI-MOESP	0.0206	0.0141	0.0481	0.0311
PI-REG ($\gamma = 0.001$)	0.2375	0.2499	0.4975	0.4327
PI-REG ($\gamma = 0.01$)	0.0830	0.0595	0.1776	0.1215
PI-REG ($\gamma = 0.1$)	0.0217	0.0147	0.0486	0.0303
PI-REG ($\gamma = 1$)	0.0182	0.0119	0.0428	0.0263
PI-REG ($\gamma = 10$)	0.0173	0.0115	0.0420	0.0261
PI-REG ($\gamma = 100$)	0.0173	0.0115	0.0421	0.0262
PI-REG ($\gamma = 500$)	0.0173	0.0115	0.0421	0.0262
PI-REG ($\gamma = 1000$)	0.0173	0.0115	0.0421	0.0262
PI-REG ($\gamma = 1500$)	0.0173	0.0115	0.0421	0.0262
PI-REG ($\gamma = 3000$)	0.0173	0.0115	0.0421	0.0262
PI-REG ($\gamma = 10000$)	0.0173	0.0115	0.0421	0.0262

Table 5.3: Mean and standard deviation of the distances between the true and the estimated transfer functions over all 1000 datasets, and this for PI-N4SID, the PI-MOESP and the regularized PI-N4SID with various regularization constants.

Proof. With $\widehat{L}_1^0$ and $\widehat{L}_2^0$ the unregularized oblique projection estimates ($\gamma = 0$), we have (see 5.10):

$$\begin{aligned}
\begin{bmatrix} \widehat{L}_1^\gamma & \widehat{L}_2^\gamma \end{bmatrix} &= \begin{bmatrix} \widehat{L}_1^0 & \widehat{L}_2^0 \end{bmatrix} \begin{bmatrix} I_{im} & \Lambda & 0 \\ \Lambda & I_{im} & 0 \\ 0 & 0 & I_{il} \end{bmatrix} \begin{bmatrix} I & \Lambda & 0 \\ \Lambda & (1+\gamma)I & 0 \\ 0 & 0 & (1+\gamma)I \end{bmatrix}^{-1} \\
&= \begin{bmatrix} \widehat{L}_1^0 & \widehat{L}_2^0 \end{bmatrix} \begin{bmatrix} I_{im} & \Lambda & 0 \\ \Lambda & I_{im} & 0 \\ 0 & 0 & I_{il} \end{bmatrix} \begin{bmatrix} \frac{(1+\gamma)I}{(1+\gamma)I-\Lambda^2} & \frac{-\Lambda}{(1+\gamma)I-\Lambda^2} & 0 \\ \frac{-\Lambda}{(1+\gamma)I-\Lambda^2} & \frac{I}{(1+\gamma)I-\Lambda^2} & 0 \\ 0 & 0 & \frac{1}{1+\gamma}I_{il} \end{bmatrix} \\
&= \begin{bmatrix} \widehat{L}_1^0 & \widehat{L}_2^0 \end{bmatrix} \begin{bmatrix} I_{im} & 0 & 0 \\ \frac{\gamma\Lambda}{(1+\gamma)I_{im}-\Lambda^2} & \frac{1-\Lambda^2}{(1+\gamma)I_{im}-\Lambda^2} & 0 \\ 0 & 0 & \frac{1}{1+\gamma}I_{il} \end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
\begin{bmatrix} \widetilde{L}_1 & \widetilde{L}_2 \end{bmatrix} &= \begin{bmatrix} \widehat{L}_1^0 & \widehat{L}_2^0 \end{bmatrix} \begin{bmatrix} I_{im} & \Lambda & 0 \\ \Lambda & I_{im} & 0 \\ 0 & 0 & I_{il} \end{bmatrix} \begin{bmatrix} I_{im} & \Lambda & 0 \\ \Lambda & I_{im} + \tilde{\gamma}\Lambda^2 & 0 \\ 0 & 0 & I_{il} \end{bmatrix}^{-1} \\
&= \begin{bmatrix} \widehat{L}_1^0 & \widehat{L}_2^0 \end{bmatrix} \begin{bmatrix} I_{im} & \Lambda & 0 \\ \Lambda & I_{im} & 0 \\ 0 & 0 & I_{il} \end{bmatrix} \begin{bmatrix} \frac{I_{im}+\tilde{\gamma}\Lambda^2}{I_{im}+\tilde{\gamma}\Lambda^2-\Lambda^2} & \frac{-\Lambda}{I_{im}+\tilde{\gamma}\Lambda^2-\Lambda^2} & 0 \\ \frac{-\Lambda}{I_{im}+\tilde{\gamma}\Lambda^2-\Lambda^2} & \frac{I_{im}}{I_{im}+\tilde{\gamma}\Lambda^2-\Lambda^2} & 0 \\ 0 & 0 & I_{il} \end{bmatrix} \\
&= \begin{bmatrix} \widehat{L}_1^0 & \widehat{L}_2^0 \end{bmatrix} \begin{bmatrix} I_{im} & 0 & 0 \\ \frac{\tilde{\gamma}\Lambda^3}{I_{im}+\tilde{\gamma}\Lambda^2-\Lambda^2} & \frac{I_{im}-\Lambda^2}{I_{im}+\tilde{\gamma}\Lambda^2-\Lambda^2} & 0 \\ 0 & 0 & I_{il} \end{bmatrix}
\end{aligned}$$

Hence, the projections $\mathcal{O}_i$ and $\mathcal{P}_i$ are given as

$$\begin{aligned}\mathcal{O}_i &= \left[\widehat{L}_2^0(:,1:im)\tfrac{(I-\Lambda^2)}{(1+\gamma)I-\Lambda^2} \quad \tfrac{1}{1+\gamma}\widehat{L}_2^0(:,im+1:i(m+l))\right] W_p,\\ \mathcal{P}_i &= \left[\widehat{L}_2^0(:,1:im)\tfrac{(I-\Lambda^2)}{I+\tilde{\gamma}\Lambda^2-\Lambda^2} \quad \widehat{L}_2^0(:,im+1:i(m+l))\right] W_p.\end{aligned}$$

Substituting $\tilde{\gamma} = \frac{\gamma}{1+\gamma}$ in the expression for $\mathcal{P}_i$, it is now easily seen that

$$\mathcal{P}_i = \left[\widehat{L}_2^0(:,1:im)\tfrac{(1+\gamma)(I-\Lambda^2)}{(1+\gamma)I-\Lambda^2} \quad \widehat{L}_2^0(:,im+1:i(m+l))\right] W_p = (1+\gamma)\mathcal{O}_i,$$

and $\mathcal{P}_i$ and $\mathcal{O}_i$ are equivalent up to a scalar multiplication which ends the proof. □

Note from Lemma 5.4 that if $\gamma \rightarrow \infty$, $\frac{\gamma}{1+\gamma} \rightarrow 1$. Hence, in order to be able to explore the range $\widetilde{\gamma} = \frac{\gamma}{1+\gamma} > 1$, it is recommended to replace the regularized N4SID with a weighted regularized N4SID. The weighting matrix $S = \begin{bmatrix}\Lambda & 0\\ 0 & 0\end{bmatrix}$ attributes the largest weights along principal directions corresponding to small principal angles, and therefore directly acts on those components of the projection which tend to suffer the most from ill-conditioning problems. The projection $\mathcal{O}_i$ is then calculated as follows:

$$\begin{aligned}(\widetilde{L}_1, \widetilde{L}_2) &= \arg\min_{L_1, L_2} \left(\left\| Y_f - \begin{bmatrix} L_1 & L_2 \end{bmatrix} \begin{bmatrix} U_f \\ W_p \end{bmatrix} \right\|_F^2 + \widetilde{\gamma} \left\| L_2 \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} W_p \right\|_F^2 \right),\\ \mathcal{O}_i &= \widetilde{L}_2 W_p.\end{aligned}$$

The performance of the weighted regularized N4SID algorithm, was evaluated along the lines of the Monte-Carlo analysis in Section 5.5. Results, following the same notations and definitions as in Section 5.5, are shown in Table 5.4. Note that for $\widetilde{\gamma} = \frac{1}{2}$, we obtain exactly the same results as for $\gamma = 1$ in Table 5.3, which is explained by Lemma 5.3. It is clear from Table 5.4 that the optimal $\widetilde{\gamma}$ satisfies $\widetilde{\gamma} < 1$. Hence, on this particular example it does not make any difference whether one uses the regularized N4SID as discussed in Section 5.4 or the weighted regularized N4SID. However, there is no reason to assume that this will be the case in all practical applications.

5.7 A real-life example

The orthogonal decomposition method and the weighted regularized N4SID were tested on measurements from a flexible robot arm available from the online database DaISy (Database for the Identification of Systems) [45]. The arm is installed on an electrical motor. The input, the measured reaction torque of the structure on the ground, is a band-limited signal which is displayed in Figure 5.13. The output is the acceleration of the flexible arm. The entire dataset consists of 1000 input/output measurements. The experimental transfer

$\tilde{\gamma}$	γ	Mean(d_2)	Std(d_2)	Mean(d_∞)	Std(d_∞)
0.5	1.00	0.0178	0.0119	0.0429	0.0263
0.6	1.50	0.0176	0.0117	0.0424	0.0260
0.7	2.33	0.0174	0.0116	0.0421	0.0260
0.8	4.00	0.0173	0.0115	0.0420	0.0260
0.9	9.00	0.0173	0.0115	0.0420	0.0261
1.0	$+\infty$	0.0173	0.0115	0.0421	0.0262
1.1	-	0.0173	0.0115	0.0423	0.0264
1.2	-	0.0174	0.0116	0.0426	0.0266
1.3	-	0.0175	0.0116	0.0428	0.0269
1.4	-	0.0176	0.0117	0.0431	0.0271
1.5	-	0.0177	0.0117	0.0434	0.0274
1.6	-	0.0178	0.0118	0.0437	0.0276
1.7	-	0.0179	0.0119	0.0441	0.0279
1.8	-	0.0180	0.0120	0.0441	0.0282
1.9	-	0.0182	0.0121	0.0448	0.0285
2.0	-	0.0183	0.0121	0.0451	0.0288
3.0	-	0.0198	0.0132	0.0490	0.0319
4.0	-	0.0215	0.0142	0.0532	0.0349
5.0	-	0.0232	0.0154	0.0575	0.0380

Table 5.4: Mean and standard deviation of the distances between the true and the estimated transfer functions over all 1000 datasets for the weighted regularized PI-N4SID with various regularization constants.

function is also displayed in Figure 5.13 and consists of two clear peaks in the frequency band which is excited by the input.

The first 500 points of the dataset were used to estimate a linear dynamical model using the orthogonal decomposition method and the weighted regularized N4SID. The remaining 500 points were divided in a validation set and a test set, each with 250 datapoints. The validation set was used to determine the order of the model [2], chosen equal to 4, and the regularization constant $\tilde{\gamma}$, chosen equal to 1.

The resulting models were validated on the testset and resulted in a relatively small mean squared error of 0.0257 for the orthogonal decomposition method and 0.0254 for the weighted regularized N4SID. As a comparison, for the unregularized N4SID, an unstable model was obtained. The true output on the test-set and the estimated output using the regularized weighted N4SID are displayed in Figure 5.7.

5.8 Conclusions

In this chapter, the problem of ill-conditioning in combined stochastic-deterministic subspace identification algorithms was discussed. It was seen that combined subspace identification algorithms can be ill-conditioned if the

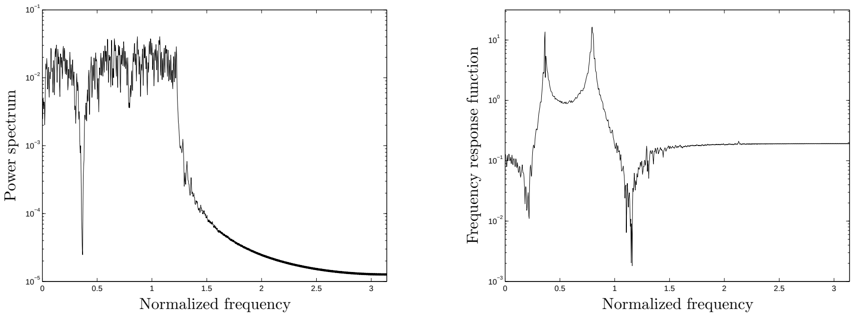


Figure 5.13: Input spectrum (left) and experimental frequency response function (right) from a flexible robot arm discussed in 5.7.

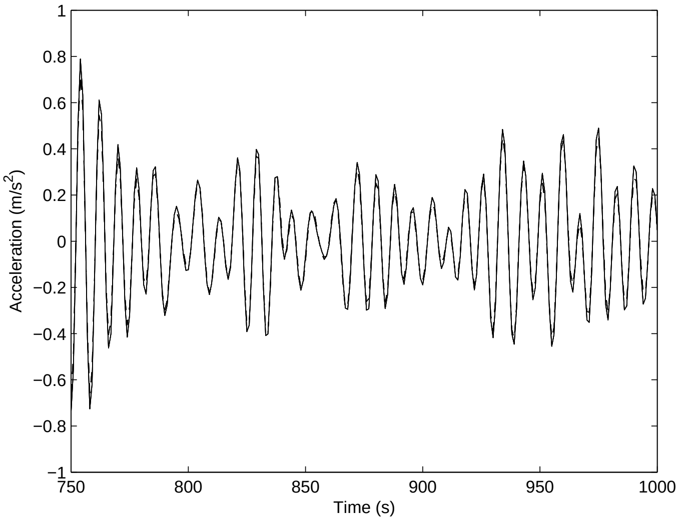


Figure 5.14: True (solid) and estimated (dashed) output on test-data from a flexible robot arm discussed in Section 5.7. The estimated output was generated using a weighted regularized N4SID model. The maximal error over the dataset is $0.07 m/s^2$. The mean squared error is $0.0254 (m/s^2)^2$.

smallest principal angle between U_f and W_p is close to zero, and/or stochastic resonances in the system are incorrectly attributed to the inputs. The orthogonal decomposition method, featuring an orthogonal projection in stead of an oblique projection, and a separation of the deterministic and the stochastic subsystem, was examined as a possible solution for the ill-conditioning problem. Thereafter, it was shown that by using weighted regularization along the principal directions of W_p and U_f corresponding to small principal angles in the N4SID algorithms, identification results could be obtained of the same, or better quality as those obtained with the orthogonal decomposition method.

Part II

Subspace identification for Hammerstein and Hammerstein-Wiener models

Chapter 6

Hammerstein, Wiener and Hammerstein-Wiener systems

In this chapter, Hammerstein, Wiener and Hammerstein-Wiener systems will be introduced. Existing identification algorithms for these classes of systems will be discussed with a special emphasis on the so-called overparameterization approach which will form the basis for a set of new identification algorithms presented in Chapters 7, 8 and 9.

6.1 The need for structured non-linear models

Throughout the last few decades, the field of linear system identification has been explored to the level that most linear identification problems can be solved efficiently with fairly standard and well known tools. However, with the advance in computer power, and considering that studied systems are often non-linear, the interest in non-linear system identification algorithms has steadily increased.

Driven by this demand and thanks to theoretical breakthroughs as in the area of splines [157], neural networks [18] and regularization networks [124], the field of non-linear modeling in general, and system identification in particular has steadily progressed over the last few years. Nevertheless, as the complexity of the identified models increases, the variance on the obtained parameters will increase as well (see e.g. [77, 100]). In extreme cases, this can lead to problems with the so-called 'curse of dimensionality', which is an inherent modeling problem closely associated with an explosion in the number of model parameters due to the presence of large input-dimensions and/or a lack of structure in the studied system and model. Hence, the interest in more structured model types, involving fewer free parameters, such as the bilinear model [105], the Hammerstein model, the Wiener model, and the Hammerstein-Wiener model [100, 160].

In part II of this thesis, a number of subspace identification algorithms will be introduced for the identification of Hammerstein and Hammerstein-Wiener systems using the theory of Least Squares Support Vector Machines (LS-SVMs) [135] and the basic ideas behind the overparameterization approach. After a discussion on the exact nature of Hammerstein and Hammerstein-Wiener systems in this chapter, LS-SVMs will be reviewed in Chapter 7 and shown to be suited for the identification of Hammerstein systems in the relatively simple ARX form. Armed with knowledge obtained from Chapter 7, the more complex N4SID subspace algorithm will be extended towards Hammerstein systems in Chapter 8. In Chapter 9 finally, a subspace identification algorithm will be proposed for the identification of Hammerstein-Wiener systems.

6.2 Hammerstein model identification

Hammerstein systems, in their most basic form, consist of a static memoryless non-linearity, followed by a linear dynamical system as shown in Figure 6.1. Due to their particularly simple structure, Hammerstein systems have been extensively studied in the context of system identification. Techniques for Hammerstein identification mainly distinguish themselves in the way the static non-linearity is represented and in the type of optimization problem that is finally obtained. In parametric approaches, the static non-linearity is expressed in terms of a finite number of parameters. Known approaches include the expansion of the non-linearity as a sum of (orthogonal or non-orthogonal) basis functions [104, 110, 114], the use of a finite number of cubic spline functions as presented in [47], piecewise linear functions [149] and neural networks [83]. Regardless of the parameterization scheme that is chosen, the final cost function will involve cross products between parameters describing the static non-linearity, and those describing the linear dynamical system. Employing a maximum likelihood criterion results in a so-called bi-convex optimization problem where global convergence is not guaranteed [131]. Hence, in order to find a good optimum for these techniques, a proper initialization is necessary [31].

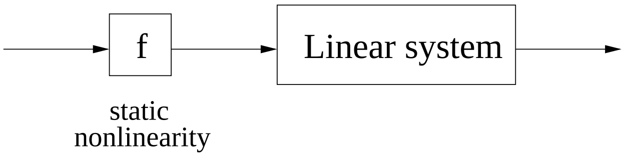


Figure 6.1: A Hammerstein system consists of a memoryless static non-linearity f followed by a linear dynamical system.

Although lots of techniques have been proposed to solve the bi-convex optimization problems typically encountered in Hammerstein system-identification, we will focus on three particular approaches that will turn out to be relevant in the remaining part of this thesis. These are the iterative approach, the stochastic approach, and the overparameterization approach. All three of them will briefly be described below.

6.2.1 Iterative approach

When in the bi-convex problem described earlier, either the parameters describing the linear system or the parameters describing the static non-linearity are kept constant, the remaining problem is solvable using known linear identification or non-linear regression algorithms. Based on this idea, one could start with an initial guess for the linear model and/or the static non-linearity, and estimate new linear models and static non-linearities in an iterative fashion. An example of an implementation of this idea is found in [110]. However, a drawback of such iterative techniques is that convergence is not guaranteed and that in some cases even a divergent behavior can be observed [134]. The iterative approach should therefore always be applied with caution, and at least a decent initial estimate is required. Such an initial estimate can conveniently be provided using the convex approaches that will be introduced in Subsections 6.2.2 and 6.2.3 below.

6.2.2 Stochastic approaches

It was shown in several publications that the typically bi-convex optimization problems at the core of Hammerstein identification algorithms can be replaced by convex optimization problems by assuming that certain statistical properties are satisfied by the input sequence u [13, 72]. A typical assumption thereby is whiteness of the inputs. This is easily understood by realizing that in this case $f(u)$ is also white, and the linear dynamical system can for instance be obtained from an output-only identification. Once the linear system is known, estimation of the static non-linearity is straightforward. More elaborate variations on this theme using white or multi-sine inputs and non-parametric estimates for the linear system are for instance found in [29–31]. Although these methods are in general easy to implement and exhibit a reasonable performance, an obvious drawback is the restrictions that are placed on the input. This limits their applicability to those practical cases where the user has full control over the experimental setup.

6.2.3 The overparameterization method

Another possibility to transform the bi-convex optimization problem into a convex one is by using a process known as overparameterization [12, 20]. In the latter, one replaces every crossproduct of unknowns by new independent parameters resulting in a convex but overparameterized optimization problem. In a second stage the obtained solution is projected onto the Hammerstein model class. Some examples of overparameterization approaches applied to the

Hammerstein identification problem are found in [12,104,104,110,114,156,159]. A classical problem with the overparameterization approach is the increased variance of the estimates due to the increased number of unknowns in the first stage. Nevertheless, due to their particularly attractive convexity property, and the fact that no restrictive assumptions are necessary on the inputs u we will mainly focus on overparameterization algorithms in this part of the thesis and demonstrate that the key ideas behind the overparameterization approach can conveniently be combined with the theory of LS-SVM regression to yield reliable ARX and subspace identification algorithms for Hammerstein systems.

Derivation of overparameterization

Mathematically, the idea of overparameterization can be summarized as writing the static non-linearity f as a linear combination of n_f general non-linear basis functions f_k, each with a certain weight c_k such that $f(u_t) = \sum_{k=1}^{n_f} c_k f_k(u_t)$. The functions f_1, f_2, and f_{n_f} are chosen beforehand. Assuming that the linear dynamical system is of the ARX form, the resulting Hammerstein model is given as follows:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j f(u_{t-j}) + e_t.$$

Substituting the expansion for f leads to:

$$\begin{aligned} y_t &= \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} \sum_{k=1}^{n_f} b_j c_k f_k(u_{t-j}) + e_t && (6.1) \\ &= \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} \sum_{k=1}^{n_f} \theta_{j,k} f_k(u_{t-j}) + e_t, && (6.2) \end{aligned}$$

which can be solved for $\theta_{j,k} = b_j c_k$, $j = 0, \ldots, m$, $k = 1, \ldots, n_f$ using a least squares algorithm. Denoting the estimates for $\theta_{j,k}$ by $\hat{\theta}_{j,k}$, estimates for the b_j and c_k are thereafter recovered from the SVD of:

$$\begin{bmatrix} \hat{\theta}_{0,1} & \hat{\theta}_{0,2} & \ldots & \hat{\theta}_{0,n_f} \\ \hat{\theta}_{1,1} & \hat{\theta}_{1,2} & \ldots & \hat{\theta}_{1,n_f} \\ \vdots & \vdots & & \vdots \\ \hat{\theta}_{m,1} & \hat{\theta}_{m,2} & \ldots & \hat{\theta}_{m,n_f} \end{bmatrix}. \tag{6.3}$$

Potential problems in overparameterization

Note that with $\mathcal{F}_j(\cdot) = \sum_{k=1}^{n_f} \theta_{j,k} f_k(\cdot)$, equation (6.2) is rewritten as:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} \mathcal{F}_j(u_{t-j}) + e_t.$$

Hence, another way to interpret the overparameterization method is as an estimation procedure of individual components $\mathcal{F}_j$ in a sum of non-linearities.

Estimating individual components in a sum of non-linearities is not without risks. Suppose for instance that $m = 1$, then it is easily seen that

$$\begin{aligned}\mathcal{F}_0(u_t) + \mathcal{F}_1(u_{t-1}) &= \mathcal{F}_0(u_t) + \delta + \mathcal{F}_1(u_{t-1}) - \delta \\ &= \mathcal{F}_0'(u_t) + \mathcal{F}_1'(u_{t-1})\end{aligned}$$

with δ an arbitrary constant and $\mathcal{F}_0'(u_t) = \mathcal{F}_0(u_t) + \delta$, $\mathcal{F}_1'(u_{t-1}) = \mathcal{F}_1(u_{t-1}) - \delta$. Similarly, note that for any set of variables $\epsilon_k, k = 1, \ldots, n_f$ with $\forall u \in \mathbb{R}, \sum_{k=1}^{n_f} \epsilon_k f_k(u) = \text{constant}$ and any set $\alpha_j, j = 0, \ldots, m$ such that $\sum_{j=0}^{m} \alpha_j = 0$, $\theta'_{j,k} = \theta_{j,k} + \alpha_j \epsilon_k$ is also a solution to (6.2) [77].

Hence, given a sequence of input/output measurements, all non-linearities estimated on these measurements will only be determined up to a set of constants. This problem is often overlooked in existing overparameterization techniques and may lead to conditioning problems and destroy the low-rank property of (6.3). In fact, many published overparameterization approaches applied to more complex Hammerstein systems lead to results which are far from optimal if no measures are taken to overcome this problem [62]. One possible solution is to use the estimates for $\theta_{j,k}, \ j = 0, \ldots, m, \ k = 1, \ldots, n_f$ to calculate:

$$A = \begin{bmatrix} \hat{\theta}_{0,1} & \hat{\theta}_{0,2} & \ldots & \hat{\theta}_{0,n_f} \\ \hat{\theta}_{1,1} & \hat{\theta}_{1,2} & \ldots & \hat{\theta}_{1,n_f} \\ \vdots & \vdots & & \vdots \\ \hat{\theta}_{m,1} & \hat{\theta}_{m,2} & \ldots & \hat{\theta}_{m,n_f} \end{bmatrix} \begin{bmatrix} f_1(u_1) & \ldots & f_1(u_N) \\ f_2(u_1) & \ldots & f_2(u_N) \\ \vdots & & \vdots \\ f_{n_f}(u_1) & \ldots & f_{n_f}(u_N) \end{bmatrix},$$

with $u_t, t = 1, \ldots, N$ the inputs of the system, subtract the mean of every row in A and take the SVD of the remaining matrix, from which estimates for the b_j can be extracted. Estimates for the c_k can then be found in a second round by solving (6.1). It is this approach that will be used when results from classical overparameterization approaches are discussed in the following chapters.

6.3 Wiener model identification

Wiener systems are very similar to Hammerstein systems. In their basic form, Wiener systems consist of a linear system followed by a static non-linearity g such as shown in Figure 6.2. If the linear system and the static non-linearity are invertible, it is easily seen that Wiener systems can be identified by using a Hammerstein identification algorithm with the role of the inputs- and the outputs reversed. In a more general setting, but still assuming invertibility of the output non-linearity, Wiener identification algorithms have been derived along the lines of their Hammerstein counterparts. Hence, as in the Hammerstein case, a distinction between convex and non-convex methods can be made, and again as in the Hammerstein case iterative approaches, stochastic approaches and overparameterization approaches are found as common solutions for the bi-convex optimization problem [87, 112]. If the output non-linearity

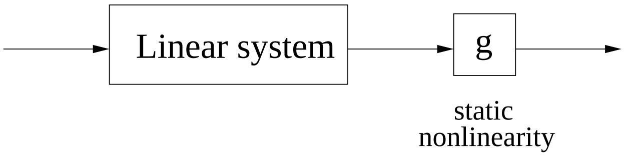


Figure 6.2: A Wiener system consists of a linear dynamical system followed by a memoryless static non-linearity g.

is not invertible, the main techniques used for Wiener model identification are stochastic in nature [17, 29, 31, 161]. In this thesis we will not further elaborate on Wiener model identification as most results which will be shown for Hammerstein models can conveniently be extended to Wiener models with invertible g. We will however treat the Hammerstein-Wiener case which is from a research perspective a much more challenging problem.

6.4 Hammerstein-Wiener model identification

Hammerstein-Wiener models are obtained by placing a Hammerstein system and a Wiener system in cascade, such as shown in Figure 6.3. In contrast to the literature on Hammerstein and Wiener systems, the available literature on the identification of Hammerstein-Wiener systems is rather sparse and related algorithms can not so easily be classified as their Hammerstein- and Wiener counterparts. In [12], a scheme for the identification of SISO Hammerstein-Wiener systems is developed based on the idea of overparameterization. However, in this scheme a very specific model structure is assumed, limiting its practical applicability. Based on [12], a more general so-called blind approach for the identification of SISO systems was proposed in [14]. An identification method for Hammerstein-Wiener MIMO systems was proposed in [29, 30] but imposes strict restrictions on the inputs and is iterative in nature. Other contributions such as [48, 166] are limited to SISO systems and/or iterative in nature.

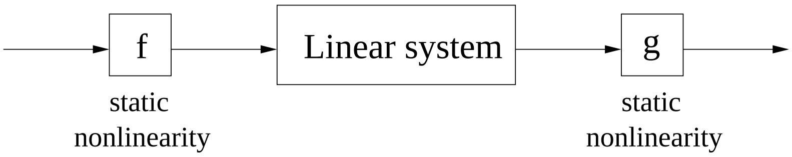


Figure 6.3: A Hammerstein-Wiener system is obtained by putting a Hammerstein- and a Wiener-system in cascade.

Hence, in general one can state that to date, no reliable MIMO identification algorithm is present which is non-iterative in nature and does not rely on restrictive assumptions on the inputs. An attempt at such an algorithm, using a combination of kernel canonical correlation analysis and the subspace intersection algorithm, will be presented in Chapter 9.

Chapter 7

Hammerstein ARX identification

7.1 Introduction

In this chapter, we explore the use of Least Squares Support Vector Machines (LS-SVMs) for Hammerstein ARX model identification. It will be shown that the linear model parameters and the static non-linearity can be obtained by solving a set of linear equations with size in the order of the number of observations. Given the convexity and the large number of parameters involved, the method may be regarded as an overparameterization approach. However, due to the presence of a regularization framework [127, 135, 150], the variance of the obtained estimates is significantly lower than in classical overparameterization approaches discussed in Subsection 6.2.3. Due to this decrease in variance, systems with several inputs and outputs can be estimated conveniently with the presented technique. Another advantage of the proposed derivation is the fact that additional 'centering'-constraints and parametric components of the linear dynamical system can naturally be included in the LS-SVM framework, due to the fact that it is closely related to optimization theory.

Furthermore, in contrast to classical parametric approaches, no specific model structure is imposed on the non-linearity other than a certain shape (e.g. a degree of smoothness). Hence, the presented technique combines a nonparametric approach with parametric assumptions on the dynamical system and on the noise model. The technique distinguishes itself from existing nonparametric approaches [70, 72, 73, 78, 91, 114, 156] in the flexibility to incorporate prior knowledge on the shape of the non-linearity by plug-in of an appropriate kernel (e.g. linear, polynomial, RBF, spline). Moreover, no restrictive assumptions on the inputs (as e.g. whiteness) need to be made.

The outline of this chapter is as follows: In Section 7.2, some basic aspects of LS-SVMs applied to static function estimation are reviewed. In Sections 7.3

and 7.4 a method for the identification of non-linear SISO Hammerstein systems is proposed. In Section 7.5 the method is extended to MIMO Hammerstein systems. In Section 7.6 a comparison is made with existing overparameterization algorithms, and in Section 7.7 the method proposed in this chapter is tested and compared to those existing methods on a number of SISO and MIMO examples.

7.2 Least Squares Support Vector Machines for function approximation

In this section, we review some elements of Least Squares Support Vector Machines for static function approximation. The theory introduced here will be extended to the estimation of Hammerstein systems in Section 7.3.

Let $\{(x_t, y_t)\}_{t=1}^{N} \subset \mathbb{R}^d \times \mathbb{R}$ be a set of input/output training data with input x_t and output y_t. Consider the regression model $y_t = f(x_t) + e_t$ where $x_1, \ldots, x_N$ are deterministic points, $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is an unknown real-valued smooth (i.e. Lipschitz continuous) function and $e_1, \ldots, e_N$ are uncorrelated random errors with $E\left[e_t\right] = 0$, $E\left[e_t^2\right] = \sigma_e^2 < \infty$. In recent years, Support Vector Machines (SVMs) [150] have been used for the purpose of estimating the non-linear f. The following model is assumed:

$$f(x) = w^T \varphi(x) + b,$$

where $\varphi(x) : \mathbb{R}^d \rightarrow \mathbb{R}^{n_H}$ denotes a potentially infinite ($n_H = \infty$) dimensional feature map, $w \in \mathbb{R}^{n_H}$, $b \in \mathbb{R}$. The regularized cost function of the Least Squares SVM (LS-SVM) [135] is given as

$$\begin{aligned} \min_{w,b,e} \mathcal{J}(w,e) &= \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{t=1}^{n} e_t^2, \\ \text{subject to} : y_t &= w^T \varphi(x_t) + b + e_t, \quad t = 1, \ldots, N. \end{aligned}$$

The relative importance between the smoothness of the solution and the data fitting is governed by the scalar $\gamma \in \mathbb{R}_0^+$ referred to as the regularization constant. The optimization performed corresponds to ridge regression [68] in feature space. In order to solve the constrained optimization problem, a Lagrangian is constructed:

$$\mathcal{L}(w, b, e; \alpha) = \mathcal{J}(w, e) - \sum_{t=1}^{N} \alpha_t \{ w^T \varphi(x_t) + b + e_t - y_t \},$$

with α_t the Lagrange multipliers. The conditions for optimality are given by:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{t=1}^{N} \alpha_t \varphi(x_t), \tag{7.1}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \quad \rightarrow \quad \sum_{t=1}^{N} \alpha_t = 0, \tag{7.2}$$

$$\frac{\partial \mathcal{L}}{\partial e_t} = 0 \quad \rightarrow \quad \alpha_t = \gamma e_t, \;\; t = 1, \ldots, N, \tag{7.3}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_t} = 0 \quad \rightarrow \quad y_t = w^T \varphi(x_t) + b + e_t, \;\; t = 1, \ldots, N. \tag{7.4}$$

Substituting (7.1)-(7.3) into (7.4) yields the following dual problem (i.e. the problem in the Lagrange multipliers):

$$\left[\begin{array}{c|c} 0 & 1_N{}^T \\ \hline 1_N & \Omega + \gamma^{-1} I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right], \tag{7.5}$$

where $y = \begin{bmatrix} y_1 & \ldots & y_N \end{bmatrix}^T$, $1_N = \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix}^T$, $\alpha = \begin{bmatrix} \alpha_1 & \ldots & \alpha_N \end{bmatrix}^T$, $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$, $\forall i, j = 1, \ldots, N$, with K the positive definite kernel function. Note that in order to solve the set of equations (7.5), the feature map φ does never have to be defined explicitly. Only its inner product, a positive definite Mercer kernel, is needed. This is called the kernel trick [127, 150]. For the choice of the kernel $K(\cdot, \cdot)$, see e.g. [127]. Typical examples are the use of a linear kernel $K(x_i, x_j) = x_i^T x_j$, a polynomial kernel $K(x_i, x_j) = (\tau + x_i^T x_j)^d$ of degree d or an RBF kernel $K(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$ where σ denotes the bandwidth of the kernel. The resulting LS-SVM model for function estimation can be evaluated at a new point x_* as

$$\hat{f}(x_*) = \sum_{t=1}^{N} \alpha_t K(x_*, x_t) + b,$$

where (b, α) is the solution to (7.5). Note that in the above, no indication is given as to how to choose free parameters such as the regularization constant γ and the bandwidth σ in an RBF kernel. These parameters, which are generally referred to as *hyper-parameters* will have to be obtained from data, e.g. by tuning on an independent validation dataset, or by using cross-validation [77].

Besides the function estimation case, the class of LS-SVMs also includes classification, kernel PCA (principal component analysis), kernel CCA, kernel PLS (partial least squares), recurrent networks and solutions to non-linear optimal control problems. For an overview on applications of the LS-SVM framework, the reader is referred to [80, 135–137].

7.3 Identification of ARX Hammerstein models

In the following derivation, we will restrict ourselves to SISO systems, but as will be shown in Section 7.5, the presented method is applicable to the MIMO case as well. For the linear dynamical part, we will assume a model structure of the ARX form [100]:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j u_{t-j} + e_t, \tag{7.6}$$

with $u_t, y_t \in \mathbb{R}, t \in \mathbb{Z}$ and $\{u_t, y_t\}$ a set of input and output measurements. The so-called equation error e_t is assumed to be white with finite second order moments, and m and n denote the order of the numerator and denominator in the transfer function of the linear model. The model structure (7.6) is generally known as the "AutoRegressive model with eXogenous inputs" (ARX) and is one of the best known model structures in linear identification. Adding a static non-linearity $f : \mathbb{R} \to \mathbb{R} : u \to f(u)$ to (7.6) leads to:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j f\left(u_{t-j}\right) + e_t, \tag{7.7}$$

which is the general model structure that is assumed in this chapter (see also Figure 6.1).

Applying LS-SVM function estimation outlined in the previous section, we assume the following structure for the static non-linearity f:

$$f(u) = w^T \varphi(u) + d_0.$$

with $\Omega_{ij} = K(u_i, u_j) = \varphi(u_i)^T \varphi(u_j)$ a kernel of choice. Hence, equation (7.7) can be rewritten as follows:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j \left(w^T \varphi\left(u_{t-j}\right) + d_0\right) + e_t. \tag{7.8}$$

We focus on finding estimates for the linear parameters $a_i, i = 1, \ldots, n$ and $b_j, j = 0, \ldots, m$ and the static non-linearity f, parameterized by w and d_0, from a finite set of measurements $\{u_t, y_t\}$, $t = 1, \ldots, N$. With $r = \max(m, n) + 1$, the resulting optimization problem is:

$$\min_{w,a,b,d_0,e} \mathcal{J}(w, e) = \min_{w,a,b,d_0,e} \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{t=r}^{T} e_t^2,$$

subject to (7.8). The Lagrangian of the resulting estimation problem is given by

$$\begin{aligned} \mathcal{L}(w, d_0, b, e, a; \alpha) = \mathcal{J}(w, e) - \sum_{t=r}^{N} \alpha_t \{ \sum_{i=1}^{n} a_i y_{t-i} & \\ + \sum_{j=0}^{m} b_j \left(w^T \varphi\left(u_{t-j}\right) + d_0\right)\right) + e_t - y_t \}. \end{aligned} \tag{7.9}$$

The conditions for optimality are given by:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \quad \to \quad w = \sum_{t=r}^{N} \sum_{j=0}^{m} \alpha_t b_j \varphi(u_{t-j}), \tag{7.10}$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial d_0} = 0 \quad &\rightarrow \quad \sum_{t=r}^{N} \sum_{j=0}^{m} \alpha_t b_j = 0, \\
\frac{\partial \mathcal{L}}{\partial a_i} = 0 \quad &\rightarrow \quad \sum_{t=r}^{N} \alpha_t y_{t-i} = 0, \;\; i = 1, \ldots, n, \\
\frac{\partial \mathcal{L}}{\partial b_j} = 0 \quad &\rightarrow \quad \sum_{t=r}^{N} \alpha_t \left(w^T \varphi(u_{t-j}) + d_0 \right) = 0, \\
\frac{\partial \mathcal{L}}{\partial e_t} = 0 \quad &\rightarrow \quad \alpha_t = \gamma e_t, \;\; t = r, \ldots, N, \qquad (7.11) \\
\frac{\partial \mathcal{L}}{\partial \alpha_t} = 0 \quad &\rightarrow \quad (7.8), \;\; t = r, \ldots, N. \qquad (7.12)
\end{aligned}
$$

Substituting (7.10) and (7.11) in (7.12) leads to:

$$
\sum_{j=0}^{m} \sum_{q=r}^{N} \sum_{p=0}^{m} b_j \left(b_p \alpha_q \varphi\left(u_{q-p}\right)^T \varphi\left(u_{t-j}\right) + d_0 \right) + \sum_{i=1}^{n} a_i y_{t-i} + e_t - y_t = 0, t = r, \ldots, N. \qquad (7.13)
$$

If the b_j values were known, the resulting problem would be linear in the unknowns and easy to solve as:

$$
\left[\begin{array}{c|c|c}
0 & 0 & \tilde{b} \cdot 1_{N-r+1}^T \\ \hline
0 & 0 & \mathcal{Y}_p \\ \hline
\tilde{b} \cdot 1_{N-r+1} & \mathcal{Y}_p^T & \mathcal{K} + \gamma^{-1} I
\end{array}\right]
\left[\begin{array}{c}
d_0 \\ \hline a \\ \hline \alpha
\end{array}\right]
=
\left[\begin{array}{c}
0 \\ \hline 0 \\ \hline \mathcal{Y}_f
\end{array}\right], \qquad (7.14)
$$

with

$$
\begin{aligned}
\alpha &= \begin{bmatrix} \alpha_r & \ldots & \alpha_N \end{bmatrix}^T, \;\; \tilde{b} = \sum_{j=0}^{m} b_j, \\
a &= \begin{bmatrix} a_1 & \ldots a_n \end{bmatrix}^T, \;\; \mathcal{Y}_f = \begin{bmatrix} y_{r+1} & \ldots & y_N \end{bmatrix}^T, \\
\mathcal{Y}_p &= \begin{bmatrix} y_{r-1} & y_r & \ldots & y_{N-1} \\ y_{r-2} & y_{r-1} & \ldots & y_{N-2} \\ \vdots & \vdots & & \vdots \\ y_{r-n} & y_{r-n+1} & \ldots & y_{N-n} \end{bmatrix}, \\
\mathcal{K}(p,q) &= \sum_{j=0}^{m} \sum_{l=0}^{m} b_j b_l \Omega_{p+r-j-1, q+r-l-1}, \\
\Omega_{k,l} &= \varphi(u_k)^T \varphi(u_l), \;\; k, l = 1, \ldots, N.
\end{aligned}
$$

Since the b_j values are in general not known and the solution to the resulting third order estimation problem (7.13) is by no means trivial, we will use an approximative method to obtain models of the form (7.7).

7.4 An approximative method

7.4.1 Optimization using collinearity constraints

In order to avoid solving the problem (7.13), we rewrite (7.8) as follows:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} w_j^T \varphi(u_{t-j}) + d + e_t, \tag{7.15}$$

which can conveniently be solved using LS-SVM's. Note, however, that the resulting model class is wider than (7.8) due to the replacement of one single w by several vectors $w_j, j = 0, \ldots, m$. The model class (7.15) is therefore not necessarily limited to the description of Hammerstein systems. A sufficient condition for the estimated model to belong to this class of systems is that the obtained w_j must be collinear in which case w_j is seen as a replacement for $b_j w$. Taking this into account during the estimation leads to extra constraints requiring the angles between any pair $\{w_j, w_k\}, j, k = 0, \ldots, m$ to be zero, or $\left(w_j^T w_k\right)^2 = \sqrt{w_j^T w_j}\sqrt{w_k^T w_k}$. Alternatively, the collinearity constraint can be written as: rank $\begin{bmatrix} w_0 & \ldots & w_m \end{bmatrix} = 1$, which is equivalent to ensuring that a set of $\frac{m(m+1)n_H(n_H-1)}{4}$ 2×2 determinants are zero. As n_H (the dimension of w) is unknown and possibly very high, it is obvious that including such constraints in the Lagrangian would again lead to a non-convex optimization problem.

Considering that ARX Hammerstein models are contained in the set of models of the form (7.15), we therefore propose to remove the collinearity constraints from the Lagrangian altogether, solve the more general problem (7.15), and project the obtained model onto the model-set (7.8) later. Hereby, we assume that even though collinearity was not explicitly imposed, it will automatically be nearly satisfied in the estimated model of the form (7.15). Although this approach may seem ad-hoc at first, it is essentially an application of Bai's overparameterization approach [12] to LS-SVMs. As was seen in Subsection 6.2.3, in essence, in overparameterization approaches the static non-linearity is written as a linear combination of general non-linear basis-functions f_i, each with a certain weight c_i, e.g.

$$f(u_t) = \begin{bmatrix} c_1 & \ldots & c_{n_f} \end{bmatrix} \begin{bmatrix} f_1(u_t) & \ldots & f_{n_f}(u_t) \end{bmatrix}^T,$$

where f_1, f_2, and f_{n_f} are chosen beforehand. This substitution was seen to lead to a classical linear identification algorithm where linear model parameters $p_1, p_2, \ldots$ are replaced by vectors $p_1 \begin{bmatrix} c_1 & \ldots & c_{n_f} \end{bmatrix}, p_2 \begin{bmatrix} c_1 & \ldots & c_{n_f} \end{bmatrix}, \ldots$. Afterwards, collinearity of these vectors is imposed, e.g. by applying an SVD and taking a rank one approximation, and the original model parameters $p_1, p_2, \ldots$ are recovered.

7.4.2 Optimization without collinearity constraints

Disregarding the collinearity constraints, the optimization problem that is ultimately solved is the following:

$$\min_{w_j,a,d,e} \mathcal{J}(w_j,e) = \min_{w_j,a,d,e} \frac{1}{2}\sum_{j=0}^{m} w_j^T w_j + \gamma\frac{1}{2}\sum_{t=r}^{N} e_t^2, \tag{7.16}$$

subject to

$$\begin{aligned} \sum_{j=0}^{m} w_j^T \varphi(u_{t-j}) + \sum_{i=1}^{n} a_i y_{t-i} + d + e_t - y_t &= 0, \quad t = r,\ldots,N, & (7.17)\\ \sum_{t=1}^{N} w_j^T \varphi(u_t) &= 0, \quad j = 0,\ldots,m. & (7.18) \end{aligned}$$

The problem (7.16)-(7.18), is known as a component-wise LS-SVM regression problem and was first introduced in [118]. The term component-wise refers to the fact that the output is ultimately written as the sum of a set of linear and non-linear components. As will be seen shortly, the derivation of a solution to a component-wise LS-SVM problem follows the same kind of reasoning as that of an ordinary LS-SVM regression problems. Also note the additional constraints (7.18) to center the non-linear functions $w_j^T\varphi(\cdot), j = 0,\ldots,m$ around their average over the training set. This removes the uncertainty resulting from the fact that any set of constants can be added to the terms of the additive non-linear function (7.15), as long as the sum of the constants is zero (see 6.2.3 for the equivalent in classical overparameterization approaches). Removing this uncertainty will facilitate the extraction of the parameters b_j in (7.7) later. Furthermore, this constraint enables us to give a clear meaning to the bias parameter d, namely $d = \sum_{j=0}^{m} b_j \left(\frac{1}{N}\sum_{k=1}^{N} f(u_k)\right)$. An extra advantage of the LS-SVM approach is that constraints of the form (7.18) can naturally be included in the Lagrangian.

Lemma 7.1. *Given the system (7.15), the LS-SVM estimates for the non-linear functions $w_j^T\varphi : \mathbb{R} \to \mathbb{R}$, $j = 0,\ldots,m$, are given as:*

$$w_j^T\varphi(u_*) = \sum_{t=r}^{N} \alpha_t K(u_{t-j}, u_*) + \beta_j \sum_{t=1}^{N} K(u_t, u_*) \tag{7.19}$$

where the parameters $\alpha_t, t = r,\ldots,N$, $\beta_j, j = 0,\ldots,m$, as well as the linear model parameters $a_i, i = 1,\ldots,n$ and d are obtained from the following set of linear equations:

$$\left[\begin{array}{c|c|c|c} 0 & 0 & 1^T & 0 \\ \hline 0 & 0 & \mathcal{Y}_p & 0 \\ \hline 1 & \mathcal{Y}_p^T & \mathcal{K} + \gamma^{-1} I & K^0 \\ \hline 0 & 0 & {K^0}^T & 1_N^T \Omega 1_N \cdot I_{m+1} \end{array}\right] \left[\begin{array}{c} d \\ \hline a \\ \hline \alpha \\ \hline \beta \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline \mathcal{Y}_f \\ \hline 0 \end{array}\right], \tag{7.20}$$

with

$$\beta = \begin{bmatrix}\beta_0 & \dots & \beta_m\end{bmatrix}^T, K^0(p,q) = \textstyle\sum_{t=1}^{N} \Omega_{t,r+p-q},$$
$$\mathcal{K}(p,q) = \textstyle\sum_{j=0}^{m} \Omega_{p+r-j-1,q+r-j-1},$$

and 1_N *is a column vector of length* N *with elements* 1.

Proof. This follows directly from the Lagrangian:

$$\mathcal{L}(w_j, d, a, e; \alpha, \beta) = \mathcal{J}(w_j, e) - \sum_{j=0}^{m} \beta_j \{\sum_{t=1}^{N} w_j^T \varphi(u_t)\} - \sum_{t=r}^{N} \alpha_t \{\sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} w_j^T \varphi(u_{t-j}) + d + e_t - y_t\}, \quad (7.21)$$

by taking the conditions for optimality: $\frac{\partial \mathcal{L}}{\partial w_j} = 0$, $\frac{\partial \mathcal{L}}{\partial a_i} = 0$, $\frac{\partial \mathcal{L}}{\partial d} = 0$, $\frac{\partial \mathcal{L}}{\partial e_t} = 0$, $\frac{\partial \mathcal{L}}{\partial \alpha_t} = 0$, $\frac{\partial \mathcal{L}}{\partial \beta_j} = 0$. □

Note that the martix $\mathcal{K}$, which figures at the left hand side of (7.20) and plays a similar role as the kernel matrix Ω in (7.5), actually represents a sum of kernels in (7.20). This follows as a typical property of the solution of component-wise LS-SVM problems [118].

7.4.3 Projecting the unconstrained solution onto the class of ARX Hammerstein models

The projection of the obtained model onto (7.7) goes as follows. Estimates for the autoregressive parameters $a_i, i = 1, \dots, n$ are directly obtained from (7.20). Furthermore, for the training input sequence $\begin{bmatrix}u_1 & \dots & u_N\end{bmatrix}$, we have:

$$\begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix} \begin{bmatrix} \underline{\hat{f}}(u_1) \\ \vdots \\ \underline{\hat{f}}(u_N) \end{bmatrix}^T = \begin{bmatrix} \alpha_N & \dots & \alpha_r & & & 0 \\ & \alpha_N & \dots & \alpha_r & & \\ & & \ddots & & \ddots & \\ 0 & & & \alpha_N & \dots & \alpha_r \end{bmatrix} \times \begin{bmatrix} \Omega_{N,1} & \Omega_{N,2} & \dots & \Omega_{N,N} \\ \Omega_{N-1,1} & \Omega_{N-1,2} & \dots & \Omega_{N-1,N} \\ \vdots & \vdots & & \vdots \\ \Omega_{r-m,1} & \Omega_{r-m,2} & \dots & \Omega_{r-m,N} \end{bmatrix} + \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_m \end{bmatrix} \sum_{t=1}^{N} \begin{bmatrix} \Omega_{t,1} \\ \vdots \\ \Omega_{t,N} \end{bmatrix}^T, \quad (7.22)$$

with $\underline{\hat{f}}(u)$ an estimate for

$$\underline{f}(u) = f(u) - \frac{1}{N} \sum_{t=1}^{N} f(u_t).$$

Hence, estimates for b_j and the static non-linearity f can be obtained from a rank 1 approximation of the right hand side of (7.22), for instance using a singular value decomposition. Again, this is the equivalent of the SVD-step that is generally encountered in overparameterization methods [12, 20]. Once all the elements b_j are known, $\sum_{t=1}^{N} f(u_k)$ can be obtained as $\sum_{t=1}^{N} f(u_t) = \frac{Nd}{\sum_{j=0}^{m} b_j}$.

7.4.4 Further comments

Persistency of excitation

Given the fact that regularization is inherently present in the proposed identification technique, lack of persistency of excitation will not lead to any numerical problems. However, to ensure that all aspects of the linear system are properly identified, persistency of excitation of $f(u)$ of at least order $n+m+1$ is desired [100]. For some non-linear functions f, persistency of excitation of $f(u)$ can be guaranteed if u is persistently exciting (see [156] for a discussion on this issue).

Iterative identification

Though outside the scope of the present chapter, one possible extension to the algorithm presented in this section would be to use the algorithm described in the former subsection as an initialization to the biconvex problem encountered in Section 7.3. Given a good initialization, the latter could be solved using iterative techniques such as presented in Subsection 6.2.1, where the linear and non-linear parameters are alternatively kept constant, while an optimization is performed over the remaining parameters, using for instance the linear set of equations (7.14).

7.5 Extension to the MIMO case

Technically, an extension of the algorithms presented in the former section to the MIMO case is straightforward, but the calculations involved are quite extensive. Assuming a MIMO Hammerstein system of the form:

$$y_t = \sum_{i=1}^{n} A_i y_{t-i} + \sum_{j=0}^{m} B_j f\left(u_{t-j}\right) + e_t, \tag{7.23}$$

with $y_t, e_t \in \mathbb{R}^{n_y}$, $u_t \in \mathbb{R}^{n_u}$, $A_i \in \mathbb{R}^{n_y \times n_y}$, $B_j \in \mathbb{R}^{n_y \times n_u}$, $t = 1, \ldots, N$, $i = 1, \ldots, n$, $j = 0, \ldots, m$, and $f : \mathbb{R}^{n_u} \to \mathbb{R}^{n_u} : u \to f(u) = \begin{bmatrix} f_1(u) & \ldots & f_{n_u}(u) \end{bmatrix}^T$, we have for every row s in (7.23), that

$$y_t(s) = \sum_{i=1}^{n} A_i(s,:) y_{t-i} + \sum_{j=0}^{m} B_j(s,:) f\left(u_{t-j}\right) + e_t(s). \tag{7.24}$$

Note that for every non-singular matrix $V \in \mathbb{R}^{n_u \times n_u}$, and for any $j = 0, \ldots, m$:

$$B_j(s,:) f\left(u_{t-j}\right) = B_j(s,:) V V^{-1} f\left(u_{t-j}\right). \tag{7.25}$$

Hence, any model of the form (7.23) can be replaced with an equivalent model by applying a linear transformation on the components of f and the columns of B_j. This will have to be taken into account when identifying models of the form (7.23) without any prior knowledge of the non-linearity involved.
Substituting $f(u) = \begin{bmatrix} f_1(u) & \dots & f_{n_u}(u) \end{bmatrix}^T$ in (7.24) leads to:

$$y_t(s) = \sum_{i=1}^{n} A_i(s,:)y_{t-i} + \sum_{j=0}^{m}\sum_{k=1}^{n_u} B_j(s,k) f_k\left(u_{t-j}\right) + e_t(s). \tag{7.26}$$

By replacing $\sum_{k=1}^{n_u} B_j(s,k)f_k(u_{t-j})$ by $w_{j,s}^T\varphi(u_{t-j}) + d_{s,j}$ this reduces to

$$y_t(s) = \sum_{i=1}^{n} A_i(s,:)y_{t-i} + \sum_{j=0}^{m} \omega_{j,s}^T \varphi\left(u_{t-j}\right) + d_s + e_t(s). \tag{7.27}$$

where

$$d_s = \sum_{j=0}^{m} d_{s,j}. \tag{7.28}$$

The primal problem that is subsequently obtained is the following:

$$\min_{\omega_{j,s},e} \mathcal{J}(\omega_{j,s}, e) = \sum_{j=0}^{m}\sum_{s=1}^{n_y} \frac{1}{2}\omega_{j,s}^T\omega_{j,s} + \sum_{s=1}^{n_y}\sum_{t=r}^{N} \frac{\gamma_s}{2} e_t(s)^2. \tag{7.29}$$

subject to (7.27) and $\sum_{t=1}^{N} w_{j,s}^T\varphi(u_t) = 0,\ \ j = 0,\dots,m,\ \ s = 1,\dots,n_y.$

Lemma 7.2. *Given the primal problem (7.29), the LS-SVM estimates for the non-linear functions* $w_{j,s}^T\varphi : \mathbb{R} \to \mathbb{R}$, $j = 0,\dots,m$, $s = 1,\dots,n_y$, *are given as:*

$$w_{j,s}^T\varphi(u_*) = \sum_{t=r}^{N} \alpha_{t,s} K(u_{t-j}, u_*) + \beta_{j,s} \sum_{t=1}^{N} K(u_t, u_*) \tag{7.30}$$

where the parameters $\alpha_{t,s}, t = r,\dots,N$, $s = 1,\dots,n_y$, $\beta_{j,s}, j = 0,\dots,m$, $s = 1,\dots,n_y$ *as well as the linear model parameters* $A_i, i = 1,\dots,n$ *and* d_s, $s = 1,\dots,n_y$ *are obtained from the following set of linear equations:*

$$\begin{bmatrix} L_1 & & \\ & \ddots & \\ & & L_{n_y} \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_{n_y} \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ R_{n_y} \end{bmatrix}, \tag{7.31}$$

where

$$L_s = \left[\begin{array}{c|c|c|c} 0 & 0 & 1^T & 0 \\ \hline 0 & 0 & \mathcal{Y}_p & 0 \\ \hline 1 & \mathcal{Y}_p^T & \mathcal{K} + \gamma_s^{-1} I & \mathcal{S} \\ \hline 0 & 0 & \mathcal{S}^T & \mathcal{T} \end{array}\right], \quad X_s = \left[\begin{array}{c} d_s \\ \hline \mathcal{A}_s \\ \hline \alpha_s \\ \hline \beta_s \end{array}\right],$$

$$
\begin{aligned}
R_s &= \left[\ 0 \mid 0 \mid \mathcal{Y}_{f,s}^T \mid 0\ \right]^T, \ \mathcal{Y}_{f,s} = \left[y_r(s)^T \quad \ldots \quad y_N(s)^T\right]^T, \\
\mathcal{A}_s &= \begin{bmatrix} A_1(s,:)^T \\ \vdots \\ A_m(s,:)^T \end{bmatrix}, \ \overline{\alpha}_s = \begin{bmatrix} \alpha_{r,s} \\ \vdots \\ \alpha_{N,s} \end{bmatrix}, \ \mathcal{S}(p,q) = \sum_{t=1}^{N} \Omega_{t,r+p-q}, \\
\overline{\beta}_s &= \left[\beta_{0,s} \quad \ldots \quad \beta_{m,s}\right]^T, \ \Omega_{p,q} = \varphi(u_p)^T \varphi(u_q), \\
\mathcal{K}(p,q) &= \sum_{j=0}^{m} \Omega_{p+r-j-1,q+r-j-1}, \ \mathcal{T} = 1_N^T \Omega 1_N \cdot I_{m+1}.
\end{aligned}
$$

Proof. This directly follows from the Lagrangian:

$$
\mathcal{L}(\omega_{j,s}, d_s, A, e; \alpha, \beta) = \mathcal{J}(\omega_{j,s}, e) - \sum_{t=r}^{N} \sum_{s=1}^{n_y} \alpha_{t,s} \left\{ \sum_{i=1}^{n} A_i(s,:) y_{t-i} + \right.
$$

$$
\left. \sum_{j=0}^{m} \omega_{j,s}^T \varphi\left(u_{t-j}\right) + d_s + e_t(s) - y_t(s) \right\} - \sum_{j=0}^{m} \sum_{s=1}^{n_y} \beta_{j,s} \left\{ \sum_{t=1}^{N} \omega_{j,s}^T \varphi(u_t) \right\}, \quad (7.32)
$$

by taking the conditions for optimality: $\frac{\partial \mathcal{L}}{\partial \omega_{j,s}} = 0$, $\frac{\partial \mathcal{L}}{\partial A_i(s,:)} = 0$, $\frac{\partial \mathcal{L}}{\partial d_s} = 0$, $\frac{\partial \mathcal{L}}{\partial e_t(s)} = 0$, $\frac{\partial \mathcal{L}}{\partial \alpha_{t,s}} = 0$, $\frac{\partial \mathcal{L}}{\partial \beta_{j,s}} = 0$. □

Note that the matrices $L_s, s = 1, \ldots, n_y$ in (7.31) are almost identical, except for the different regularization constants γ_s. In many practical cases, however, and if there is no reason to assume that a certain output is more important than another, it is recommended to set $\gamma_1 = \gamma_2 = \ldots = \gamma_{n_y}$. This will speed up the estimation algorithm since $L_1 = L_2 = \ldots = L_{n_y}$ needs to be calculated only once, but most importantly, it will reduce the number of hyper-parameters to be tuned.

The projection of the obtained model onto (7.26) is similar as in the SISO case. Estimates for the autoregressive matrices $A_i, i = 1, \ldots, n$ are directly obtained from (7.31). For the training input sequence $[\ u_1 \quad \ldots \quad u_N\ ]$ and every $k = 1, \ldots, n_u$, we have:

$$
\left[\begin{array}{c} B_0(1,:) \\ \vdots \\ B_m(1,:) \\ \hline \vdots \\ \hline B_0(n_y,:) \\ \vdots \\ B_m(n_y,:) \end{array}\right]
\begin{bmatrix} \underline{\hat{f}}^T(u_1) \\ \vdots \\ \underline{\hat{f}}^T(u_N) \end{bmatrix}^T
=
\left[\begin{array}{c} \beta_{0,1} \\ \vdots \\ \beta_{m,1} \\ \hline \vdots \\ \hline \beta_{0,n_y} \\ \vdots \\ \beta_{m,n_y} \end{array}\right]
\sum_{t=1}^{N} \begin{bmatrix} \Omega_{t,1} \\ \vdots \\ \Omega_{t,N} \end{bmatrix}^T
$$

$$
+ \left[\begin{array}{c} \mathcal{A}_1 \\ \hline \vdots \\ \hline \mathcal{A}_{n_y} \end{array}\right] \times \begin{bmatrix} \Omega_{N,1} & \Omega_{N,2} & \ldots & \Omega_{N,N} \\ \Omega_{N-1,1} & \Omega_{N-1,2} & \ldots & \Omega_{N-1,N} \\ \vdots & \vdots & & \vdots \\ \Omega_{r-m,1} & \Omega_{r-m,2} & \ldots & \Omega_{r-m,N} \end{bmatrix} \quad (7.33)
$$

with $\underline{\hat{f}}(u)$ an estimate for

$$\underline{f}(u) = f(u) - g, \tag{7.34}$$

and g a constant vector such that:

$$\sum_{j=0}^{m} B_j g = \begin{bmatrix} d_1 \\ \vdots \\ d_{n_y} \end{bmatrix}. \tag{7.35}$$

Estimates for $\underline{f}$ and the $B_j, j = 0, \ldots, m$, can be obtained through a rank-n_u approximation of the right hand side of (7.33). If a singular value decomposition is used, the resulting columns of the left hand side matrix of (7.33) containing the elements of $B_j, j = 0, \ldots, m$, can be made orthonormal, effectively fixing the choice of V in (7.25).

From estimates for $\underline{f}$ in (7.34) and g in (7.35), finally, an estimate for the non-linear function f can be obtained. Note that if the row-rank of $\sum_{j=0}^{m} B_j$ is smaller than the column-rank, multiple choices for g are possible. This results as an inherent property of blind MIMO Hammerstein identification. The choice of a particular g is left to the user.

7.6 Comparison with existing overparameterization algorithms

As was mentioned in Subsection 7.4.1, the presented technique is closely related to the overparameterization approach [12,20]. Remember from Subsection 6.2.3 that the idea of overparameterization can be summarized as writing the static non-linearity f as a linear combination of general non-linear basis functions f_k, $f(u_t) = \sum_{k=1}^{n_f} c_k f_k(u_t)$, where-after a least squares regression problem is solved in parameters $\theta_{j,k} = b_j c_k$. The original system parameters are then recovered by means of an SVD of the matrix (6.3). It was also seen in Subsection 6.2.3 that in principle, some additional measures need to be taken in order to avoid loosing the rank-one property of (6.3). The approach proposed in Subsection 6.2.3 is to first calculate:

$$A = \begin{bmatrix} \hat{\theta}_{0,1} & \hat{\theta}_{0,2} & \ldots & \hat{\theta}_{0,n_f} \\ \hat{\theta}_{1,1} & \hat{\theta}_{1,2} & \ldots & \hat{\theta}_{1,n_f} \\ \vdots & \vdots & & \vdots \\ \hat{\theta}_{m,1} & \hat{\theta}_{m,2} & \ldots & \hat{\theta}_{m,n_f} \end{bmatrix} \begin{bmatrix} f_1(u_1) & \ldots & f_1(u_N) \\ f_2(u_1) & \ldots & f_2(u_N) \\ \vdots & & \vdots \\ f_{n_f}(u_1) & \ldots & f_{n_f}(u_N) \end{bmatrix},$$

with $u_t, t = 1, \ldots, N$ the inputs of the system. Then we subtract the mean of every row in A and take the SVD of the remaining matrix. From the SVD estimates for the b_j can be extracted. Estimates for the c_k can then be found in a second round by solving (6.1). It is this approach that will be used for the implementation of the classical overparameterization methods in the following section. Note that this approach amounts to setting the mean of $\hat{f} = \sum_{k=1}^{N} \hat{f}_k$

over the inputs $u_1, \dots, u_N$ to zero, which is similar to what was done for the LS-SVM, with the exception that in the latter case this constraint was explicitly introduced in the Lagrangian (7.21).

7.7 Illustrative examples

7.7.1 SISO system

The algorithm proposed in Section 7.4 was used for identification on the following SISO Hammerstein system:

$$A(z)y = B(z)f(u) + e, \tag{7.36}$$

with A and B polynomials in the forward shift operator z where $B(z) = z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3$, $A(z) = (z - 0.98e^{\pm i})(z - 0.98e^{\pm 1.6i})(z - 0.97e^{\pm 0.4i})$, and $f : \mathbb{R} \rightarrow \mathbb{R} : f(u) = \mathrm{sinc}(u)u^2$ the static non-linearity.

A white Gaussian input sequence u with length 400, zero mean and standard deviation 2 was generated and fed into the system (7.36). During the simulation the equation noise e was chosen white Gaussian with zero mean and as standard deviation 10% of the standard deviation of the sequence $f(u)$. The last 200 data-points of u and the generated output y were used for identification using the following three techniques:

- **LS-SVM:** The LS-SVM estimation procedure as described in Section 7.4: The linear system (7.20) is solved for d, a, α, β. An SVD of the right hand side of (7.22) is thereafter performed to obtain estimates for the linear system and the static non-linearity. For the example, an RBF-kernel with $\sigma = 1$ was used. Different values for the regularization parameter γ were tested by applying the obtained model to an independent validation sequence. From these tests $\gamma = 500$ was selected as the best candidate.

- **Hermite:** The overparameterization algorithm described in Subsection 6.2.3 with $f_k(u) = e^{u^2}(d^{k-1}/du^{k-1})e^{-u^2}$, the Hermite polynomial of order $k - 1$. This expansion was used in [70] for Hammerstein and in [71] for Wiener systems.

- **RBF network (Gaussian):** The general algorithm described in Subsection 6.2.3 with $f_k(\cdot), k = 1, \dots, n_f$ localized Gaussian density functions with mean depending on the value of k. As no prior information about the nature of the static non-linearity is assumed during the identification step, the means of the Gaussian non-linearities were chosen equidistantly spread between -4 and 4. The variance of the density functions was chosen to be 1, in line with the $\sigma = 1$ choice for LS-SVM. The main reason for considering this algorithm is that it is a parametric counterpart to the LS-SVM approach with an RBF-kernel, where the final solution is expressed as a sum of Gaussian density functions around the training data-points.

100 Monte-Carlo experiments were performed following the description above with $n = 6, m = 3$. For each experiment and each obtained estimate $\hat{f}$ for the static non-linearity f, the distance $d = \int_{-4}^{4} \|f(x) - \hat{f}(x)\| dx$ was calculated. The mean and variance of the distances so obtained using the LS-SVM technique are compared to those obtained from the Hermite and Gaussian approach using different values for n_f. The results are displayed in Table 7.1. Note that the LS-SVM technique clearly performs better than the Hermite-approach and about 3 times better than the Gaussian approach. The Gaussian and the LS-SVM technique are similar in nature as in both cases the estimated non-linearity is written as a sum of Gaussian basis functions with fixed bandwidth 1. However, it should be noted at this point that the RBF-kernel is but one possible choice in the LS-SVM algorithm, and that in principle any positive definite kernel can be chosen. A big disadvantage for the Gaussian approach is that it suffers from over-fitting once the parameter n_f is chosen too high, even though with the 200 data-points available and $n = 6, m = 3$, one could easily go to $n_f = 46$ before the resulting set of linear equations becomes under-determined. To avoid the increased variance, an extra regularization term $\gamma^{-1} \sum_{k=1}^{n_f} \sum_{j=0}^{n} \theta_{j,k}^2$ can be applied to the estimation problem (6.2). Results for the Gaussian approach including such a regularization term, and with $n_f = 46$, are displayed in Table 7.2. Note that the performance of the Gaussian estimator has drastically improved, but is still about 50% worse than the LS-SVM estimator. The same observation can be made by looking at Figure 7.1, where the true non-linearity is displayed together with the estimated non-linearities for the 3 alternative methods described in this section. 90% error bounds following from the Monte Carlo simulation are also displayed. The estimated linear systems for the LS-SVM and the Gaussian case with regularization (with $\gamma = 10^{11}$ obtained using validation on an independent dataset) are compared in Figure 7.2. Again, LS-SVM is seen to perform better than the Gaussian approach.

Method	mean(d)	std(d)
LS-SVM $\gamma = 500$	0.0064	0.0041
Hermite $n_f = 15$	0.2203	0.7842
Hermite $n_f = 20$	0.7241	2.3065
Hermite $n_f = 25$	1.1217	2.9660
Hermite $n_f = 30$	1.0118	2.9169
Gaussian $n_f = 18$	0.0142	0.0141
Gaussian $n_f = 24$	0.0193	0.1055
Gaussian $n_f = 30$	0.0168	0.0693
Gaussian $n_f = 36$	0.0188	0.0764

Table 7.1: Mean and standard deviation of obtained distances between estimated and true non-linearities in a SISO example.

Method	mean(d)	std(d)
LS-SVM $\gamma = 500$	0.0064	0.0041
Gaussian $\gamma = 10^{13}$	0.0457	0.1028
Gaussian $\gamma = 10^{12}$	0.0089	0.0071
Gaussian $\gamma = 10^{11}$	0.0088	0.0060
Gaussian $\gamma = 10^{10}$	0.0112	0.0086

Table 7.2: Mean and variances of obtained distances between estimated and true non-linearities in a SISO example.

7.7.2 MIMO system

In a second example, the MIMO identification method proposed in Section 7.5 was applied to a 2×2 MIMO system with a static non-linearity involving saturation and a saddle point. The MIMO system used is:

$$y = \begin{bmatrix} \frac{b_1(z)}{a_1(z)} & \frac{b_2(z)}{a_1(z)} \\ \frac{b_1(z)}{a_2(z)} & \frac{b_2(z)}{a_2(z)} \end{bmatrix} f(u) + \begin{bmatrix} \frac{1}{a_1(z)} \\ \frac{1}{a_2(z)} \end{bmatrix} e \tag{7.37}$$

with

$$\begin{aligned} a_1(z) &= (z - 0.98e^{\pm i})(z - 0.98e^{\pm 1.6i})(z - 0.97e^{\pm 0.4i}), \\ a_2(z) &= (z - 0.97e^{\pm 0.7i})(z - 0.98e^{\pm 1.4i})(z - 0.97e^{\pm 2.3i}), \\ b_1(z) &= z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3, \\ b_2(z) &= z^6 + 0.9z^5 + 0.7z^4 + 0.2z^3, \\ f(u) &= \begin{bmatrix} -\arctan(u(1)) \arctan(u(2)) \\ \arctan(u(1)) - \arctan(u(2)) \end{bmatrix}. \end{aligned}$$

A two-component zero mean white Gaussian input sequence u with length 500 and standard deviation 1 was generated and fed into the system (7.37). During the simulation the two components of the equation noise were chosen mutually uncorrelated white Gaussian with zero mean and standard deviation 0.1. Based on u and the obtained output y, estimates for $a_1(z), a_2(z), b_1(z), b_2(z)$ and f are obtained using the MIMO Hammerstein identification algorithm as described in Subsection 7.7.2, whereby $n = 6, m = 3$, and a classical linear ARX algorithm with the same orders for the numerator and the denominator. The hyper-parameters in the LS-SVM approach were chosen as $\sigma = 1, \gamma_1 = \gamma_2 = 300$, using 10-fold cross-validation.

The results of a simulation on an independent test-set using the obtained linear and Hammerstein model is shown in Figure 7.3. The results for the LS-SVM Hammerstein estimator are clearly better than those for the linear ARX estimator.

Note further that in the examples shown, m and n were considered to be known. In practical applications this will often not be the case. As in

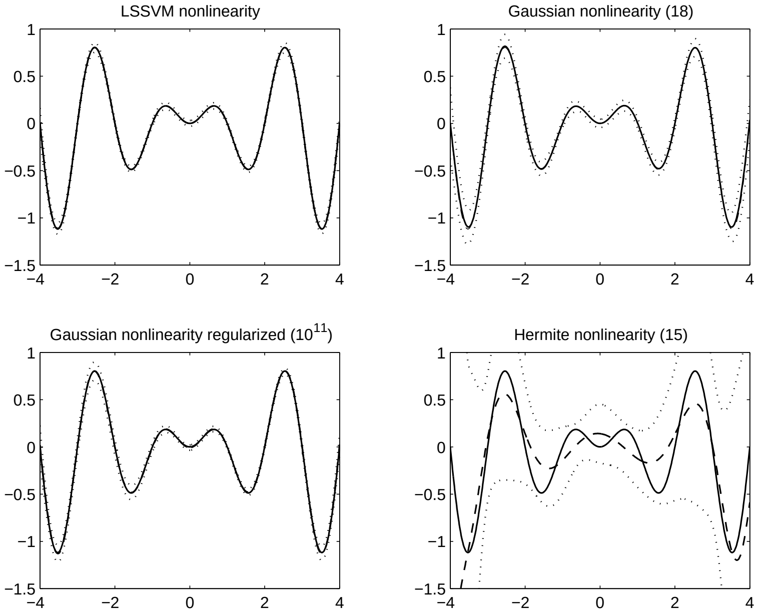


Figure 7.1: True non-linearity (solid) and mean estimated non-linearity (dashed) for the different techniques compared in a Monte-carlo simulation of a SISO system. Results for the LS-SVM algorithm with $\gamma = 500$ are displayed in the top-left figure, those for the Gaussian approach with $n_f = 18$ and without regularization in the top-right figure. The bottom-left figure displays the results for the Gaussian algorithm with $n_f = 46$ and constant $\gamma = 10^{11}$ tuned using validation on an independent dataset. The bottom-right figure displays the results for the Hermitian algorithm with $n_f = 15$. 90% error bounds on the estimated non-linearities, following from the Monte Carlo simulation, are included in each plot (dotted). The Hermite-approach is obviously inferior to the Gaussian and the LS-SVM technique. The best performance is obtained by using the LS-SVM algorithm.

linear identification problems, several identification runs followed by a selection of the best performing model (e.g. on a validation set) might therefore be necessary to obtain the best possible model. As a general rule, however, a slight overestimation of m and n is, as in the linear case, not a problem.

7.8 Conclusions

In this chapter, a technique for the identification of MIMO Hammerstein ARX systems was proposed. The method is based on Least Squares Support Vector Machines function approximation and allows to determine the memoryless

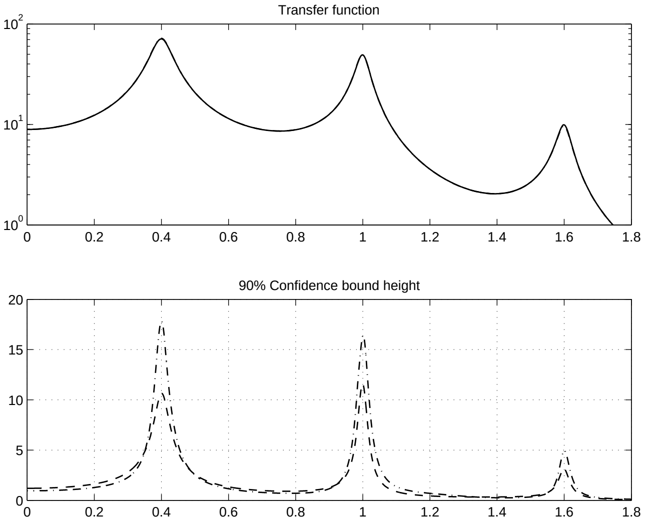
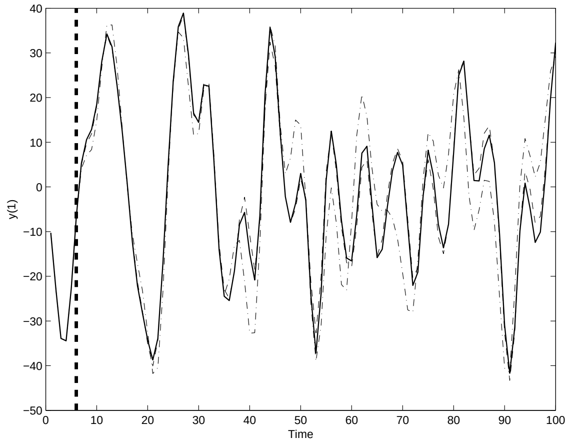
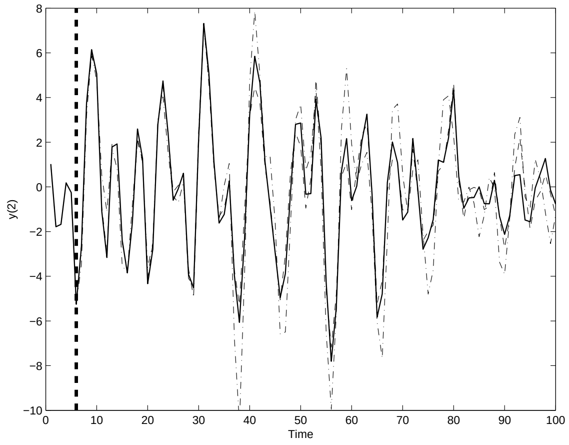


Figure 7.2: True transfer function (solid) and mean estimated ones for the LS-SVM estimator (dashed) and the Gaussian estimator with regularization (dash-dotted) in a Monte-carlo simulation of a SISO system (top-figure). The width of the 90% error bounds on the estimated transfer function (in log norm and obtained from the Monte Carlo simulations) is included in the plot below. Note that the transfer functions are visually indistinguishable in the top-figure, but the width of the error bounds clearly shows the significantly improved behavior of the LS-SVM approach.

static non-linearity as well as the linear model parameters from a linear set of equations. The method was compared to results of two other Hammerstein identification algorithms to illustrate its performance. This combined with the straightforward derivation of the results, the availability of a strong regularization framework [127, 135, 150], and the freedom that one gets in modeling the non-linearity by the design of an appropriate positive definite kernel makes the proposed technique an excellent candidate for Hammerstein model identification.



(a)



(b)

Figure 7.3: Simulation on an independent test-set (full line) using an LS-SVM Hammerstein estimator (dashed line) and a linear ARX estimator (dash-dotted line), identified on a MIMO example. The first component of the output is displayed in (a), the second component in (b). All simulations are initialized using the first 6 input/output measurements on the test set (at the left of the dashed vertical line).

Chapter 8

Hammerstein N4SID identification

In this chapter, a method for the identification of multi-input/multi-output Hammerstein systems is presented. The method extends the N4SID linear subspace identification algorithm, mainly by rewriting the oblique projection in the N4SID algorithm as a set of component-wise LS-SVM regression problems. The linear model and static non-linearities are obtained from a low rank approximation of a matrix produced by this regression problem.

8.1 Introduction

The primal-dual optimization framework characterizing LS-SVMs and the particularly simple (analytical) form of the solution have been shown to be well suited for the estimation of ARX Hammerstein models in Chapter 7 because linear structure and constraints can be incorporated easily in the optimization framework. However, ARX models are not suited for identification of linear dynamical systems under certain experimental conditions, such as the ones discussed in Subsection 5.2.2. Furthermore, the ARX model class is by construction a restricted one. To this extent it would be preferable to extend the use of subspace identification methods to Hammerstein systems. In this chapter, we investigate the use of LS-SVMs for the estimation of Hammerstein models using extended versions of the classical N4SID subspace algorithm as presented in [146, 147].

Following the notation and definitions in Chapter 3, we recall that the basic projection at the heart of the N4SID algorithm is the oblique projection of the future outputs along the future inputs onto the past.

$$\left\{ \begin{array}{rcl} \mathcal{O}_i & = & Y_f /_{U_f} W_p, \\ \mathcal{O}_{i+1} & = & Y_f^- /_{U_f^-} W_p^+, \end{array} \right. \tag{8.1}$$

As was seen in Section 5.2, this projection can be implemented using a least squares algorithm of the following form:

$$\begin{aligned}
(\widehat{L}_u, \widehat{L}_y) &= \arg\min_{L_u, L_y} \left\| \left[\begin{array}{c|c} L_u & L_y \end{array}\right] \left[\begin{array}{c} U_p \\ U_f \\ \hline Y_p \end{array}\right] - Y_f \right\|_F^2 , \\
(\widehat{L}_u^-, \widehat{L}_y^-) &= \arg\min_{L_u^-, L_y^-} \left\| \left[\begin{array}{c|c} L_u^- & L_y^- \end{array}\right] \left[\begin{array}{c} U_p^+ \\ U_f^- \\ \hline Y_p^+ \end{array}\right] - Y_f^- \right\|_F^2 , \\
\mathcal{O}_i &= \widehat{L}_u(:, 1 : im)U_p + L_y Y_p, \\
\mathcal{O}_{i+1} &= \widehat{L}_u^-(:, 1 : (i+1)m)U_p^+ + L_y^- Y_p^+,
\end{aligned}$$

where a slightly different notation and ordering in the regression matrix was used as in Section 5.2 to facilitate the derivations in the following sections. Once $\mathcal{O}_i$ and $\mathcal{O}_{i+1}$ are known, the system matrices A, B, C and D can be obtained using any of the algorithms described in Subsection 3.5.6. In this chapter, we will consider the biased state-space algorithm summarized in Figure 3.7, and show that it can conveniently be extended towards the identification of Hammerstein systems. The reason for choosing the biased algorithm is its straightforward form, keeping the derivations both focused and insightful. We note that the principles of the derivations given in this chapter can equally well be applied to any other N4SID subspace identification algorithm, such as the unbiased version presented in [144, 147].

8.2 Extending the N4SID algorithm towards identification of Hammerstein systems

A linear system is transformed into a Hammerstein system by introducing a static non-linearity $f : \mathbb{R}^m \to \mathbb{R}^m$ which is applied to the inputs u. With the introduction of the non-linearity f, the following state-space model is obtained:

$$\left\{ \begin{aligned} x_{t+1} &= Ax_t + Bf(u_t) + w_t, \\ y_t &= Cx_t + Df(u_t) + v_t. \end{aligned} \right. \tag{8.2}$$

Inputs and outputs $\{(u_t, y_t)\}_{t=0}^{N-1}$, are assumed to be available. The process and measurement noise w_t and v_t follow the same statistics as in Chapter 3. We define the matrix operator Φ as an operator on a block Hankel matrix and a nonlinear function ρ on $\mathbb{R}^m$ which applies $\rho(\cdot)$ to every block matrix Z_i in Z and stacks the results in the original Hankel configuration:

$$\Phi_\rho \left(\begin{bmatrix} Z_1 & Z_2 & \dots & Z_p \\ Z_2 & Z_3 & \dots & Z_{p+1} \\ \vdots & \vdots & & \vdots \\ Z_q & Z_q + 1 & \dots & Z_{p+q-1} \end{bmatrix} \right) = \begin{bmatrix} \rho(Z_1) & \rho(Z_2) & \dots & \rho(Z_p) \\ \rho(Z_2) & \rho(Z_3) & \dots & \rho(Z_{p+1}) \\ \vdots & \vdots & & \vdots \\ \rho(Z_q) & \rho(Z_q + 1) & \dots & \rho(Z_{p+q-1}) \end{bmatrix}.$$

8.2.1 Overparameterization for the oblique projection $\mathcal{O}_i$

The oblique projection $\mathcal{O}_i = Y_f/_{U_f} W_p$ can be calculated from estimates for L_u, L_y and f obtained by minimizing the residuals E of the following equation [147]:

$$Y_f = \begin{bmatrix} L_u & L_y \end{bmatrix} \begin{bmatrix} \Phi_f(U_{0|2i-1}) \\ Y_p \end{bmatrix} + E, \tag{8.3}$$

in a least-squares sense. This can be rewritten as

$$Y_f(s,t) = L_y(s,:)Y_p(:,t) + \sum_{h=1}^{2i} L_u(s,(h-1)m+1:hm) f(u_{h+t-2}) + E(s,t), \tag{8.4}$$

for $s = 1,\ldots,il$ and $t = 1,\ldots,j$. Once estimates for $\widehat{L}_u$, $\widehat{L}_y$ and $\hat{f}$ occuring in equations (8.3) and (8.4) are obtained, the oblique projection is calculated as:

$$\mathcal{O}_i(s,t) = \widehat{L}_y(s,:)Y_p(:,t) + \sum_{h=1}^{i} \widehat{L}_u(s,(h-1)m+1:hm)\hat{f}(u_{h+t-2}), \tag{8.5}$$

for $s = 1,\ldots,il$ and $t = 1,\ldots,j$. Note that in (8.4) and (8.5), products between parameter matrices L_u and L_y and the static non-linearity f appear which were already seen to lead to a difficult non-convex optimization problem in Chapter 7. Again we will use an LS-SVM approach, combined with ideas from the overparameterization technique by introducing a set of functions $g_{h,s} : \mathbb{R}^m \rightarrow \mathbb{R}$ such that [12]:

$$g_{h,s} \triangleq c_{h,s}^T f, \quad \text{s.t.} \quad \begin{aligned} & c_{h,s}^T = L_u(s,(h-1)m+1:hm), \\ & \forall h = 1,\ldots,2i, \;\; s = 1,\ldots,il. \end{aligned} \tag{8.6}$$

With these new functions we obtain a generalization to (8.4) and (8.5):

$$\begin{aligned} Y_f(s,t) &= L_y(s,:)Y_p(:,t) + \sum_{h=1}^{2i} g_{h,s}(u_{h+t-2}) + E(s,t), && (8.7) \\ \mathcal{O}_i(s,t) &= \widehat{L}_y(s,:)Y_p(:,t) + \sum_{h=1}^{i} \hat{g}_{h,s}(u_{h+t-2}), && (8.8) \end{aligned}$$

for $s = 1,\ldots,il$ and $t = 1,\ldots,j$. Note that (8.7) is now linear in the functions $g_{h,s} : \mathbb{R}^m \rightarrow \mathbb{R}$. In line with the approach in Chapter 7, the central idea behind the algorithm presented in this chapter is that the functions $g_{h,s}$ in (8.7) can be determined from data using the concept of component-wise LS-SVM regression as presented in [117]. Once estimates for the $g_{h,s}$ are obtained, Equation (8.6) will be used in combination with a rank reduction technique such as the singular value decomposition to obtain estimates for the non-linear functions f and the elements in L_u, similar to the singular value decomposition step in classical overparameterization algorithms (see Subsection 6.2.3).

Let the kernel function be defined as $K : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ such that $K(u_p, u_q) = \varphi(u_p)^T \varphi_k(u_q)$ for all $p, q = 0, \ldots, N-1$ and the kernel matrix $\Omega \in \mathbb{R}^{N \times N}$ such that $\Omega(i, j) = K(u_{i-1}, u_{j-1})$ for all $i = 1, \ldots, N,\ j = 1, \ldots, N$. Substituting $g_{h,s}$ for the primal model $w_{h,s}^T \varphi$ in (8.7) results in

$$\begin{aligned} Y_f(s,t) &= L_y(s,:)Y_p(:,t) + \sum_{h=1}^{2i} w_{h,s}^T \varphi(u_{h+t-2}) + E(s,t), \\ & \quad \forall s = 1, \ldots, li,\ \ t = 1, \ldots, j. \end{aligned} \tag{8.9}$$

Introducing centering

As argued in 6.2.3, the expansion of a non-linear function as the sum of a set of non-linear functions is not unique, e.g.

$$\left(w_1^T \varphi_1(u)\right) + \left(w_2^T \varphi_2(u)\right) = \left(w_1^T \varphi_1(u) + \delta\right) + \left(w_2^T \varphi_2(u) - \delta\right),$$

for all $\delta \in \mathbb{R}$. It was seen in Chapter 7 that this problem can be avoided by including a centering constraint of the form

$$\sum_{t=0}^{N-1} f(u_t) = 0. \tag{8.10}$$

This constraint can always be applied since for any constant δ_u, and any function $\underline{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ such that $f = \underline{f} + \delta_u$ there exists a state transformation $\xi_t = \Psi(x_t)$ with $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and a constant δ_y such that (8.2) is transformed as follows:

$$\left\{ \begin{aligned} \xi_{t+1} &= A\xi_t + B\underline{f}(u_t) + \nu_t, \\ y_t - \delta_y &= C\xi_t + D\underline{f}(u_t) + v_t, \end{aligned} \right. \tag{8.11}$$

with $\xi_t \in \mathbb{R}^n$ and $\delta_y \in \mathbb{R}^l$ defined as

$$\left\{ \begin{aligned} \xi_t &= \Psi(x_t) = x_t - (I - A)^{-1} B \delta_u, \\ \delta_y &= \left(C(I - A)^{-1} B + D\right) \delta_u. \end{aligned} \right.$$

Hence, the constraint (8.10) can be applied provided that a new parameter δ_y is added to the model, transforming (8.9) into

$$\begin{aligned} Y_f(s,t) + [1_i \otimes \delta_y](s) &= L_y(s,:)(Y_p(:,t) + 1_i \otimes \delta_y) + \sum_{h=1}^{2i} w_{h,s}^T \varphi(u_{h+t-2}) \\ &+ \ E(s,t),\ \ \forall s = 1, \ldots, li,\ \ t = 1, \ldots, j, \end{aligned}$$

where $\otimes$ denotes the matrix kronecker product. Through the equality $w_{h,s}^T \varphi = g_{h,s} = c_{h,s}^T f$ for all $h = 1, \ldots, 2i,\ \ s = 1, \ldots, il$, the constraint (8.10) amounts to

$$\sum_{t=0}^{N-1} w_{h,s}^T \varphi(u_t) = 0,\ \ \forall h, s.$$

The LS-SVM primal problem is then formulated as a constrained optimization problem:

$$
\min_{w_{h,s},L_y,E,\delta_y} \mathcal{J}(w_{h,s},L_y,E,\delta_y) = \frac{1}{2}\sum_{s=1}^{il}\sum_{h=1}^{2i} w_{h,s}^T w_{h,s} + \frac{\gamma}{2}\sum_{s=1}^{il}\sum_{t=1}^{j} E(s,t)^2,
$$
$$
\text{s.t.} \begin{cases} Y_f(s,t) + [1_i \otimes \delta_y](s) = L_y(s,:)(Y_p(:,t) + 1_i \otimes \delta_y) & (a) \\ \quad + \sum_{h=1}^{2i} w_{h,s}^T \varphi(u_{h+t-2}) + E(s,t), & \\ \qquad \forall s = 1,\dots,il,\ t = 1,\dots,j, & \\ \sum_{t=0}^{N-1} w_{h,s}^T \varphi(u_t) = 0, & (b) \\ \qquad \forall h = 1,\dots,2i, s = 1,\dots,li. & \end{cases} \tag{8.12}
$$

Solving the componentwise LS-SVM regression problem

Lemma 8.1. *Given the primal problem (8.12), estimates for L_y and δ_y follow from the dual system:*

$$
\left[\begin{array}{c|c|c|c} 0 & 0 & 1^T & 0 \\ \hline 0 & 0 & Y_p & 0 \\ \hline 1 & Y_p^T & \mathcal{K}_p + \mathcal{K}_f + \gamma^{-1} I & \mathcal{S} \\ \hline 0 & 0 & \mathcal{S}^T & \mathcal{T} \end{array}\right] \left[\begin{array}{c} \overline{d} \\ \hline L_y^T \\ \hline \mathcal{A} \\ \hline \mathcal{B} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline Y_f^T \\ \hline 0 \end{array}\right],
$$

where $\overline{d} = (1_i \otimes I_l - L_y(1_i \otimes I_l))\,\delta_y$, 1_j is a column vector of length j with elements 1, $\mathcal{T} = I_{2i} \times 1_N^T \Omega^1 1_N$, $\mathcal{S}_q = \sum_{t=1}^{N} \Omega(t,q)$ and

$$
\mathcal{A} = \begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} & \dots & \alpha_{li,1} \\ \alpha_{1,2} & \alpha_{2,2} & \dots & \alpha_{li,2} \\ \vdots & \vdots & & \vdots \\ \alpha_{1,j} & \alpha_{2,j} & \dots & \alpha_{li,j} \end{bmatrix}, \mathcal{B} = \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,li} \\ \beta_{2,1} & \beta_{2,2} & \dots & \beta_{2,li} \\ \vdots & \vdots & & \vdots \\ \beta_{2i,1} & \beta_{2i,2} & \dots & \beta_{2i,li} \end{bmatrix},
$$

$$
\mathcal{S} = \begin{bmatrix} \mathcal{S}_1 & \mathcal{S}_2 & \dots & \mathcal{S}_{2i} \\ \mathcal{S}_2 & \mathcal{S}_3 & \dots & \mathcal{S}_{2i+1} \\ \vdots & \vdots & & \vdots \\ \mathcal{S}_j & \mathcal{S}_{j+1} & \dots & \mathcal{S}_N \end{bmatrix}.
$$

The matrices $\mathcal{K}_p \in \mathbb{R}^{j \times j}$ and $\mathcal{K}_f \in \mathbb{R}^{j \times j}$ have elements:

$$
\mathcal{K}_p(p,q) = \sum_{h=1}^{i} K(u_{h+p-2}, u_{h+q-2}),\ \ \mathcal{K}_f(p,q) = \sum_{h=i+1}^{2i} K(u_{h+p-2}, u_{h+q-2}),
$$

for all $p,q = 1,\dots,j$. Estimates for the $g_{h,s}$ in (8.7) for all h,s are given as:

$$
\hat{g}_{h,s} : \mathbb{R}^m \to \mathbb{R} : u^* \to \sum_{t=1}^{j} \alpha_{s,t} K(u_{h+t-2}, u^*) + \beta_{h,s} \sum_{t=0}^{N-1} K(u_t, u^*). \tag{8.13}
$$

Proof. This directly follows from the Lagrangian

$$\begin{aligned}\mathcal{L}(w, d, L_y, E; \alpha, \beta, d_s) = \mathcal{J}(w, E) - \sum_{h=1}^{2i}\sum_{s=1}^{li} \beta_{h,s} \left\{ \sum_{t=0}^{N-1} w_{h,s}^T \varphi(u_t) \right\} - \\ \sum_{s=1}^{li}\sum_{t=1}^{j} \alpha_{s,t} \left\{ L_y(s,:) Y_p(:,t) + \sum_{h=1}^{2i} w_{h,s}^T \varphi(u_{h+t-2}) + d_s + E(s,t) - Y_f(s,t) \right\},\end{aligned}$$

with $\overline{d} = \begin{bmatrix} d_1^T & \dots & d_l^T \end{bmatrix}^T$ by taking the conditions for optimality $\frac{\partial \mathcal{L}}{\partial w_{h,s}} = 0$, $\frac{\partial \mathcal{L}}{\partial L_y(s,:)} = 0$, $\frac{\partial \mathcal{L}}{\partial E(s,t)} = 0$, $\frac{\partial \mathcal{L}}{\partial d_s} = 0$, $\frac{\partial \mathcal{L}}{\partial \alpha_{s,t}} = 0$, $\frac{\partial \mathcal{L}}{\partial \beta_{h,s,k}} = 0$, $\frac{\partial \mathcal{L}}{\partial d_s} = 0$ and after elimination of the primal variables $w_{h,s}$ and E. □

Combining the results from Lemma (8.1) with equation (8.8), we have

$$\begin{aligned}\mathcal{O}_i &= \sum_{h=1}^{i} \left(\begin{bmatrix} \Phi_{\hat{g}_{h,1}} U_{h|h} \\ \Phi_{\hat{g}_{h,2}} U_{h|h} \\ \vdots \\ \Phi_{\hat{g}_{h,2li}} U_{h|h} \end{bmatrix} \right) + \widehat{L}_y \left(Y_p - 1_{li} 1_{li}^T \otimes \hat{\delta}_y \right) \\ &= \mathcal{A}^T \mathcal{K}_p + \mathcal{B}_p^T \mathcal{S}_p^T + \widehat{L}_y \left(Y_p - (1_i 1_j^T) \otimes \hat{\delta}_y \right), \qquad (8.14)\end{aligned}$$

with $\mathcal{B}_p = \mathcal{B}(1:i,:)$ and $\mathcal{S}_p = \mathcal{S}(:,1:i)$.

8.2.2 Calculating the oblique projection $\mathcal{O}_{i+1}$

The calculation of $\mathcal{O}_{i+1}$ is entirely equivalent to that of $\mathcal{O}_i$. Without further proof, we state that $\mathcal{O}_{i+1}$ is obtained as:

$$\mathcal{O}_{i+1} = (\mathcal{A}^-)^T (\mathcal{K}_p^+)^T + (\mathcal{B}_p^-)^T (\mathcal{S}_p^-)^T + \widehat{L}_y^- \left(Y_p^+ - 1_{(i+1)} 1_j^T \otimes \delta_y \right) \qquad (8.15)$$

with $\mathcal{K}_p^+(p,q) = \sum_{h=1}^{i+1} K(u_{h+p-2}, u_{h+q-2})$ for all $p, q = 1, \dots, j$, and

$$\begin{aligned}\mathcal{B}_p^- &= \mathcal{B}^-(1:i+1,:), \\ \mathcal{S}_p^- &= \mathcal{S}^-(:,1:i+1).\end{aligned}$$

$\mathcal{A}^-, \mathcal{B}^-$ and $\widehat{L}_y^-$ follow from:

$$\left[\begin{array}{c|c|c|c} 0 & 0 & 1^T & 0 \\ \hline 0 & 0 & Y_p^+ & 0 \\ \hline 1 & (Y_p^+)^T & \mathcal{K}_p + \mathcal{K}_f + \gamma^{-1} I & \mathcal{S} \\ \hline 0 & 0 & \mathcal{S}^T & \mathcal{T} \end{array}\right] \left[\begin{array}{c} \overline{d^-} \\ \hline (L_y^-)^T \\ \hline \mathcal{A}^- \\ \hline \mathcal{B}^- \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline (Y_f^-)^T \\ \hline 0 \end{array}\right],$$

with

$$\overline{d^-} = \left((1_{(i-1)} \otimes I_l) - L_y^- (1_{(i+1)} \otimes I_l) \right) \delta_y.$$

8.2.3 Obtaining estimates for the states

The state sequences $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ can now be determined from $\mathcal{O}_i$ and $\mathcal{O}_{i+1}$ in line with what is done in linear subspace algorithms (see for instance Figure 3.7). These state sequences will be used in a second step of the algorithm to obtain estimates for the system matrices and the non-linearity f. Note that in the linear case, it is well known that the obtained state sequences $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ can be considered as the result of a bank of non steady state Kalman filters working in parallel on the columns of the block-Hankel matrix W_p [147]. In the Hammerstein case, and if f were known, this relation would still hold provided that W_p is replaced by $\begin{bmatrix}\Phi_f(U_p)^T & Y_p^T\end{bmatrix}^T$. However, an estimate $\hat{f}$ for f based on a finite amount of data will in general be subject to approximation errors [150]. As the classical results for the bank of linear Kalman filters are not applicable if the inputs $\hat{f}(u_t)$ to the linear model are not exact the obtained states $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ can no longer be seen as the result of a bank of Kalman filters working on $\begin{bmatrix}\Phi_{\hat{f}}(U_p)^T & Y_p^T\end{bmatrix}^T$. Despite the loss of this property, it will be illustrated in the examples that the proposed method outperforms existing Hammerstein approaches such as approaches based on ARX models and N4SID identification algorithms with an expansion in Hermite polynomials.

8.2.4 Extraction of the system matrices and the non-linearity f

The linear model and static non-linearity are estimated from:

$$(\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}, \hat{f}) = \underset{A,B,C,D,f}{\arg\min} \left\| \begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} - \delta_y \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \widetilde{X}_i \\ \Phi_f\left(U_{i|i}\right) \end{bmatrix} \right\|_F^2 . \quad (8.16)$$

It will be shown in this subsection that this least-squares problem can again be written as an LS-SVM regression problem. Denoting

$$\mathcal{X}_{i+1} = \begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} - \delta_y \end{bmatrix}, \; \Theta_{AC} = \begin{bmatrix} A \\ C \end{bmatrix}, \; \Theta_{BD} = \begin{bmatrix} B \\ D \end{bmatrix}, \quad (8.17)$$

and replacing $\Theta_{BD}(s,:)f$ by $\omega_s^T \varphi$, where again an expansion of a product of scalars and non-linear functions is written as a linear combination of non-linear functions, we have:

$$\mathcal{X}_{i+1} = \Theta_{AC} \widetilde{X}_i + \begin{bmatrix} \omega_1^T \\ \omega_2^T \\ \vdots \\ \omega_{n+l}^T \end{bmatrix} \Phi_\varphi(U_{i|i}) + E,$$

with E the residuals of (8.16). The resulting LS-SVM primal problem can be written as

$$\min_{\omega_s, E, \Theta_{AC}} \mathcal{J}(\omega, E) = \frac{1}{2}\sum_{s=1}^{n+l} \omega_s^T \omega_s + \frac{\gamma_{BD}}{2}\sum_{s=1}^{n+l}\sum_{t=1}^{j} E(s,t)^2,$$

$$\text{s.t.} \begin{cases} \mathcal{X}_{i+1}(s,t) = \Theta_{AC}(s,:)\widetilde{X}_i(:,t) + \omega_s^T \varphi(u_{i+t-1}), & (a) \\ \quad \forall s = 1, \ldots, li, \; t = 1, \ldots, j, & \\ \sum_{t=0}^{N-1} \omega_s^T \varphi(u_t) = 0, \; \forall s = 1, \ldots, li, & (b) \end{cases}$$

where γ_{BD} denotes a regularization constant which can be different from the γ used in Subsection 8.2.1.

Lemma 8.2. *Estimates for A and C in Θ_{AC} are obtained from the following dual problem*

$$\left[\begin{array}{c|c|c} 0 & \widetilde{X}_i & 0 \\ \hline \widetilde{X}_i^T & \mathcal{K}_{BD} + \gamma_{BD}^{-1} I & \mathcal{S}_{BD} \\ \hline 0 & \mathcal{S}_{BD}^T & \mathcal{T}_{BD} \end{array}\right] \left[\begin{array}{c} \Theta_{AC}^T \\ \hline \mathcal{A}_{BD} \\ \hline \mathcal{B}_{BD} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathcal{X}_{i+1}^T \\ \hline 0 \end{array}\right], \tag{8.18}$$

whereby $\omega_s = \sum_{t=1}^{j} \alpha_{s,t}\varphi(u_{i+t-1}) + \beta_s \sum_{t=0}^{N-1} \varphi(u_t)$ for all $s = 1, \ldots, n+l$, $\mathcal{K}_{BD}(p,q) = K(u_{i+p-1}, u_{i+q-1})$ for all $p, q = 1, \ldots, j$, $\mathcal{B}_{BD} = \begin{bmatrix}\beta_1 & \beta_2 & \ldots & \beta_{n+l}\end{bmatrix}$, $\mathcal{T}_{BD} = 1_N^T K 1_N$, and

$$\mathcal{A}_{BD} = \begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} & \ldots & \alpha_{n+l,1} \\ \alpha_{1,2} & \alpha_{2,2} & \ldots & \alpha_{n+l,2} \\ \vdots & \vdots & & \vdots \\ \alpha_{1,j} & \alpha_{2,j} & \ldots & \alpha_{n+l,j} \end{bmatrix}, \mathcal{S}_{BD} = \begin{bmatrix} \mathcal{S}_{i+1} \\ \mathcal{S}_{i+2} \\ \vdots \\ \mathcal{S}_{i+j} \end{bmatrix}.$$

Proof. This follows directly from the Lagrangian

$$\begin{aligned} \mathcal{L} = \mathcal{J}(\omega, E) &- \sum_{s=1}^{n+l} \beta_s \left\{ \sum_{t=0}^{N-1} \omega_s \varphi(u_t) \right\} \\ &- \sum_{s=1}^{n+l} \alpha_{s,t} \left\{ \mathcal{X}_{i+1}(s,t) - \Theta_{AC}(s,:)\widetilde{X}_i(:,t) - \omega_s^T \varphi(u_{i+t-1}) \right\}, \end{aligned}$$

by taking the conditions for optimality $\frac{\partial \mathcal{L}}{\partial \omega_s} = 0$, $\frac{\partial \mathcal{L}}{\partial E} = 0$, $\frac{\partial \mathcal{L}}{\partial \Theta_{AC}} = 0$, $\frac{\partial \mathcal{L}}{\partial \alpha_{s,t}} = 0$, $\frac{\partial \mathcal{L}}{\partial \beta_s} = 0$, and after elimination of the primal variables ω_s and E. □

By combining the results from Lemma 8.2 with (8.16) and (8.17), we have:

$$\Theta_{BD} \begin{bmatrix} f(u_0) & \ldots & f(u_{N-1}) \end{bmatrix} = \mathcal{A}_{BD}^T \Omega(i+1 : i+j, :) + \mathcal{B}_{BD}^T \sum_{t=1}^{N} \Omega(t,:). \tag{8.19}$$

Hence, estimates for B, D in Θ_{BD} and the non-linearity f can be obtained from a rank m approximation of the right hand side of (8.19), for instance using a singular value decomposition. This is a typical step in overparameterization approaches [12] and amounts to projecting the results for the overparameterized model as used in the estimation onto the class of Hammerstein models.

8.2.5 Practical implementation

Following the discussion in the previous sections, the final algorithm for Hammerstein N4SID subspace identification can be summarized as follows:

1. Find estimates for the oblique projections $\mathcal{O}_i$ and $\mathcal{O}_{i+1}$ from (8.14) and (8.15).
2. Find estimates for the state following the procedure outlined in Subsection 8.2.3.
3. Obtain estimates for A, C, $\mathcal{A}_{BD}$ and $\mathcal{B}_{BD}$ following the procedure outlined in Subsection 8.2.4.
4. Obtain estimates for B, D en f from a rank-m approximation of (8.19).

It should be noted at this point that given the fact that regularization is inherently present in the proposed identification technique, lack of persistency of excitation will not lead to any numerical problems. However, in order to ensure that all aspects of the linear system are properly identified, persistency of excitation of $f(u)$ of at least order $2i$ is desired (see also Subsection 3.5.5). Persistency of excitation of $f(u)$ can for some nonlinear functions f be expressed as a condition on the original inputs u but the relation is certainly not always straightforward (see for instance [156] for a discussion on this issue).

Furthermore, it is important to remark that the estimate of the static nonlinearity will only be reliable in regions where the input density is sufficiently high.

8.3 Illustrative examples

We consider the SISO system presented in Subsection 7.7.2, augmented with an output noise term ν:

$$A(z)(y+\nu) = B(z)f(u) + e. \tag{8.20}$$

As in Subsection 7.7.2, A and B are polynomials in the forward shift operator z where $B(z) = z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3$, $A(z) = (z - 0.98e^{\pm i})(z - 0.98e^{\pm 1.6i})(z - 0.97e^{\pm 0.4i})$, and $f : \mathbb{R} \to \mathbb{R} : f(u) = \mathrm{sinc}(u)u^2$ the static non-linearity. A dataset was generated from this system where $u_t \sim \mathcal{N}(0, 2)$ is a white Gaussian noise sequence for $t = 0, \ldots, N-1$ with $N = 1000$ and e_t is a sequence of Gaussian white noise with a level of 10% of the level of the non-linearity $f(u)$.

8.3.1 Comparison with the Hammerstein ARX approach in Chapter 7

Without measurement noise

The measurement noise terms ν_t were chosen to be zero for $t = 0, \ldots, N-1$, in which case the system (8.20) belongs to the class of Hammerstein ARX systems, identical to the one used as the primary example for the LS-SVM based Hammerstein ARX identification algorithm derived in Chapter 7. The latter was shown to outperform classical Hammerstein ARX identification methods based on orthogonal and non-orthogonal basis functions.

The Hammerstein N4SID subspace identification algorithm as derived in Section 8.2 was used to extract the linear model and the static non-linearity f from the dataset described above. The number of block-rows i in the block Hankel matrices was set to 10. An advantage of the N4SID algorithm is that the model order, 6, automatically follows from the SVD of $\mathcal{O}_i$. The hyper-parameters in the LS-SVM N4SID algorithm were selected as $\sigma = 0.1$, $\gamma = 1000$, $\gamma_{BD} = 10$ by validation on an independent validation set. The resulting linear system and static non-linearity are displayed in Figure 8.1.

As a comparison, the results of the LS-SVM ARX estimator from Chapter 7 are also displayed in Figure 8.1. For the ARX-estimator, the number of poles and zeros were assumed to be fixed a priori. Two hyper-parameters (the regularization constant and the bandwidth of the RBF kernel) which need to be set in this method were chosen in accordance with the choices reported in Chapter 7. One observes that in this particular case of no output noise, the ARX algorithm outperforms the N4SID. This can be attributed to the fact that a linear system without output noise belongs to the ARX class of models, which features less parameters than the wider class of models which can be identified using N4SID methods. Hence, the variance on the model parameters obtained using N4SID identification can be expected to be larger than those on the parameters obtained using ARX identification.

including measurement noise

In order to highlight the advantages of the N4SID algorithm, in a second example, a noise sequence $\nu_t \sim \mathcal{N}$ was added to the output such that a signal to noise ratio of 10 was obtained on the output signal. Again results for the N4SID and the Hammerstein ARX algorithm are reported. For the N4SID case, the hyper-parameters were now selected as $\sigma = 1, \gamma = 10, \gamma_{BD} = 1$, again via validation on an independent validation set. For the ARX case, the obtained optimal hyper-parameters were as in the previous subsection. The results are displayed in Figure 8.2. Note that the addition of output noise has had little impact on the performance of the N4SID-algorithm. The ARX algorithm on the other hand suffers from a bias due to the true model not belonging to the ARX model class and performs very badly in this case. This serves to highlight one of the main advantages of the use of subspace identification methods [147] over more classical ARX procedures, namely that they are not limited to a restricted class of linear models.

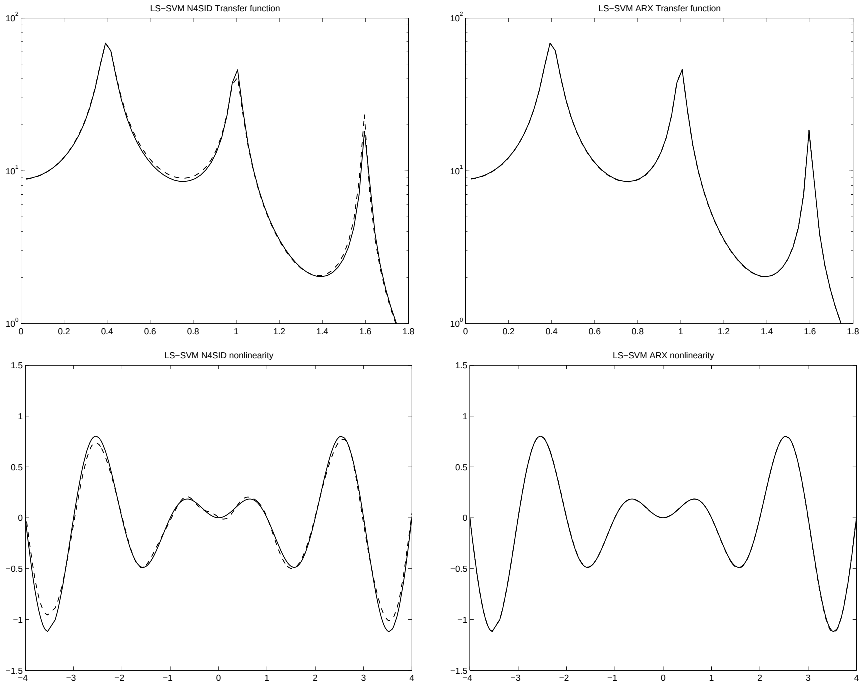


Figure 8.1: True transfer function (solid) and mean estimated ones (dashed) for the LS-SVM N4SID subspace algorithm (top-left) and the LS-SVM ARX algorithm (top-right), as estimated from a sequence of 1000 input/output measurements on a simulated system, without addition of output noise. The true non-linearities (solid) and estimated ones (dashed) are displayed below the transfer functions, for the N4SID case (lower-left), and the ARX-case (lower-right).

8.3.2 Comparison with classical subspace over-parameterization approaches

As mentioned before, a classical approach to Hammerstein system identification is to expand the static nonlinearity in a set of orthogonal or non-orthogonal basis-functions [114]. The same idea can be applied to subspace algorithms [104]. Once a set of basis-functions is considered, the one-dimensional input is transformed into a higher-dimensional input vector which contains the coefficients of the expansion of $f(u)$ in its basis. The classical N4SID subspace algorithm presented in Figure 3.7 is thereafter applied. The linear system and static nonlinearities can be obtained from the obtained matrices B and D (see [104] for a detailed procedure).

This example will adopt the common choice of the Hermite polynomials as a basis. The best results on the dataset with output noise were obtained when

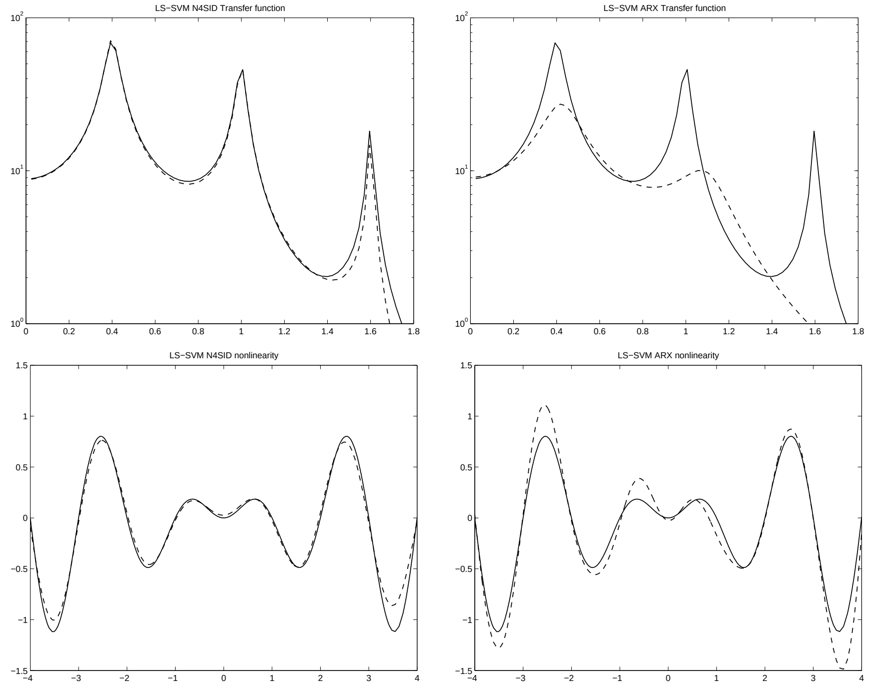


Figure 8.2: True transfer function (solid) and estimated one (dashed) for the LS-SVM N4SID subspace algorithm (top-left) and the LS-SVM ARX algorithm (top-right), as estimated from a sequence of 1000 input/output measurements on a simulated system, with the addition of 10% output noise. The true nonlinearities (solid) and estimated ones (dashed) are displayed below the transfer functions, for the N4SID case (lower-left), and the ARX-case (lower-right).

selecting 7 Hermite polynomials with orders ranging from 0 to 6. The obtained linear system corresponding to this choice of basis functions is displayed in Figure 8.3. Note the rather poor performance of this method, compared to the LS-SVM N4SID algorithm. This can largely be attributed to the fact that the performance of subspace algorithms degrades as the number of inputs increases, certainly if these inputs are highly correlated (see [25] and Chapter 5). This as a result of a bad conditioning of the matrices U_p and U_f as the number of rows increases and these rows get more correlated. For the 0^{th} order Hermite polynomial (which is a constant) this is certainly the case but also when leaving out this polynomial, condition numbers of 10^5 and higher are encountered. This problem does not occur in the N4SID LS-SVM algorithm as the latter features an inherently available regularization framework. An additional advantage is the flexibility one gets by plugging in an appropriate kernel and the fact that if localized kernels are used, no specific choices have to be made for their locations.

The locations follow directly from the formulation of costfunctions as (8.12).

8.4 Conclusions

In this chapter, a method for the identification of Hammerstein systems was presented based on the well-known N4SID subspace identification algorithm. The basic framework of the N4SID algorithm is largely left untouched, except for the ordinary least squares steps which is replaced by a set of component-wise LS-SVM regressions. The proposed algorithm was observed to be able to extract the linear system and the non-linearity from data, even in the presence of output noise.

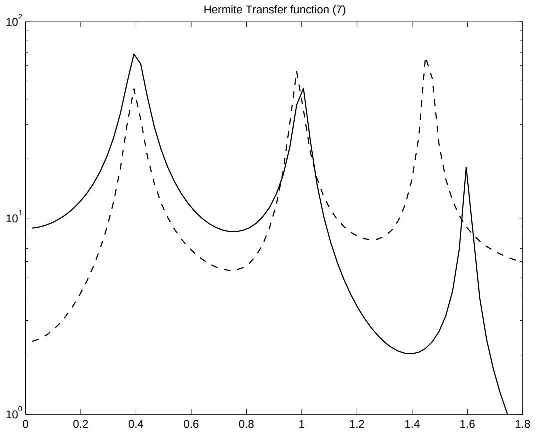


Figure 8.3: True transfer function (solid) and the estimated one (dashed) for the Hermite N4SID subspace algorithm as estimated from a sequence of 1000 input/output measurements on a simulated system, with the addition of 10% output noise.

Chapter 9

Hammerstein-Wiener identification using subspace intersection

In this chapter, a method for the identification of Hammerstein-Wiener systems is presented. The method extends the linear subspace intersection algorithm, mainly by introducing a Kernel Canonical Correlation Analysis (KCCA) to calculate the state as the intersection of past and future, in stead of the more classical CCA approach. The linear model and static non-linearities on input and output are readily obtained once the state is known using Least Squares Support Vector Machines (LS-SVM)-regression.

9.1 Introduction

Following the approach set out in the former chapters for the extension of ARX and N4SID identification algorithms to the identification of Hammerstein systems, in this chapter we will consider the extension of the classical subspace intersection algorithm (see 3.4.2) to Hammerstein-Wiener systems in state-space form:

$$\begin{cases} x_{t+1} = Ax_t + Bf(u_t), \\ g^{-1}(y_t) = Cx_t + Du_t. \end{cases} \tag{9.1}$$

Hereby $u_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}^l$ are the input and output at time t and $x_t \in \mathbb{R}^n$ denotes the state. $f : \mathbb{R}^m \to \mathbb{R}^m$ and $g : \mathbb{R}^l \to \mathbb{R}^l$ are static non-linear mappings with g such that g^{-1} exists for all possible outputs of the system. The extension will be obtained by replacing the linear CCA-step, used for the estimation of the state by a non-linear kernel CCA (KCCA) approximator. In a second step, the system matrices A, B, C and D and the non-linearities f and g will be obtained from the solution of an LS-SVM regression problem, similarly to what was done in the extension of the N4SID identification algorithm towards Hammerstein systems in Chapter 8.

A clear advantage of the proposed technique in this chapter is that it does not rely on restrictive assumptions on the inputs such as white noise or periodicity, that it is non-iterative in nature, and that it can conveniently be applied to MIMO systems. This in contrast to existing algorithms for the identification of Hammerstein-Wiener systems described in Section 6.4. Furthermore, other than the invertibility of g and a certain degree of smoothness, no specific restrictions are imposed on the non-linear maps f and g.

The outline of this chapter is as follows: In Section 9.2 the basic ingredients of the subspace intersection algorithm for linear systems are reviewed briefly. Section 9.3 extends the linear intersection algorithm towards a non-linear setting using a variation on the theme of LS-SVMs and kernel CCA. Section 9.4, finally, presents some illustrative examples.

9.2 Brief review of the subspace intersection algorithm

The subspace intersection algorithm was originally proposed in [38, 107] and is largely based on the idea that the state of a linear or non-linear model can be considered as the minimal intersection between past and future measurement data [94].

Following the notations and definitions of Chapter 3, and assuming a finite set of training data $\{(u_t, y_y)\}_{t=0}^{N-1}$, the main reasoning behind the subspace intersection algorithm for linear systems follows from the fact that under the assumptions that:

1. the input u_t is persistently exciting of order $2i$, i.e. the input block Hankel matrix $U_{0|2i-1}$ is of full rank,
2. the intersection of the row space of U_f (the future inputs) and the row space of X_p (the past states) is empty,

the following relation holds:

$$\text{Row}(X_f) = \text{Row}(W_p) \cap \text{Row}(W_f).$$

Hence, the order of the system and a realization of the state can be obtained from the intersection of past and future. Mathematically, this step is typically performed using a CCA algorithm, and retaining the canonical variates corresponding to canonical correlations equal to 1. Once the state is known, extraction of A, B, C and D is straightforward.

Without going into further theoretical details of the subspace intersection algorithm (interested readers are referred to [38, 107]), we summarize here a practical implementation that will be used towards the Hammerstein-Wiener model extension:

1. Perform Canonical Correlation analysis on W_p and W_f:

$$\begin{array}{rcl} W_p W_f^T V_f & = & W_p W_p^T V_p \Lambda, \\ W_f W_p^T V_p & = & W_f W_f^T V_f \Lambda, \end{array} \tag{9.2}$$

   with Λ a diagonal matrix containing the canonical correlations.

2. Determine the order n from the number of canonical correlations equal to 1. Retain X_f as the n corresponding canonical variates in W_p.

$$X_f = V_p(:, 1:n)^T W_p.$$

3. Extract A, B, C and D from:

$$\begin{bmatrix} X_f(:, 2:j) \\ Y_{i|i}(:, 1:j-1) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_f(:, 1:j-1) \\ U_{i|i}(:, 1:j-1) \end{bmatrix}. \tag{9.3}$$

The algorithm so obtained is mostly used for identification of purely deterministic systems, and therefore generally referred to as a deterministic subspace identification algorithm. Nevertheless, it was proven in [107] that when both the inputs and outputs are corrupted by additive spatially and temporary white noise sequences of equal covariance, a consistent estimate $\widehat{X}_f$ for the state sequence X_f is still obtained when using the algorithm described above. When this assumption is violated, it is possible to alter the algorithm by introducing weights based on the knowledge of the noise correlation [109]. For instructive purposes, we will however only consider the deterministic case in this chapter.

Note that in the third step of the subspace intersection algorithm as presented above, the same state sequence, up to a shift in time, is used at the left and right hand side of (9.3). This in contrast to most subspace identification algorithms described in Chapter 3 which feature two state sequences $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$ obtained from two different projection steps with shifted block Hankel matrices. However, in the latter case, a trick involving the observability matrix (see Subsection 3.5.7) was needed to ensure that both states were estimated in the same basis. As the observability matrix is not immediately accessible in the intersection algorithm as introduced above, such an approach can not be followed and we are forced to use the same state sequence at both sides of (9.3). As mentioned earlier, the drawback of this strategy is that the subspace intersection algorithm is mostly limited to deterministic systems.

9.3 Hammerstein-Wiener subspace intersection

9.3.1 Introducing the static non-linearities

A classical state space system is transformed into a Hammerstein-Wiener system by introducing two static non-linearities $f : \mathbb{R}^m \to \mathbb{R}^m$ and $g : \mathbb{R}^l \to \mathbb{R}^l$. With this definition for the non-linearities, and assuming that $g : \mathbb{R}^l \to \mathbb{R}^l$ is such that g^{-1} exists for all possible outputs of the system, the following state-space model will be studied in this chapter:

$$\begin{cases} x_{t+1} = Ax_t + Bf(u_t), \\ g^{-1}(y_t) = Cx_t + Df(u_t). \end{cases}$$

As mentioned in 9.2, a CCA algorithm could be used to extract the state x if $f(u)$ and $g^{-1}(y)$ were known. The state is then obtained from

$$\begin{aligned} \mathcal{F}(W_p)\mathcal{F}(W_f)^T V_f &= \mathcal{F}(W_p)\mathcal{F}(W_p)^T V_p \Lambda, \\ \mathcal{F}(W_f)\mathcal{F}(W_p)^T V_p &= \mathcal{F}(W_f)\mathcal{F}(W_f)^T V_f \Lambda, \end{aligned}$$

where $\mathcal{F}(W_p)$ is defined as follows

$$\mathcal{F}(W_p) \triangleq \begin{bmatrix} f(u_0) & f(u_1) & \dots & f(u_{j-1}) \\ \vdots & \vdots & & \vdots \\ f(u_{i-1}) & f(u_i) & & f(u_{i+j-2}) \\ g^{-1}(y_0) & g^{-1}(y_1) & \dots & g^{-1}(y_{j-1}) \\ \vdots & \vdots & & \vdots \\ g^{-1}(y_{i-1}) & g^{-1}(y_i) & & g^{-1}(y_{i+j-2}) \end{bmatrix},$$

with an equivalent definition for $\mathcal{F}(W_f)$. However, because $f(u)$ and $g^{-1}(y)$ are unknown, another approach is required to extract the state. A well-suited technique to fulfill this task is kernel CCA, a non-linear extension of CCA, which will be treated in Subsection 9.3.2. Once the state is known, $f(u)$ and $g^{-1}(y)$ will be estimated using a second step described in Subsections 9.3.4 and 9.3.5.

9.3.2 Introducing the kernel

To extract a state of a non-linear dynamical system, a non-linear extension of CCA is employed, known as kernel CCA or KCCA [10, 93]. In kernel methods [135] the available data are mapped into a high-dimensional feature space of dimension n_H, where classical CCA is applied. As was also seen in the former chapters, the non-linearity is thereby condensed in the transformation, which is represented by feature maps $\varphi^u : \mathbb{R}^m \to \mathbb{R}^{n_H}$ and $\varphi^y : \mathbb{R}^l \to \mathbb{R}^{n_H}$. Using the mapped past data points $\varphi^u(u_t)$ and $\varphi^y(y_t)$, $\forall t$, one constructs a feature matrix

$$\Phi_p \triangleq \Phi(W_p) \triangleq \begin{bmatrix} \varphi^u(u_0) & \varphi^u(u_1) & \dots & \varphi^u(u_{j-1}) \\ \vdots & \vdots & & \vdots \\ \varphi^u(u_{i-1}) & \varphi^u(u_i) & & \varphi^u(u_{i+j-2}) \\ \varphi^y(y_0) & \varphi^y(y_1) & \dots & \varphi^y(y_{j-1}) \\ \vdots & \vdots & & \vdots \\ \varphi^y(y_{i-1}) & \varphi^y(y_i) & & \varphi^y(y_{i+j-2}) \end{bmatrix} \in \mathbb{R}^{2i(m+l)n_H \times j}, \quad (9.4)$$

with a similar definition for $\Phi_f \triangleq \Phi(W_f)$.

The kernels associated to φ^u and φ^y will be denoted by K^u and K^y respectively:

$$\begin{aligned} \varphi^u(u_s)^T\varphi^u(u_t) &= K^u(u_s,u_t), \\ \varphi^y(y_s)^T\varphi^y(y_t) &= K^y(y_s,y_t). \end{aligned}$$

For further reference, we also define $K_p \triangleq \Phi_p^T\Phi_p$ and $K_f \triangleq \Phi_f^T\Phi_f$.

9.3.3 From CCA to KCCA: the state estimate

For reasons of clarity of presentation we adopt here a formal introduction into the KCCA algorithm as it was initially presented in [10,93]. For a more rigorous description of the main concepts behind KCCA, the reader is kindly referred to the latter references.

By mapping the elements of W_p and W_f the CCA problem in feature space becomes:

$$\begin{aligned} \Phi_p\Phi_f^T\ V_f &= \Phi_p\Phi_p^T\ V_p\Lambda, \\ \Phi_f\Phi_p^T\ V_p &= \Phi_f\Phi_f^T\ V_f\Lambda, \end{aligned} \tag{9.5}$$

Remark that the coefficient matrices V_p, V_f are elements of $\mathbb{R}^{2i(m+l)n_H\times 2i(m+l)n_H}$ where n_H can be potentially infinite-dimensional, which is not practical. If however these matrices are restricted to the subspace spanned by the mapped data by defining:

$$\mathcal{V}_p = \Phi_p V_p,\ \mathcal{V}_f = \Phi_f V_f, \tag{9.6}$$

and the first and second equation of (9.5) are left multiplied by Φ_p^T and Φ_f^T, respectively, we obtain:

$$\begin{aligned} K_pK_f\ \mathcal{V}_f &= K_pK_p\ \mathcal{V}_p\Lambda, \\ K_fK_p\ \mathcal{V}_p &= K_fK_f\ \mathcal{V}_f\Lambda. \end{aligned} \tag{9.7}$$

Assuming that K_p and K_f are invertible, which can be shown to be automatically satisfied if complex non-linear kernels such as the RBF are involved, this can further be reduced to

$$\begin{aligned} K_f\ \mathcal{V}_f &= K_p\ \mathcal{V}_p\ \Lambda \\ K_p\ \mathcal{V}_p &= K_f\ \mathcal{V}_f\ \Lambda, \end{aligned} \tag{9.8}$$

which is the classical form for of the KCCA algorithm as presented in [10,93]. A disadvantage of this KCCA version is the fact that the used kernel derivations do not contain regularization leaving the possibility of a severe over-fitting of the non-linearities involved.

The KCCA version proposed in [135] is formulated using a least squares support vector machine approach [150] with primal-dual optimization problems and an additional centering of the data-points in feature space. Regularization is thereby incorporated within the primal formulation in a well-established manner leading to numerically better conditioned solutions. Without going into the details of this algorithm (the interested reader is kindly referred to [135]), we state here the final generalized eigenvalue problem:

$$\begin{aligned} K_f^c\,\mathcal{V}_f &= \left(K_p^c + \frac{1}{\gamma} I_j\right)\,\mathcal{V}_p\,\Lambda, \\ K_p^c\,\mathcal{V}_p &= \left(K_f^c + \frac{1}{\gamma} I_j\right)\,\mathcal{V}_f\,\Lambda, \end{aligned}$$

where

$$\begin{aligned} K_p^c &= (\Phi_p - 1_j^T \otimes \mu_p)^T(\Phi_p - 1_j^T \otimes \mu_p), \\ K_f^c &= (\Phi_f - 1_j^T \otimes \mu_f)^T(\Phi_f - 1_j^T \otimes \mu_f), \end{aligned}$$

are the so–called centered kernels and $\mu_p = (1/j)\sum_{s=1}^{j}\Phi_p(:,s)$ and $\mu_f = (1/j)\sum_{s=1}^{j}\Phi_f(:,s)$ are the mean centers of the mapped past and future. The tuning parameter γ controls the amount of regularization. A comparison with the derived result without centering yields that $K_f^c = M_c K_f M_c$ with $M_c = (I_j - (1/j)11_j^T)$ [128].

Thus by solving a generalized eigenvalue problem in the dual space, one can find the canonical correlations and the non-linear canonical variates, gathered respectively in the KCCA estimates $\widehat{\Lambda}$, $\widehat{\mathcal{V}}_p$ and $\widehat{\mathcal{V}}_f$. From the number of canonical correlations equal to one, we determine the order n. The estimated state is obtained as the n corresponding linear combinations of the centered variates in Φ_p, which comes down to:

$$\widehat{X}_f = \widehat{\mathcal{V}}_p(1:n,:)^T K_p^c. \tag{9.9}$$

9.3.4 Estimation of A, B and the non-linear function f

After the estimation of the state, the system matrices A and B and the non-linear function f are estimated in a second step as follows:

$$(\widehat{A}, \widehat{B}, \hat{f}) = \arg\min_{A,B,f} \left\| \widehat{X}_{i+1} - \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \widehat{X}_i \\ \mathcal{U}_f \end{bmatrix} \right\|_F^2, \tag{9.10}$$

with

$$\mathcal{U}_f \triangleq \begin{bmatrix} f(u_i) & f(u_{i+1}) & \dots & f(u_{i+j-2}) \end{bmatrix},$$

and where we have conveniently redefined $\widehat{X}_i$ and $\widehat{X}_{i+1}$ as follows

$$\widehat{X}_i \triangleq \widehat{X}_f(:,1:j-1), \tag{9.11}$$

$$\widehat{X}_{i+1} \triangleq \widehat{X}_f(:,2:j), \tag{9.12}$$

to ensure some continuity in notation with respect to Chapter 8 on Hammerstein N4SID identification. Again this least-squares problem will be written as a classical LS-SVM regression problem. The first step towards such a regression problem is to make the replacement

$$Bf = \begin{bmatrix} w_{f,1}^T \\ w_{f,2}^T \\ \vdots \\ w_{f,n}^T \end{bmatrix} \varphi^u, \tag{9.13}$$

with φ^u the feature-map introduced in 9.3. With this replacement, equation (9.10) is rewritten as

$$(\widehat{A}, \widehat{B}, \hat{w}) = \arg \min_{A,B,f} \left\| \widehat{X}_{i+1} - A\widehat{X}_i - \begin{bmatrix} w_{f,1}^T \\ w_{f,2}^T \\ \vdots \\ w_{f,n}^T \end{bmatrix} \mathcal{U}_\varphi \right\|_F^2 ,$$

with

$$\mathcal{U}_\varphi \triangleq \begin{bmatrix} \varphi^u(u_i) & \varphi^u(u_{i+1}) & \ldots & \varphi^u(u_{i+j-2}) \end{bmatrix}.$$

The resulting LS-SVM primal problem is as follows:

$$\begin{aligned} \min_{w,E,A} \mathcal{J}(w, E) &= \tfrac{1}{2} \textstyle\sum_{s=1}^{n} w_{f,s}^T w_{f,s} + \tfrac{\gamma_u}{2} \sum_{s=1}^{n} \sum_{t=1}^{j-1} E(s,t)^2, \\ \text{subject to} \quad & \widehat{X}_{i+1}(s,t) = A\widehat{X}_i(:,t) + w_{f,s}^T \mathcal{U}_\varphi(:,t) + E(s,t), \\ & \forall s = 1, \ldots, n, \; t = 1, \ldots, j-1. \end{aligned} \tag{9.14}$$

Unlike in the chapters on Hammerstein ARX and Hammerstein N4SID identification, no extra constraints are added to the primal problem to ensure that the estimated non-linearities are centered around zero. These constraints were necessary in the ARX and N4SID case as sums of non-linear functions were estimated in a least-squares sense with the possibility of random constants being added to each of the non-linearities as long as the sum of these constants was zero. Hence, by retaining only those non-linearities associated with the past data in the oblique projection in the N4SID case for instance, unwanted offsets which do not necessarily sum to zero (as the future constants are removed) were inserted into the obtained models. In the proposed intersection algorithm, this is no longer a concern as after the KCCA step no specific set of non-linearities is removed when calculating the state estimate in (9.9). Hence, we can safely continue without the centering constraint.

Lemma 9.1. Primal-dual characterization: *Given the least squares problem (9.10) and related primal problem (9.14), LS-SVM estimates for B and the transformed inputs $\mathcal{U}_f$ can be obtained from a rank m approximation of $\mathcal{A}^T\mathcal{K}^u$. $\mathcal{A}$ and A are obtained from the following set of linear equations:*

$$\left[\begin{array}{c|c} 0 & \widehat{X}_i \\ \hline \widehat{X}_i^T & \mathcal{K}^u + \gamma_u^{-1} I_{j-1} \end{array}\right] \left[\begin{array}{c} A^T \\ \hline \mathcal{A} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \widehat{X}_{i+1}^T \end{array}\right], \tag{9.15}$$

with

$$\begin{aligned}\mathcal{A} &= \begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} & \dots & \alpha_{n,1} \\ \alpha_{1,2} & \alpha_{2,2} & \dots & \alpha_{n,2} \\ \vdots & \vdots & & \vdots \\ \alpha_{1,j-1} & \alpha_{2,j-1} & \dots & \alpha_{n,j-1} \end{bmatrix}, \\ w_{f,s} &= \textstyle\sum_{t=1}^{j-1} \alpha_{s,t} \varphi^u(u_{i+t-1}),\ s = 1, \dots, n, \\ \mathcal{K}^u(p,q) &= K^u(u_{i+p-1}, u_{i+q-1}),\ p, q = 1, \dots, j-1. \end{aligned}$$

Proof. This directly follows from the Lagrangian:

$$\begin{aligned}\mathcal{L}(w, A, E; \alpha) &= \mathcal{J}(w, E) \\ &- \sum_{s=1}^{n} \sum_{t=1}^{j-1} \alpha_{s,t} \left(\widehat{X}_{i+1}(s,t) - A\widehat{X}_i(:,t) - w_{f,s}^T \mathcal{U}_\varphi(:,t) - E(s,t) \right), \end{aligned}$$

by taking the conditions for optimality: $\frac{\partial \mathcal{L}}{\partial w_{f,s}} = 0$, $\frac{\partial \mathcal{L}}{\partial A} = 0$, $\frac{\partial \mathcal{L}}{\partial E(s,t)} = 0$, $\frac{\partial \mathcal{L}}{\partial \alpha_{s,t}} = 0$ and by observing that:

$$B\mathcal{U}_f = \begin{bmatrix} w_{f,1}^T \\ w_{f,2}^T \\ \vdots \\ w_{f,n}^T \end{bmatrix} \mathcal{U}_\varphi = \mathcal{A}^T \mathcal{K}^u. \tag{9.16}$$

□

Note that the estimates obtained in Lemma 9.1 will in general not be uniquely defined, especially if $n \leq m$. This is an intrinsic property of Hammerstein-Wiener models and the choice of the actual representation is left to the user. From $\mathcal{U}_f$ and the inputs $u_t,\ t = i, \dots, i+j-2$, obtaining an estimate for f is a straightforward matter.

9.3.5 Estimation of C, D and the non-linear function g

Once an estimate $\widehat{\mathcal{U}_f}$ for $\mathcal{U}_f$ has been found, estimates for the system matrices C and D and the non-linearity g^{-1} are obtained from:

$$(\widehat{C}, \widehat{D}, \hat{g}^{-1}) = \underset{C,D,g^{-1}}{\arg\min} \left\| \mathcal{Y}_g - \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} \widehat{X}_i \\ \widehat{\mathcal{U}_f} \end{bmatrix} \right\|_F^2, \tag{9.17}$$

with

$$\mathcal{Y}_g \triangleq \begin{bmatrix} g^{-1}(y_i) & g^{-1}(y_{i+1}) & \dots & g^{-1}(y_{i+j-2}) \end{bmatrix}.$$

Remark that C, D and g^{-1} in (9.17) are only defined up to a constant scaling factor. To avoid conditioning problems, we therefore fix $C(:,1) = 1_l$, which is a feasible assumption for SISO systems given that the first component of the state is generally observable in subspace models [145,147]. For MIMO systems, it is possible that certain outputs are unexcited by the first state component, in which case a more complicated constraint (such as $\sum_{k=1}^{n} C(:,k) = 1_l$) might be more appropriate. Although such constraints can easily be incorporated into the LS-SVM framework [19,135], we will further assume that $C(:,1) = 1_l$ for simplicity. Derivations involving other constraints can easily be derived along the lines of the calculations below. With $C(:,1) = 1_l$, the resulting LS-SVM problem is as follows:

$$\begin{aligned} \min_{w,E,C,D} \mathcal{J}(w,E) &= \tfrac{1}{2}\textstyle\sum_{s=1}^{l} w_{g,s}^T w_{g,s} + \tfrac{\gamma_y}{2}\sum_{s=1}^{n}\sum_{t=1}^{j-1} E(s,t)^2, \\ \text{subject to} \qquad & \widehat{X}_i(1,t) = w_{g,s}^T \mathcal{Y}_\varphi(:,t) - C(s,2:n)\widehat{X}_i(2:n,t) \\ & \quad -D(s,:)\mathcal{U}_f(:,t) - E(s,t), \\ & \quad \forall s = 1,\ldots,l,\ t = 1,\ldots,j-1. \end{aligned} \tag{9.18}$$

with

$$\mathcal{Y}_\varphi = \begin{bmatrix} \varphi^y(y_i) & \varphi^y(y_{i+1}) & \ldots & \varphi^y(y_{i+j-2}) \end{bmatrix},$$

whereby φ^y is as in Section 9.3.

Lemma 9.2. Primal-dual characterization: *Given the least squares problem (9.17), and the related primal problem (9.18), LS-SVM estimates for the transformed outputs $\mathcal{Y}_g$ are obtained as $\mathcal{A}^T\mathcal{K}^y$. $\mathcal{A}$, C and D are obtained from the following set of linear equations:*

$$\left[\begin{array}{c|c|c} 0 & 0 & \overline{X}_i \\ \hline 0 & 0 & \widehat{\mathcal{U}}_f \\ \hline \overline{X}_i^T & \widehat{\mathcal{U}}_f^T & \mathcal{K}^y + \gamma_y^{-1} I_{j-1} \end{array}\right] \left[\begin{array}{c} -C(:,2:n)^T \\ \hline -D^T \\ \hline \mathcal{A} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline 1_l^T \otimes \widehat{X}_i(1,:)^T \end{array}\right], \tag{9.19}$$

whereby

$$\begin{aligned} \mathcal{A} &= \begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} & \ldots & \alpha_{l,1} \\ \alpha_{1,2} & \alpha_{2,2} & \ldots & \alpha_{l,2} \\ \vdots & \vdots & & \vdots \\ \alpha_{1,j-1} & \alpha_{2,j-1} & \ldots & \alpha_{l,j-1} \end{bmatrix}, \\ \overline{X}_i &= \widehat{X}_i(2:n,:), \\ w_{g,s} &= \textstyle\sum_{t=1}^{j-1} \alpha_{s,t}\varphi^y(y_{i+t-1}),\ s = 1,\ldots,l, \\ \mathcal{K}^y(p,q) &= K^y(y_{i+p-1}, y_{i+q-1}),\ p,q = 1,\ldots,j-1. \end{aligned}$$

Proof. This directly follows from the Lagrangian:

$$\begin{aligned} \mathcal{L}(w,E,C,D;\alpha) &= \mathcal{J}(w,E) - \sum_{s=1}^{l}\sum_{t=1}^{j-1} \alpha_{s,t}\Big(w_{g,s}^T \mathcal{Y}_\varphi(:,t) - \widehat{X}_i(1,t) \\ &- C(s,2:n)\widehat{X}_i(2:n,t) - D(s,:)\mathcal{U}_f(:,t) - E(s,t)\Big). \end{aligned}$$

by taking the conditions for optimality: $\frac{\partial \mathcal{L}}{\partial w_{g,s}} = 0$, $\frac{\partial \mathcal{L}}{\partial C} = 0$, $\frac{\partial \mathcal{L}}{\partial D} = 0$, $\frac{\partial \mathcal{L}}{\partial E(s,t)} = 0$, $\frac{\partial \mathcal{L}}{\partial \alpha_{s,t}} = 0$ and by observing that:

$$\mathcal{Y}_g = \begin{bmatrix} w_{g,1}^T \\ w_{g,2}^T \\ \vdots \\ w_{g,n}^T \end{bmatrix} \mathcal{Y}_\varphi = \mathcal{A}^T \mathcal{K}^y. \tag{9.20}$$

$\square$

Again, from $\mathcal{Y}_g$ in Lemma 9.2 and the outputs y_t, $t = i, \ldots, i+j-2$, obtaining an estimate for g is a trivial matter.

9.3.6 Practical implementation

Following the discussion in the former sections, the final algorithm for the estimation of Hammerstein-Wiener systems can be summarized as follows:

1. Find estimates for the state sequences $\widehat{X}_i$ and $\widehat{X}_{i+1}$ from (9.9), (9.11) and (9.12).
2. Obtain estimates for A, B and f following the procedure outlined in Subsection 9.3.4.
3. Obtain estimates for C, D and g following the procedure outlined in Subsection 9.3.5.

Some practical issues remain, regarding the tuning of the hyper-parameters, the need for persistently exciting inputs, and the behavior of the presented algorithm under the presence of process and/or measurement-noise.

Tuning of the hyper-parameters

Many tunable parameters, the so-called hyper-parameters, are present in the proposed algorithm such as the system order n, the number of block rows, i, in the block Hankel matrices, the regularization parameters γ, γ_u and γ_y in the KCCA- and LS-SVM estimation steps, and other potential kernel parameters such as the bandwidths σ_u and/or σ_y when RBF-kernels are used. In principle, these parameters could be tuned by validating the performance of the obtained Hammerstein-Wiener model on an independent validation dataset. However, as this would constitute a highly non-convex high-dimensional search, the resulting identification algorithm would computationally be too extensive for most modern computers.

Fortunately, it can be shown that the tuning problem can be split up in several sub-problems. From the resulting $\mathcal{V}_p$ and $\mathcal{V}_f$ from the KCCA step in Section 9.3, for instance, a validation state based on past data can be calculated as follows:

$$\widehat{X}_f^{\text{val}} = \widehat{\mathcal{V}}_p(1:n,:)^T K_p^{\text{val}},$$

with

$$K_p^{\mathrm{val}} = (\Phi_p - 1_j^T \otimes \mu_p)^T (\Phi_p^{\mathrm{val}} - 1_j^T \otimes \mu_p),$$

and the superscript $\cdot^{\mathrm{val}}$ denoting that inputs and outputs from the validation dataset, rather than the training-dataset are considered. If the KCCA-step is well tuned, the following relation should hold:

$$\mathrm{Row}\left(\widehat{X}_f^{\mathrm{val}}\right) = \mathrm{Row}\left(\widehat{\mathcal{V}}_f(1:n,:)^T K_f^{\mathrm{val}}\right), \tag{9.21}$$

with

$$K_f^{\mathrm{val}} = (\Phi_f - 1_j^T \otimes \mu_f)^T (\Phi_f^{\mathrm{val}} - 1_j^T \otimes \mu_f),$$

the equivalent validation state based on future data. The extent to which relation (9.21) holds can easily be checked by calculating the largest canonical angle between both row-spaces. Hence, the hyper-parameters necessary for the KCCA step can be obtained without having to estimate the full Hammerstein-Wiener model. A similar reasoning can be used for the other steps in the proposed algorithm such as the estimation of A, B and f in Subsection 9.3.4.

Persistency of excitation

Given the fact that regularization is inherently present in the proposed identification algorithm, in line with the results obtained in Chapter 8 lack of persistency of excitation will not lead to any numerical problems. However, to ensure that all aspects of the linear system are properly identified, persistency of excitation of $f(u)$ of at least order $2im$ is necessary. As was noted in Chapter 8, for some non-linear functions f, persistency of excitation of $f(u)$ can be guaranteed if u is persistently exciting (see [156] for a discussion on this issue).

Behavior in noisy circumstances

As noted in the introduction to the linear subspace intersection algorithm in Section 9.2, the intersection algorithm is mostly limited to noiseless systems. Nevertheless, the intersection algorithm is known to perform adequately, even if small amounts of noise are present on the inputs and/or outputs. To further illustrate this point, a modest amount of output-noise will be added to some of the simulations in Section 9.4.

Re-estimation of the linear model

Following the discussion on the behavior of the intersection algorithm in noisy circumstances, it is in principle possible to use the estimates for f and g^{-1} in the proposed intersection algorithm to generate the inputs $f(u)$ and outputs $g^{-1}(y)$ to the linear system. Based on these data-sequences a more robust subspace identification such as the PO-MOESP algorithm [155] can be used in a second step to replace the original model obtained using the intersection algorithm. It will be shown in the examples in Section 9.4 that such a second step can indeed improve upon the accuracy of the obtained linear model, certainly in noisy conditions.

9.4 Illustrative examples

9.4.1 A SISO system

Consider the following artificial linear system which belongs to the class of Hammerstein-Wiener models:

$$y = g\left(\frac{B(z)}{A(z)}f(u)\right), \tag{9.22}$$

with A and B polynomials in the forward shift operator z where $B(z) = z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3$ and $A(z) = (z - 0.98e^{\pm i})(z - 0.98e^{\pm 1.6i})(z - 0.97e^{\pm 0.4i})$, The input- and output-non-linearities are given by $f : \mathbb{R} \rightarrow \mathbb{R} : f(u) = \mathrm{sinc}(u)$ and

$$g : \mathbb{R} \rightarrow \mathbb{R} : g(y) = \begin{cases} y/12, & y \leq 0, \\ \mathrm{tanh(y/4)}, & \mathrm{y} > 0. \end{cases} \tag{9.23}$$

Two datasets were generated from this system with the inputs $u_t \sim \mathcal{N}(0, 2)$ white Gaussian noise sequences for $t = 0, \ldots, N-1$ with $N = 500$. Although the intersection algorithm is in principle only designed for deterministic systems, 5% of zero mean white Gaussian noise was added to the outputs in both datasets. The first dataset so obtained was used to train the model, the second one was used to tune the model. Only the less critical number of block-rows in the Hankel matrices was fixed beforehand at 10, a common choice in subspace algorithms.

For K^u and K^y, RBF kernels were chosen with $\sigma_u = 1$ and $\sigma_y = 0.5$ respectively. These kernel bandwidth parameters, together with the hyper-parameter $\gamma = 1$ and the obtained V_p and V_f from the KCCA estimation step outlined in Subsection 9.3.3, were validated on the validation dataset according to the procedure explained in Subsection 9.3.6. In a second step, estimates for A, B and the non-linear function f_u were obtained using the procedure presented in Subsection 9.3.4 where $\gamma_u = 1$ was chosen after validation of the relation (9.15) on the validation dataset. Finally C, D and the non-linear function f_y were obtained following the procedure presented in 9.3.5 with $\gamma_y = 315$, again chosen by validation on the validation dataset.

The obtained non-linear functions $\hat{f}$ and $\hat{g}$ evaluated on the validation inputs and outputs, are compared with the true functions f and g in Figure 9.1. As can be seen in the figure, the obtained estimates are quite reliable. The obtained linear system is compared with the true system in Figure 9.2. Note that the first two resonances are nicely caught by the model, whereas the less energetic third resonance is not found. In itself not a bad result considering that it has already been shown in Subsection 5.2.2 that the used system is a difficult one to estimate with classical techniques such as ARX, and that the intersection algorithm is in essence a deterministic algorithm. Nevertheless, as mentioned in Subsection 9.3.6 a further improvement is possible by rerunning a linear subspace algorithm such as the PO-MOESP [155] on the estimated inputs $\hat{f}(u)$ and outputs $\hat{g}^{-1}(y)$ of the linear system. That such a second step indeed improves the accuracy of the estimate of the linear model is shown in Figure 9.3.

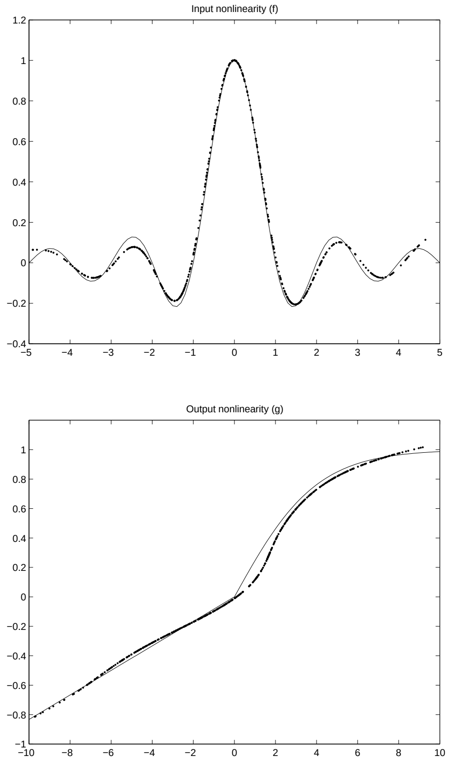


Figure 9.1: Estimated input non-linearity f and output non-linearity g evaluated on the validation inputs and outputs (dots) compared with the true non-linearities (solid line) for the SISO example described in Section 9.4.

9.4.2 A MIMO system

To illustrate the freedom one gets by plug-in of an appropriate kernel, in a second example, the proposed identification method was applied to the MIMO Hammerstein system presented in Subsection 7.7.2. We recall that the system is given as:

$$y = \begin{bmatrix} \frac{b_1(z)}{a_1(z)} & \frac{b_2(z)}{a_1(z)} \\ \frac{b_1(z)}{a_2(z)} & \frac{b_2(z)}{a_2(z)} \end{bmatrix} f(u) + \begin{bmatrix} \frac{1}{a_1(z)} \\ \frac{1}{a_2(z)} \end{bmatrix} e \tag{9.24}$$

with

$$\begin{aligned} a_1(z) &= (z - 0.98e^{\pm i})(z - 0.98e^{\pm 1.6i})(z - 0.97e^{\pm 0.4i}), \\ a_2(z) &= (z - 0.97e^{\pm 0.7i})(z - 0.98e^{\pm 1.4i})(z - 0.97e^{\pm 2.3i}), \\ b_1(z) &= z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3, \end{aligned}$$

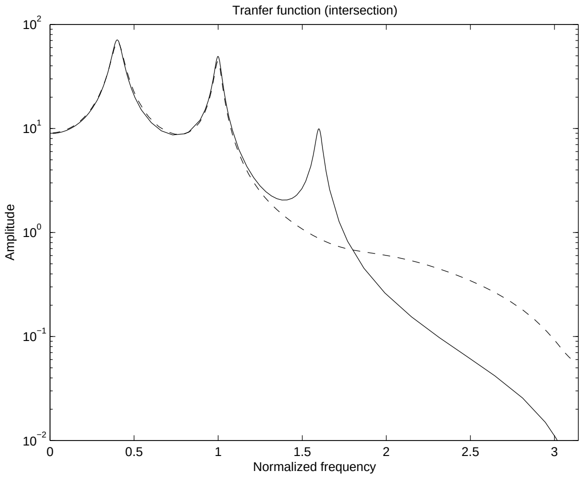


Figure 9.2: Estimated transfer functions (dashed) for the SISO example described in Section 9.4 using the intersection-algorithm. The true transfer function is displayed in solid.

$$b_2(z) \quad = \quad z^6 + 0.9z^5 + 0.7z^4 + 0.2z^3,$$

and

$$f(u) = \begin{bmatrix} -\arctan(u(1))\arctan(u(2)) \\ \arctan(u(1)) - \arctan(u(2)) \end{bmatrix}.$$

A two-component zero mean white Gaussian input sequence u with length 500 and standard deviation 1 was generated and fed into the system (9.24). Based on u and the obtained output y, estimates for the linear system and f are obtained using the Hammerstein-Wiener identification algorithm proposed in this chapter, whereby an RBF kernel was chosen for K^u and a linear kernel for K^y. The latter is necessary to effectively limit the Hammerstein-Wiener algorithm to Hammerstein systems.

As in the SISO example, the number of block-rows in the Hankel matrices was chosen equal to 10. The hyper-parameters were again obtained by evaluation on a validation set and chosen as $\sigma_u = 1$, $\gamma = 0.1$ and $\gamma_u = \gamma_y = 1$. The order was easily found to be 12 from an inspection of the canonical correlations in the kernel CCA step (see Figure 9.4). The results from a simulation on an independent test-set using the obtained model is shown in Figure 9.5 for the first component of the output and Figure 9.6 for the second component of the output. Also available in the figure are the results of a classical linear PO-MOESP subspace estimator (same order) which are clearly inferior to those obtained using the Hammerstein-Wiener approach.

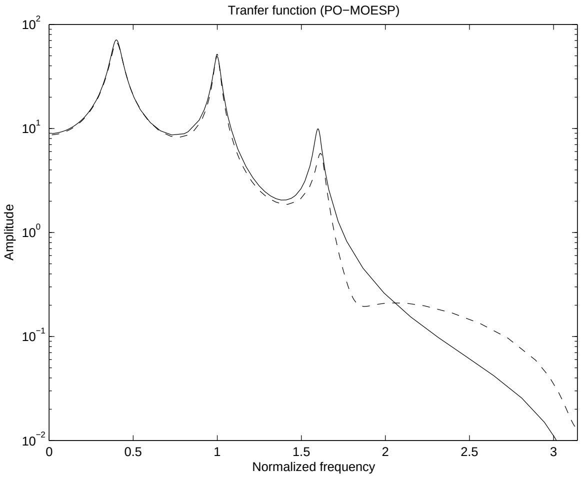


Figure 9.3: Estimated transfer functions (dashed) for the SISO example described in Section 9.4 using a PO-MOESP after estimation of the functions f and g. The true transfer function is displayed in solid.

9.5 Conclusions

In this chapter, a method for the identification of Hammerstein-Wiener systems was presented based on the method of kernel canonical correlation analysis and Least Squares Support Vector Machines. The proposed algorithm is applicable to SISO and MIMO systems and does not impose restrictive assumptions on the input sequence, in contrast to most existing Hammerstein-Wiener approaches. Furthermore, the algorithm was seen to work well on a set of examples.

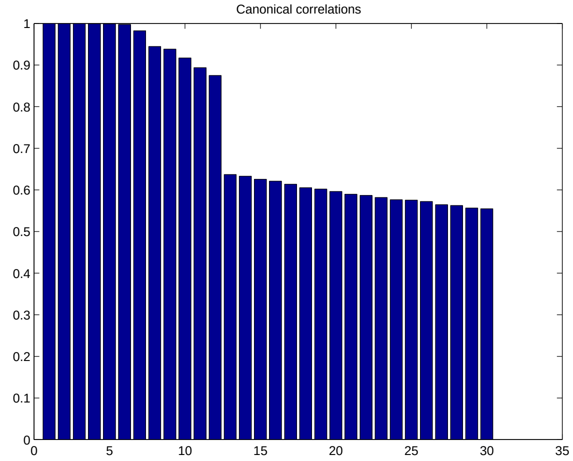


Figure 9.4: Canonical correlations obtained using kernel CCA on a twelfth order MIMO Hammerstein system described in 9.4.2. The order of the system is clearly seen to be equal to 12.

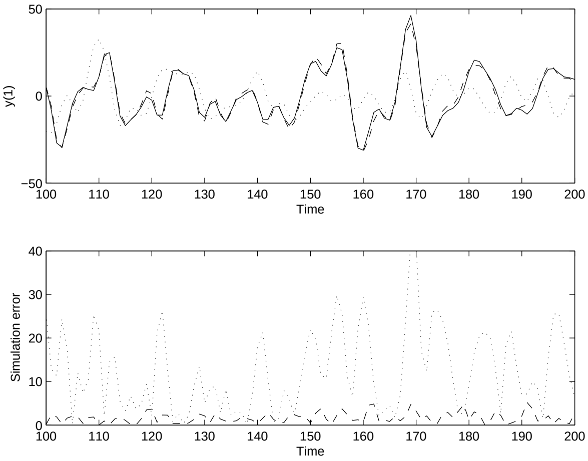


Figure 9.5: Simulation on an independent test-set of the first output $y(1)$ of a twelfth order MIMO Hammerstein model described in 9.4.2 using an LS-SVM Hammerstein-Wiener estimator (dashed line) and a linear PO-MOESP subspace estimator (dotted line). The true output is depicted with a solid line. All simulations are initialized with $x_0 = 0$. The error between the estimated and the true output is shown in the lower figure.

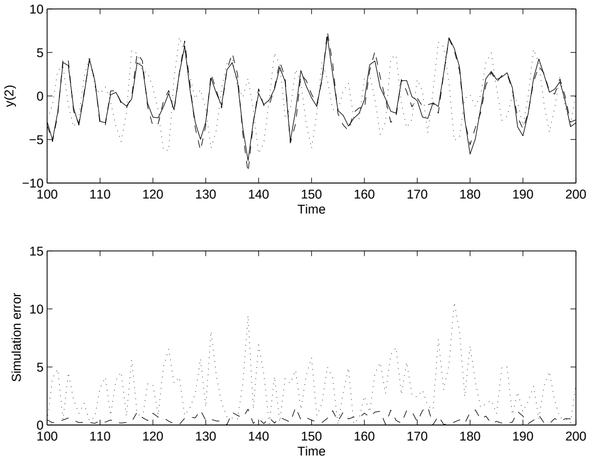


Figure 9.6: Simulation on an independent test-set of the second output $y(2)$ of a twelfth order MIMO Hammerstein model described in 9.4.2 using an LS-SVM Hammerstein-Wiener estimator (dashed line) and a linear PO-MOESP subspace estimator (dotted line). The true output is depicted with a solid line. All simulations are initialized with $x_0 = 0$. The error between the estimated and the true output is shown in the lower figure.

Chapter 10

Conclusions, future research, and open problems

General conclusions

In this thesis, we have studied subspace identification for linear, Hammerstein and Hammerstein-Wiener systems. For linear systems it was seen that despite the widely perceived robustness of subspace identification algorithms, under some specific experimental conditions, they may fail or yield unreliable results. Several solutions were proposed and tested in this thesis.

For Hammerstein and Hammerstein-Wiener systems, reliable subspace identification algorithms were obtained by combining ideas from LS-SVM function regression with the principal projections underlying subspace identification algorithms. More detailed conclusions are found below.

Conclusions for Part I

The first issue which was treated in Part I of this thesis is the so-called positive-realness problem. In Chapter 4, it was seen that positive-realness of the covariance model, obtained as an intermediate step in stochastic subspace identification, is essential if one wants to obtain statistical information regarding the noise acting on the system.

Several existing techniques to deal with a possible lack of positive realness were discussed, and it was seen that some of these techniques were limited to stable systems while others were applicable to stable as well as unstable systems. A new algorithm based on Tikhonov regularization was thereafter proposed. It was proven that using this new approach, positive-realness can be guaranteed provided the amount of regularization that is applied is chosen sufficiently high.

Furthermore, the newly proposed algorithm was seen to outperform existing approaches on a set of examples.

A second issue which was treated in Part I of this thesis is the possible ill-conditioning of combined stochastic-deterministic subspace identification algorithms, and especially the N4SID, under the presence of highly colored inputs. In Chapter 5, two reasons were given for this ill-conditioning. First of all, the oblique projection present in N4SID subspace identification algorithms was seen to be badly conditioned if the input coloring is high. Secondly, all subspace identification algorithms of the combined stochastic-deterministic kind suffer from possible weak correlations between the stochastic state and the system inputs, with grave consequences if highly colored inputs are involved.

It was seen that the orthogonal decomposition method presented in [26] behaves much better than classical approaches such as the N4SID under the presence of highly colored inputs. Two reasons were given for this behavior. First of all, the orthogonal decomposition method features an orthogonal projection to obtain the state instead of an oblique projection as is commonly found in N4SID algorithms. This alone led to a considerable improvement in the estimates for the system poles. Secondly, and especially important for the estimation of B and D, the orthogonal projection is based on a separate parameterization of the stochastic and the deterministic subsystem, preventing problems associated with weak correlations between the stochastic state and the system inputs.

In order to maintain the oblique projection, it was thereafter shown that when using weighted Tikhonov regularization, a considerable improvement in the accuracy of the system pole estimates was obtained. This, combined with a separate parameterization for the stochastic and the deterministic subsystem led to a regularized N4SID algorithm that outperforms the orthogonal projection on a set of examples and serves to highlight the possibilities of using regularization in subspace identification.

Conclusions for part II

In part II of this thesis, we have mainly focused on extending linear subspace identification techniques to Hammerstein and Hammerstein-Wiener systems. These extensions were obtained by combining the idea of over-parameterization with LS-SVM function regression.

A first conclusion that could be drawn from the discussion on existing over-parameterization techniques in Chapter 6 is the need for centering to avoid the appearance of random constants in the estimates for the different nonlinearities. Centering measures were therefore included in the derivations of the Hammerstein ARX identification algorithm in Chapter 7, the Hammerstein N4SID identification algorithm in Chapter 8 and the Hammerstein-Wiener subspace intersection algorithm in Chapter 9.

With the introduction of the MIMO ARX Hammerstein identification algorithm based on component-wise LS-SVM regression in Chapter 7, it was seen that a considerable improvement in accuracy can be obtained with respect to existing over-parameterization approaches. A reason for this increase in accuracy was found in the use of regularization, which is inherently present in LS-SVM algorithms. However, even if regularization is applied to classical over-parameterization approaches, the superior performance of the LS-SVM based algorithm remains. Furthermore, LS-SVM algorithms have the advantage that constraints such as the centering of the static non-linearities can conveniently be included in the primal-dual framework. These results offer a strong indication that the use of LS-SVM regression and the related primal-dual framework is a viable, and in many cases preferable, alternative to classical over-parameterization approaches.

In Chapter 8, and based on the results for ARX Hammerstein identification, a MIMO N4SID Hammerstein identification algorithm was developed. Again LS-SVM function regression is at the core of the obtained algorithm. The advantage of the presented N4SID algorithms over the in Chapter 7 presented ARX Hammerstein algorithm is the greater flexibility in the linear model structures that can be handled. While the ARX Hammerstein identification algorithm is limited to Hammerstein systems with a linear part that can be written in ARX form, the state-space models that are obtained using N4SID algorithms cover the entire field of linear systems. Hence, if the true system has a linear part which is outside the ARX-class, the N4SID Hammerstein algorithm can be expected to outperform the ARX Hammerstein algorithm. If the true system lies within the ARX-class the greater flexibility that comes with the use of subspace identification and state-space models only leads to an unnecessarily high number of parameters and consequently a higher variance on the model estimates than what would be obtained when using the ARX Hammerstein algorithm.

In Chapter 9, finally, an algorithm was proposed for the identification of Hammerstein-Wiener systems based on kernel Canonical Correlation Analysis and LS-SVM regression. In contrast to existing methods, the proposed algorithm does not rely on restrictive assumptions on the inputs and can be applied to SISO as well as MIMO models. Again, the use of kernels in a primal-dual framework was seen to lead to a reliable identification algorithm.

Future research

Future research for Part I

The results on using regularization to deal with possible ill-conditioning in combined stochastic-deterministic subspace identification can and should be studied further. The following is a non-exhaustive list of possibilities:

1. Study the effect of regularization in the oblique projection on the obtained state. Is the unifying theorem still valid? Does the use of regularization simply lead to a change in basis? If so, what is the corresponding W_2.

2. Use regularization in the oblique projection but try to avoid using the

separate parameterization. Is it possible to use a second regularization step to deal with stochastic components of the system being incorrectly attributed to the system inputs?

Future research for Part II

In the second part of this thesis it was shown that using combinations of classical linear identification techniques and LS-SVM regression, a set of structured non-linear systems could conveniently be identified. It is clear that not all possibilities for the extension of linear subspace identification techniques to non-linear systems are hereby exhausted. Three clear possibilities for future work emerge:

1. In Chapter 7 on Hammerstein ARX identification it was seen that collinearity constraints on different vectors $w_j, j = 0, \ldots, m$ should in principle be imposed to ensure that the estimated model in the first step is in fact a Hammerstein model. It was there-after argued that adding these constraints would lead to a difficult to solve optimization problem. Hence, the constraints were dropped and it was hoped that collinearity would automatically be preserved when over-parameterizing the Hammerstein model in the first step. Although the resulting approach was seen to yield good results throughout part II of the thesis, it would still be preferable to find a way to impose the collinearity constraints directly.

2. In [151] some preliminary results were presented extending the ideas surrounding Hammerstein-Wiener identification in part II of the thesis to general non-linear models. This is a promising research area as it would enable the use of subspace identification algorithms for virtually any non-linear system. On the other hand, with the decrease in structure when moving from Hammerstein-Wiener systems to general non-linear systems, the variance on the obtained models can be expected to increase considerably. Hence, it remains to be seen whether subspace identification algorithms for general non-linear systems are useful in practice.

3. Instead of extending some of the presented results to general non-linear systems, it is worthwhile to examine whether the algorithms presented in part II of this thesis can be extended to other structured non-linear model classes, such as the Wiener-Hammerstein class, characterized by a Wiener model, followed by a Hammerstein model. Instead of only allowing an additive structure in the estimated non-linearities, one could investigate more complicated structures allowing for instance certain multiplications between inputs at various time-instances.

Appendix A

About the alternative form of the Kalman filter

In this appendix, two different forms of the recursive Kalman filter are discussed. The first is the classical form as for instance found in [9]. This form is transformed into the form of (3.33-3.35) which is more useful in the derivation of subspace identification algorithms.

A.1 Derivation of the special form

Consider the system (3.26), where we assume that A, C, Q, R and S are known. Given $\hat{x}_0$, $\widetilde{P}_0$ and $u_0, \ldots, u_{t-1}, y_0, \ldots, y_{t-1}$ the non-steady state Kalman filter state estimate $\hat{x}_t$ is given by the following set of recursive formulas [9]:

$$\hat{x}_t = A\hat{x}_{t-1} + Bu_{t-1} + K_{t-1}(y_{t-1} - C\hat{x}_{t-1} - Du_{t-1}),$$

with:

$$\begin{aligned} K_{t-1} &= (A\widetilde{P}_{t-1}C^T + S)(C\widetilde{P}_{t-1}C^T + R)^{-1}, && \text{(A.1)} \\ \widetilde{P}_k &= A\widetilde{P}_{-1}A^T + Q \\ &- (A\widetilde{P}_{t-1}C^T + R)(C\widetilde{P}_{t-1}C^T + S)(A\widetilde{P}_{t-1}C^T + R)^T. && \text{(A.2)} \end{aligned}$$

and $\widetilde{P}_t$ the error covariance matrix:

$$\widetilde{P}_t = E\left\{(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T\right\}.$$

In the derivations of subspace identification, a different form of these recursive Kalman filter equations is more useful. With A, C, Q, R, S given, the matrices $P^s = E\left\{x_t^s x_t^{sT}\right\}$, $G = E\left\{x_{t+1}^s y_t^s\right\}$ and $\Lambda_0 = E\left\{y_t^s y_t^{sT}\right\}$ can be computed as:

$$P^s = AP^sA^T + Q,$$

$$\begin{aligned} G &= AP^sC^T + R, \\ \Lambda_0 &= CP^sC^T + S. \end{aligned}$$

Defining

$$P_t = P^s - \widetilde{P}_t$$

this leads to

$$\begin{aligned} K_{t-1} &= ((AP^sC^T + R) - AP_{t-1}C^T)((CP^sC^T + S) - CP_{t-1}C^T)^{-1} \\ &= (G - AP_{t-1}C^T)(\Lambda_0 - CP_{t-1}C^T)^{-1}. \end{aligned}$$

For the Riccati equation (from (A.2)) we have:

$$\begin{aligned} P^s - P_t &= AP^sA^T - AP_{t-1}A^T + (P^s - AP^sA^T) && \text{(A.3)} \\ &- ((AP^sC^T + R) - AP_{t-1}C^T) \\ &\quad ((CP^sC^T + S) - CP_{t-1}C^T)^{-1} \\ &\quad ((AP^sC^T + R) - AP_{t-1}C^T)^T, \\ P_t &= AP_{t-1}A^T && \text{(A.4)} \\ &- (G - AP_{t-1}C^T)(\Lambda_0 - CP_{t-1}C^T)^{-1}(G - AP_{t-1}C^T)^T. \end{aligned}$$

Hence, the Kalman filter (A.3-A.4) calculates the same state estimate $\hat{x}_t$ as the original Kalman filter (A.1-A.2) with $\widetilde{P}_0 = P^s - P_0$.

Appendix B

The Schur complement

In this appendix, we introduce the Schur complement of a matrix and introduce an interesting property related to the conservation of positive definiteness when applying the Schur complement.

B.1 The Schur complement of a matrix

Suppose we partition a matrix $A \in \mathbb{R}^{n\times n}$

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

where $A_{11} \in \mathbb{R}^{r\times r}$. Assuming that A_{11} is non-singular,

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}$$

is called the Schur complement of A_{11} in A. Likewise if A_{22} is non-singular,

$$T = A_{11} - A_{12}A_{22}^{-1}A_{21}$$

is called the Schur complement of A_{22} in A.

B.2 Relation to positive definiteness of a matrix

It can be shown that if A is positive (semi)-definite, so are the Schur complements S and T [81]. Alternatively, if S and A_{11} are positive (semi)-definite, so is A [81]. Likewise, if T and A_{22} are positive (semi)-definite, the same goes for A [81]. These relations are commonly used when trying to proof (semi)-positive definiteness of a matrix and are essential in the proof of Theorem 4.1 in this thesis.

Appendix C

Tikhonov regularization

In this appendix, we will briefly discuss the idea of Tikhonov regularization. Tikhonov regularization is mostly used as a means to deal with ill-conditioned least-squares regression problems by penalizing solutions with large norms. It will be shown in the following sections that the variance on the estimated parameters in an ill-conditioned least-squares regression problems can considerably be decreased when using Tikhonov regularization. This however, at the expense of the introduction of a small bias. Finding a good balance between variance reduction and increase in bias is known as the bias-variance trade-off, and is the key idea behind any type of regularization approach.

C.1 Tikhonov regularization and its effect on the Hessian

Given $A \in \mathbb{R}^{N \times n}$ with $N \geq n$ and $b \in \mathbb{R}^N$, the aim of linear least-squares is to find an estimate $x \in \mathbb{R}^n$ such that:

$$x_{\mathrm{LS}} = \arg\min_x \|Ax - b\|_2^2.$$

A well known drawback with least-squares problems is that they yield large variances on the coefficients of the obtained solution x in case of a near collinearity in the columns of A. In this case, the condition number of A is known to be high, and the problem is said to be ill-conditioned. To deal with this ill-conditioning, and reduce the variance on the obtained parameters, in Tikhonov regularization the least-squares cost function is replaced by

$$\|Ax - b\|_2^2 + \gamma \|x\|_2^2, \tag{C.1}$$

with $\|x\|_2^2$ the so-called regularization term and $\gamma > 0$ the regularization constant. The extra regularization term reduces the effects of near collinearity in the columns of A by effectively favoring solutions x with a small 2-norm over those with larger norms. The resulting problem (C.1) is referred to as a ridge regression problem and its solution is given by

$$x_{\mathrm{RR}} = (A^T A + \gamma I_n)^{-1} A^T b.$$

The positive effect of adding a regularization term is apparent from the Hessian $H = A^T A + \gamma I_n$. In 2.1.3, it was argued that H offers insight into the shape of the cost-function close to its minimum, with $\mathrm{Cond}(H)$ a measure for the conditioning of the optimization problem. In the ridge regression case we have

$$\mathrm{Cond}(H) = \frac{\sigma_{\max}(A^T A + \gamma I_n)}{\sigma_{\min}(A^T A + \gamma I_n)} = \frac{\sigma^2_{\max}(A) + \gamma}{\sigma^2_{\min}(A) + \gamma}.$$

Clearly, for $\sigma_{\min}(A)$ small, a limited amount of regularization is sufficient to significantly decrease the condition number of H.

C.2 Variance on the obtained solution

As in the unregularized least-squares case, one can calculate the expected sensitivity of the solution x_{RR} to a perturbation δb in b with $\delta b(\delta b)^T = \sigma_b^2 I_N$. Assuming that $x_{\mathrm{RR}} = (A^T A + \gamma I_n)^{-1} A^T b$ and $x_{\mathrm{RR}} + \delta x_{\mathrm{RR}} = (A^T A + \gamma I_n)^{-1} A^T (b + \delta b)$, it follows that:

$$\delta x_{\mathrm{RR}} = (A^T A + \gamma I_n)^{-1} A^T \delta b,$$

from which

$$E\left\{\delta x_{\mathrm{RR}} (\delta x_{\mathrm{RR}})^T\right\} = \sigma_b^2 (A^T A + \gamma I_n)^{-1} A^T A (A^T A + \gamma I_n)^{-1}$$

With $A = USV^T$ the singular value decomposition of A, we have:

$$\begin{aligned}&(A^T A + \gamma I_n)^{-1} A^T A (A^T A + \gamma I_n)^{-1} = \\ &\qquad (V(S^2 + \gamma I_n)V^T)^{-1} V S^2 V^T (V(S^2 + \gamma I_n)V^T)^{-1} = V \frac{S^2}{(S^2 + \gamma I_n)^2} V^T.\end{aligned}$$

For any singular value σ of A, the function $\sigma \to \frac{(\sigma^2+\gamma)}{\sigma}$ is strictly increasing. Furthermore, the effect of Tikhonov regularization is larger for the smaller singular values. Hence, adding a regularization term decreases the variance on the obtained parameters.

C.3 Weighted Tikhonov regularization

Weighted Tikhonov regularization is in essence the same as Tikhonov regularization with the exception that a weighting matrix is used to penalize the elements of x. The resulting optimization problem, which is also known as a generalized ridge regression problem is given as

$$x_{GRR} = \arg\min_{x} \|Ax - b\|_2^2 + \gamma x^T W x,$$

with $W \in \mathbb{R}^{n \times n}$ a positive semi-definite weighting matrix. The solution is given as

$$x_{GRR} = (A^T A + \gamma W)^{-1} A^T b, \tag{C.2}$$

with Hessian

$$H = A^T A + \gamma W. \tag{C.3}$$

Note that for $W = I_n$, generalized ridge regression reduces to ordinary ridge regression. However, the addition of an extra weighting matrix allows to specifically focus the regularization effort on those components of x which exhibit large variances or which are preferably kept small (in absolute value) for any other reason. Hence, weighted Tikhonov regularization is particularly useful in the case of available prior knowledge.

Unlike for ordinary ridge regression, the resulting condition number for the Hessian $H = A^T A + \gamma W$ and the error covariance matrix $E\left\{\delta x_{\mathrm{GRR}} (\delta x_{\mathrm{GRR}})^T\right\}$ can in general not straightforwardly be derived in terms of the singular values of A and the elements of W. However, for some choices of A and W easily interpretable results can be obtained. An example is found when analyzing the oblique projection in Chapter 5.

Appendix D

Proofs

D.1 Proof of Lemma 5.1

Without loss of generality, we assume $i_1 \leq i_2$. For A rank deficient, the condition number will go to infinity, and the proof is obvious. For A of full rank, it is easily seen that for any matrix $B \in \mathbb{R}^{i\times N}$ with $\operatorname{rank}(B) < i \leq N$, we have

$$\|A-B\|_2 = \max_{x,\|x\|_2=1} \|A^T x - B^T x\|_2 \geq \|A^T x_2 - B^T x_2\|_2 = \|A^T x_2\|_2 \geq \sigma_i(A),$$

where $x, x_2 \in \mathbb{R}^i$, $x_2 \in \mathrm{null(B^T)}$, $\|x_2\|_2 = 1$. Hence, the following inequality holds:

$$\operatorname{Cond}(A) = \frac{\sigma_1(A)}{\sigma_i(A)} \geq \frac{\|A\|_2}{\|A-B\|_2}, \quad \forall B : \operatorname{rank}(B) < i.$$

Assume that the rows of $V^{(1)} \in \mathbb{R}^{i_1\times N}$ and $V^{(2)} \in \mathbb{R}^{i_2\times N}$ for orthonormal basisses for $\operatorname{Row}(A(I_1,:))$ and $\operatorname{Row}(A(I_2,:))$, respectively, so that

$$\begin{aligned}
A(I_1,:) &= S_1 V^{(1)}, \\
A(I_2,:) &= S_2 V^{(2)}, \\
V^{(1)} {V^{(2)}}^T &= \begin{bmatrix} \Lambda & 0_{i_1\times(i_2-i_1)} \end{bmatrix}, \\
V^{(1)} {V^{(1)}}^T &= I_{i_1}, \\
V^{(2)} {V^{(2)}}^T &= I_{i_2},
\end{aligned}$$

where S_1, S_2 are of full rank and Λ is a $i_1 \times i_1$ diagonal matrix containing the cosines of the principal angles between $\operatorname{Row}(A(I_1,:))$ and $\operatorname{Row}(A(I_2,:))$. Define for $k = 1, \ldots, i_1$

$$\begin{aligned}
V^{(2)}_{L,k} &= V^{(2)}(1:k-1,:), \\
V^{(2)}_{M,k} &= V^{(2)}(k,:)/V^{(1)}(k,:) = V^{(2)}(k,:)V^{(1)}(k,:)^T V^{(1)}(k,:), \\
V^{(2)}_{R,k} &= V^{(2)}(k+1:i_2,:).
\end{aligned}$$

We have:

$$
\begin{aligned}
\mathrm{Cond}(A) &\geq \mathrm{Cond}\left(\begin{bmatrix} A(I_1,:) \\ A(I_2,:) \end{bmatrix}\right), \\
&\geq \frac{\left\| \begin{bmatrix} S_1 & 0_{i_1 \times i_2} \\ 0_{i_2 \times i_1} & S_2 \end{bmatrix} \begin{bmatrix} V^{(1)} \\ V^{(2)} \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} S_1 & 0_{i_1 \times i_2} \\ 0_{i_2 \times i_1} & S_2 \end{bmatrix} \left(\begin{bmatrix} V^{(1)} \\ V^{(2)} \end{bmatrix} - \begin{bmatrix} V^{(1)} \\ V^{(2)}_{L,k} \\ V^{(2)}_{M,k} \\ V^{(2)}_{R,k} \end{bmatrix} \right) \right\|_2}, \\
&= \frac{\left\| \begin{bmatrix} S_1 & 0_{i_1 \times i_2} \\ 0_{i_2 \times i_1} & S_2 \end{bmatrix} \begin{bmatrix} V^{(1)} \\ V^{(2)} \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} S_1 & 0_{i_1 \times i_2} \\ 0_{i_2 \times i_1} & S_2 \end{bmatrix} \begin{bmatrix} 0_{i_1+k-1,N} \\ V^{(2)}(k,:)/V^{(1)}(k,:)^{\perp} \\ 0_{i_2-k,N} \end{bmatrix} \right\|_2}, \\
&\geq \frac{\|S_2\|_2}{\|V^{(2)}(k,:)/V^{(1)}(k,:)^{\perp}\| \|S_2\|_2}, \\
&= \frac{\|S_2\|_2}{\sin\left(V^{(1)}(k,:) \sphericalangle V^{(2)}(k,:)\right) \|S_2\|_2}, \\
&= \frac{1}{\sin\left(V^{(1)}(k,:) \sphericalangle V^{(2)}(k,:)\right)},
\end{aligned}
$$

which is valid for any principal angle between $A(I_1,:)$ and $A(I_2,:)$, and hence also for the smallest one.

D.2 Proof of Lemma 5.2

Denoting for convenience $U = \begin{bmatrix} U_p \\ U_f \end{bmatrix}$, we have

$$\begin{aligned}
Y_f^d/(W_p/U_f^\perp) &= (Y_f/U)\,/(W_p/U_f^\perp) \\
&= \big(Y_f/(U/U_f^\perp) + Y_f/(U/U_f)\big)\,/(W_p/U_f^\perp) \\
&= \big(Y_f/(U_p/U_f^\perp) + Y_f/(U/U_f)\big)\,/(W_p/U_f^\perp) \\
&= \big(Y_f/(U_p/U_f^\perp)\big)\,/(W_p/U_f^\perp) \\
&= Y_f/(U_p/U_f^\perp),
\end{aligned}$$

where the following properties were used:

- $A/\begin{bmatrix} B \\ C \end{bmatrix} = A/B + A/C$ if $B \perp C$.
- $\mathrm{Row}(U_p) \subset \mathrm{Row}(W_p) \Rightarrow \mathrm{Row}(U_p/U_f^\perp) \subset \mathrm{Row}(W_p/U_f^\perp)$
- $\mathrm{Row}(U/U_f^\perp) = \mathrm{Row}\left(\begin{bmatrix} U_p \\ U_f \end{bmatrix}/U_f^\perp\right) = \mathrm{Row}(U_p/U_f^\perp)$.

Furthermore, we have:

$$\begin{aligned}
\mathrm{row}(U_p/U_f^\perp) &= \mathrm{Row}\left(\begin{bmatrix} U_p/U_f^\perp \\ Y_p/(U_p/U_f^\perp) \end{bmatrix}\right) \\
&= \mathrm{Row}\left(\begin{bmatrix} U_p/U_f^\perp \\ Y_p/(U/U_f) + Y_p/(U/U_f^\perp) - Y_p/(U/U_f) \end{bmatrix}\right) \\
&= \mathrm{Row}\left(\begin{bmatrix} U_p - U_p/U_f \\ Y_p/U - Y_p/(U/U_f) \end{bmatrix}\right) \\
&= \mathrm{Row}\left((W_p - W_p/U_f)/U\right) \\
&= \mathrm{Row}\big((W_p/U_f^\perp)/U\big)\,.
\end{aligned}$$

Hence, finally we obtain the relation:

$$\begin{aligned}
Y_f^d/(W_p/U_f^\perp) &= Y_f/(U_p/U_f^\perp) \\
&= Y_f/\left((W_p/U_f^\perp)/U\right) \\
&= \big(Y_f/\left((W_p/U_f^\perp)/U\right)\big)\,/U \\
&= \big(Y_f/(W_p/U_f^\perp) - Y_f/\left((W_p/U_f^\perp)/U^\perp\right)\big)\,/U \\
&= \big(Y_f/(W_p/U_f^\perp)\big)\,/U,
\end{aligned}$$

which proofs the Lemma.

D.3 Proof of Lemma 5.3

$$\begin{aligned}
&E\left\{\left(\delta\begin{bmatrix}L_1^\gamma(t,:) & L_2^\gamma(t,:)\end{bmatrix}\right)^T\left(\delta\begin{bmatrix}L_1^\gamma(t,:) & L_2^\gamma(t,:)\end{bmatrix}\right)\right\} \\
&= \sigma_y^2\left(\begin{bmatrix}I_{im} & \Lambda & 0\\ \Lambda & (1+\gamma)I_{im} & 0\\ 0 & 0 & (1+\gamma)I_{il}\end{bmatrix}\right)^{-1}\begin{bmatrix}I_{im} & \Lambda & 0\\ \Lambda & I_{im} & 0\\ 0 & 0 & I_{il}\end{bmatrix} \\
&\quad \left(\begin{bmatrix}I_{im} & \Lambda & 0\\ \Lambda & (1+\gamma)I_{im} & 0\\ 0 & 0 & (1+\gamma)I_{il}\end{bmatrix}\right)^{-1} \\
&= \sigma_y^2\begin{bmatrix}\frac{(1+\gamma)I_{im}}{(1+\gamma)I_{im}-\Lambda^2} & \frac{-\Lambda}{(1+\gamma)I_{im}-\Lambda^2} & 0\\ \frac{-\Lambda}{(1+\gamma)I_{im}-\Lambda^2} & \frac{I_{im}}{(1+\gamma)I_{im}-\Lambda^2} & 0\\ 0 & 0 & \frac{1}{1+\gamma}I_{il}\end{bmatrix}\begin{bmatrix}I_{im} & \Lambda & 0\\ \Lambda & I_m & 0\\ 0 & 0 & I_{il}\end{bmatrix} \\
&\quad \begin{bmatrix}\frac{(1+\gamma)I_{im}}{(1+\gamma)I_{im}-\Lambda^2} & \frac{-\Lambda}{(1+\gamma)I_{im}-\Lambda^2} & 0\\ \frac{-\Lambda}{(1+\gamma)I_{im}-\Lambda^2} & \frac{I_{im}}{(1+\gamma)I_{im}-\Lambda^2} & 0\\ 0 & 0 & \frac{1}{1+\gamma}I_{il}\end{bmatrix} \\
&= \sigma_y^2\begin{bmatrix}\frac{(1+\gamma)I_{im}}{(1+\gamma)I_{im}-\Lambda^2} & \frac{-\Lambda}{(1+\gamma)I_{im}-\Lambda^2} & 0\\ \frac{-\Lambda}{(1+\gamma)I_{im}-\Lambda^2} & \frac{I_{im}}{(1+\gamma)I_{im}-\Lambda^2} & 0\\ 0 & 0 & \frac{1}{1+\gamma}I_{il}\end{bmatrix} \\
&\quad \begin{bmatrix}I_{im} & 0 & 0\\ \frac{\gamma\Lambda}{(1+\gamma)I_{im}-\Lambda^2} & \frac{I-\Lambda^2}{(1+\gamma)I_{im}-\Lambda^2} & 0\\ 0 & 0 & \frac{1}{1+\gamma}I_{il}\end{bmatrix} \\
&= \sigma_y^2\begin{bmatrix}\frac{(1+\gamma)^2I_{im}-(1+2\gamma)\Lambda^2}{((1+\gamma)I_{im}-\Lambda^2)^2} & \frac{\Lambda^3-\Lambda}{((1+\gamma)I_{im}-\Lambda^2)^2} & 0\\ \frac{\Lambda^3-\Lambda}{((1+\gamma)I_{im}-\Lambda^2)^2} & \frac{I_{im}-\Lambda^2}{((1+\gamma)I_{im}-\Lambda^2)^2} & 0\\ 0 & 0 & \frac{1}{(1+\gamma)^2}I_{il}\end{bmatrix}.
\end{aligned}$$

Bibliography

[1] M. Abdelghani, M. Verhaegen, P. Van Overschee, and B. De Moor. Comparison study of subspace identification methods applied to flexible structures. *Mechanical systems and signal processing*, 12(5):679–692, 1998.

[2] H. Akaike. Statistical predictor identification. *Annals of the Institute for Statistical Mathematics*, 22:203–217, 1973.

[3] H. Akaike. Stochastic theory of minimal realization. *IEEE Transactions on Automatic Control*, 19(6):667–673, 1974.

[4] H. Akaike. Markovian representation of stochastic processes by canonical variables. *SIAM Journal on Control*, 13(1):162–173, 1975.

[5] T.W. Anderson. *An Introduction to Multivariate Statistical Analysis.* Wiley, New York, 1984.

[6] M. Aoki. *State Space Modeling of Time Series.* Springer Verlag, Berlin, 1987.

[7] G. Arfken. *Mathematical Methods for Physicists.* Academic Press, Orlando, Fl, 3rd edition, 1985.

[8] K.S. Arun and S.Y. Kung. Balanced approximation of stochastic systems. *SIAM Journal on Matrix Analysis and Applications*, 11:42–68, 1990.

[9] K. Astrom and B. Wittenmark. *Computer Controlled Systems: Theory and Design.* Prentice Hall, 1984.

[10] F.R. Bach and M.I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[11] T. Backx. *Identification of an Industrial Process: A Markov Parameter Approach.* PhD thesis, Technical University Eindhoven, The Netherlands, 1987.

[12] E.W. Bai. An optimal two-stage identification algorithm for Hammerstein-Wiener nonlinear systems. *Automatica*, 4(3):333–338, 1998.

[13] E.W. Bai. A blind approach to Hammerstein model identification. *IEEE Transactions on Signal Processing*, 50(7):1610–1619, 2002.

[14] E.W. Bai. A blind approach to the Hammerstein-Wiener model identification. *Automatica*, 38:967–979, 2002.

[15] M. Basseville, M. Abdelghani, and A. Benveniste. Subspace-based fault detection algorithms for vibration monitoring. *Automatica*, 36(1):101–109, Jan. 2000.

[16] M. Basseville, A. Benveniste, M. Goursat, L. Hermans, L. Mevel, and H. Van der Auweraer. Output-only subspace-based structural identification: from theory to industrial testing practice. *ASME Journal of Dynamic Systems Measurement and Control, Special Issue on Identification of Mechanical Systems*, 123(4):668–676, Dec. 2001.

[17] S.A. Billings and S.Y. Fakhouri. Identification of a class of non-linear systems using correlation analysis. *Proceedings of IEE*, 125(7):697–697, 1978.

[18] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[19] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[20] F.H.I. Chang and R. Luus. A noniterative method for identification using the Hammerstein model. *IEEE Transactions on Automatic Control*, 16:464–468, 1971.

[21] A. Chiuso and G. Picci. Subspace identification by orthogonal decomposition. In *Proceedings of the 14'th IFAC World Congress*, volume 1, pages 241–246, 1999.

[22] A. Chiuso and G. Picci. Some algorithmic aspects of subspace identification with inputs. *International Journal of Applied Mathematics and Computer Science*, 11(1):55–75, 2001.

[23] A. Chiuso and G. Picci. Asymptotic variance of subspace methods by data orthogonalization and model decoupling: a comparative analysis. *Automatica*, 40(10):1705–1717, 2004.

[24] A. Chiuso and G. Picci. Numerical conditioning and asymptotic variance of subspace estimates. *Automatica*, 40(4):677–683, 2004.

[25] A. Chiuso and G. Picci. On the ill-conditioning of subspace identification with inputs. *Automatica*, 40(4):575–589, 2004.

[26] A. Chiuso and G. Picci. Subspace identification by data orthogonalization and model decoupling. *Automatica*, 40(10):1689–1703, 2004.

[27] Y.M. Cho, G. Xu, and T. Kailath. Fast recursive identification of state space models via exploitation of displacement structure. *Automatica,* Special Issue on Statistical Signal Processing and Control, 30(1):45–59, 1994.

[28] N. L. C. Chui and J.M. Maciejowski. Realization of stable models with subspace methods. *Automatica*, 32:1587–1595, 1996.

[29] P. Crama. *Identification of block-oriented nonlinear models.* PhD thesis, Vrije Universiteit Brussel, Dept. ELEC, June 2004.

[30] P. Crama and J. Schoukens. Hammerstein-Wiener system estimator initialization. In *Proc. of the International Conference on Noise and Vibration Engineering (ISMA2002), Leuven*, pages 1169–1176, 16-18 September 2002.

[31] P. Crama and J. Schoukens. Initial estimates of Wiener and Hammerstein systems using multisine excitation. *IEEE Transactions on Measurement and Instrumentation*, 50(6):1791–1795, 2001.

[32] H. Cramer. *Mathematical Methods of Statistics.* Princeton University Press, Princeton, N.J., 1946.

[33] A. Dahlén, A. Lindquist, and J. Mari. Experimental evidence showing that stochastic subspace identification methods may fail. *Systems and Control Letters*, 34:303–312, 1998.

[34] B.W. Datta. *Numerical Linear Algebra.* Brooks/Cole Publishing Company, 1995.

[35] K. De Cock. *Principal Angles in System Theory, Information Theory and Signal Processing.* PhD thesis, Katholieke universiteit Leuven, K.U.Leuven (Leuven, Belgium), May 2002.

[36] K. De Cock and B. De Moor. Subspace angles between ARMA models. *Systems and Control Letters*, 46:265–270, 2002.

[37] K. De Cock and B. De Moor. Canonical correlations between input and output processes of linear stochastic models. In *Proceedings of the Fifteenth International Symposium on the Mathematical Theory of Networks and Systems (MTNS 2002), University of Notre Dame*, Aug 12-16, 2002.

[38] B. De Moor. *Mathematical Concepts and Techniques for Modeling of Static and Dynamic Systems.* PhD thesis, Katholieke Universiteit Leuven, K.U.Leuven (Leuven, Belgium), 1988.

[39] B. De Moor, M. Moonen, L. Vandenberghe, and J. Vandewalle. The application of the canonical correlation concept to the identification of linear state space models. chapter in *Analysis and Optimization of*

Systems, A. Bensousan, J.L. Lions (Eds.), Springer Verlag, Heidelberg, pages 1103–1114, 1988.

[40] B. De Moor, M. Moonen, L. Vandenberghe, and J. Vandewalle. A geometrical approach for the identification of state space models with singular value decomposition. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* New York, pages 2244–2247, 1988.

[41] B. De Moor, M. Moonen, L. Vandenberghe, and J. Vandewalle. Identification of linear state space models with singular value decomposition using canonical correlation concepts. chapter in *SVD and Signal Processing: Algorithms, Applications and Architectures,* E. Deprettere (Ed.), Elsevier Science Publishers B.V. (North-Holland), pages 161–169, 1988.

[42] B. De Moor, M. Moonen, L. Vandenberghe, and J. Vandewalle. On and off-line identification of linear state space models. *International Journal of Control*, 49(1):219–232, 1989.

[43] B. De Moor and J. Vandewalle. A geometrical strategy for the identification of state space models of linear multivariable systems with singular value decomposition. *Proc. of the 3rd International Symposium on Applications of Multivariable Systems Techniques,* April 13-15, Plymouth, UK, pages 59–69, 1987.

[44] B. De Moor, J. Vandewalle, M. Moonen, L. Vandenberghe, and P.A. Van Mieghem. A geometrical approach for the identification of state space models with singular value decomposition. *Symposium on Identification and System Parameter Estimation,* 27-31 August, Beijing, China, pages 700–704, 1988.

[45] B. De Moor (Ed.). *DaISy: Database for the Identification of Systems, Department of Electrical Engineering*, ESAT/SISTA, K.U.leuven, Belgium, url: `http://www.esat.kuleuven.ac.be/sista/daisy/`, feb. 11, 2005. [used dataset: Data from a flexible robot arm, section: Mechanical systems, 96-009].

[46] J.W. Demmel. *Applied Numerical Linear Algebra.* SIAM Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[47] E.J. Dempsey and D.T. Westwick. Identification of Hammerstein models with cubic spline nonlinearities. *IEEE Transactions on Biomedical Engineering*, 51:237–245, 2004.

[48] A.H. Falkner. Iterative technique in the identification of a non-linear system. *International Journal of Control*, 1:385–396, 48.

[49] P. Faurre. Stochastic realization algorithms. chapter in *System identification: Advances and Case studies,* R.K. Mehra and D.G. Lainiotis (Eds.), Academic Press, New York, 1976.

[50] P. Faurre, M. Clerget, and F. German. *Opérateurs rationnels positifs, application à l'hyperstabilité et aux processus aléatoires* 8. Dunod, 1978.

[51] W. Favoreel, B. De Moor, and P. Van Overschee. Subspace identification of bilinear systems subject to white inputs. *IEEE Transactions on Automatic Control*, 44(6):1157–1165, 1999.

[52] W. Favoreel, S. Van Huffel, B. De Moor, S. Vasile, and M. Verhaegen. Comparative study between three different subspace identification algorithms. In *Proceedings of the European Control Conference 1999 (ECC-99), Karlsruhe, Germany*, August 1999.

[53] G.F. Franklin, J.D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems,* Third Edition. Addison-Wesley, 1994.

[54] M. Gevers and T. Kailath. An innovations approach to least-squares estimation, part vi: Discrete-time innovations representations and recursive estimation. *IEEE Transactions on Automatic Control*, 18:588–600, 1973.

[55] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

[56] I. Goethals and B. De Moor. Some comments on the definition of a Total Least Squares condition number as the condition number of an equivalent Least Squares problem. Technical Report 05-32, ESAT-SISTA, K.U.Leuven (Leuven Belgium), 2005, *available online at* `ftp.esat.kuleuven.ac.be/pub/SISTA/goethals/condition.ps`.

[57] I. Goethals and B. De Moor. First order perturbation analysis of data-driven subspace algorithms. Technical Report 05-31, ESAT-SISTA, K.U.Leuven (Leuven Belgium), 2005, *available online at* `ftp.esat.kuleuven.ac.be/pub/SISTA/goethals/perturbation.ps`.

[58] I. Goethals and B. De Moor. Model reduction and energy analysis as a tool to detect spurious modes. In *Proceedings of the International Conference on Noise and Vibration Engineering (ISMA), Leuven, Belgium*, Juni 2002.

[59] I. Goethals and B. De Moor. Subspace identification combined with new mode selection techniques for modal analysis of an airplane. In *Proceedings of the 13th IFAC symposium on system identification (SYSID 2003), Rotterdam, the Netherlands*, pages 695–700, Sep. 2003.

[60] I. Goethals, L. Hoegaerts, J.A.K. Suykens, V. Verdult, and B. De Moor. Hammerstein-Wiener subspace identification using kernel Canonical Correlation Analysis. Technical Report 05-30, ESAT-SISTA, K.U.Leuven (Leuven Belgium), 2005 *available online at* `ftp.esat.kuleuven.ac.be/pub/SISTA/goethals/goethals_hammer_wiener.ps`.

[61] I. Goethals, L. Mevel, A. Benveniste, and B. De Moor. Recursive output-only subspace identification for in-flight flutter monitoring. In *Proceedings of the 22nd International Modal Analysis Conference (IMAC-XXII), Dearborn, Michigan*, Jan. 2004.

[62] I. Goethals, K. Pelckmans, J.A.K. Suykens, and B. De Moor. Identification of MIMO Hammerstein models using least squares support vector machines. Technical Report 04-45, ESAT-SISTA, K.U.Leuven (Leuven Belgium), *Accepted for publication in Automatica*, 2004.

[63] I. Goethals, K. Pelckmans, J.A.K. Suykens, and B. De Moor. Subspace identification of Hammerstein systems using least squares support vector machines. Technical Report 04-114, ESAT-SISTA, K.U.Leuven (Leuven Belgium), *Submitted for publication*, 2004.

[64] I. Goethals, K. Pelckmans, J.A.K. Suykens, and B. De Moor. NARX identification of Hammerstein models using least squares support vector machines. In *Proceedings of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2004), Stuttgart, Germany*, pages 507–512, Sep. 2004.

[65] I. Goethals, T. Van Gestel, J.A.K. Suykens, P. Van Dooren, and B. De Moor. Identification of positive real models in subspace identification by using regularization. *IEEE Transactions on Automatic Control*, 48(10):1843–1847, 2003.

[66] I. Goethals, T. Van Gestel, J.A.K. Suykens, P. Van Dooren, and B. De Moor. Identifying positive real models in subspace identification by using regularization. In *Proceedings of the 13th System Identification Symposium (SYSID2003), Rotterdam, The Netherlands*, pages 1411–1416, Aug. 2003.

[67] I. Goethals, B. Vanluyten, and B. De Moor. Reliable spurious mode rejection using self learning algorithms. In *Proceedings of the International Conference on Noise and Vibration Engineering (ISMA 2004), Leuven, Belgium*, pages 991–1003, Sep. 2004.

[68] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 1989.

[69] G.H. Golub and H. Zha. The canonical correlations of matrix pairs and their numerical computation. chapter in *Linear Algebra for Signal Processing,* A. Bojanczyk and G. Cybenko (Eds.), Springer, New York, pages 59–82, 1995.

[70] W. Greblicki. Non-parametric orthogonal series identification of Hammerstein systems. *International Journal of Systems Science*, 20(12):2355–2367, 1989.

[71] W. Greblicki. Nonparametric identification of Wiener systems by orthogonal series. *IEEE Transactions on Automatic Control*, 39(10):2077–2086, 1994.

[72] W. Greblicki and M. Pawlak. Identification of discrete Hammerstein systems using kernel regression estimates. *IEEE Transactions on Automatic Control*, 31:74–77, 1986.

[73] W. Greblicki and M. Pawlak. Nonparametric identification of a cascade nonlinear time series system. *Signal Processing*, 22:61–75, 1991.

[74] S.F. Gull. Bayesian inductive inference and maximum entropy. chapter in *Maximum-Entropy and Bayesian Methods in Science and Engineering,* G.J. Erickson and R. Smith (Eds.), Kluwer, Dordrecht, 1:53–74, 1988.

[75] J. C. Gmez and E. Baeyens. Subspace identification of multivariable Hammerstein and Wiener models. In *Proceedings of the 15th IFAC World Congress, Barcelona, Spain*, 2002.

[76] P. C. Hansen. SVD-theory and applications. Technical Report 84-05, Technical High School, Lyngby, Denmark, 1984.

[77] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, Heidelberg, 2001.

[78] Z. Hasziewicz. Hammerstein system identification by the Haar multiresolution approximation. *International Journal of Adaptive Control and Signal Processing*, 13(8):691–717, 1999.

[79] B.L. Ho and R.E. Kalman. Effective construction of linear state-variable models from input-output functions. *Regelungstechnik*, 12:545–548, 1965.

[80] L. Hoegaerts, J.A.K. Suykens, J. Vandewalle, and B. De Moor. Subset based least squares subspace regression in rkhs. *Neurocomputing*, 63:293–323, 2005.

[81] R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, New York, New Rochelle, Melbourne, Sydney, 1985.

[82] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–372, 1936.

[83] A. Janczak. Neural network approach for identification of Hammerstein systems. *International Journal of Control*, 76(17):1749–1766, 2003.

[84] M. Jansson. Subspace identification and ARX modeling. In *13'th IFAC Symposium on System Identification, Rotterdam, The Netherlands*, 2003.

[85] M. Jansson. A linear regression approach to state-space subspace system identification. *Signal Processing*, 52(2):103–129, July 1996.

[86] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear estimation.* Prentice Hall, Upper Saddle River, New Jersey, 2000.

[87] A.D. Kalafatis, L. Wang, and W.R. Cluett. Identification of Wiener-type nonlinear systems in a noisy environment. *International Journal of Control*, 66:923–941, 1997.

[88] R.E. Kalman. Contributions to the theory of optimal control. *Boletin de la Sociedad Matematica Mexicana*, pages 102–119, 1960.

[89] H. Kawauchi, A. Chiuso, T. Katayama, and G. Picci. Comparison of two subspace identification methods for combined deterministic-stochastic systems. In *Proceedings of the 31st ISCIE International Symposium on Stochastic Systems Theory and its Applications, Yokohama, Japan*, 1999.

[90] H. Kawauchi, A. Chiuso, T. Katayama, and G. Picci. A comparison of two stochastic subspace system identification methods. Technical Report 2000-005, Department of Applied Mathematics and Physics, Kyoto University, 2000, Available online at `http://citeseer.ist.psu.edu/401889.html`.

[91] A. Krzyżak. Identification of discrete Hammerstein systems by the Fourier series regression estimate. *International Journal of Systems Science*, 20:1729–1744, 1989.

[92] S.Y. Kung. A new identification method and model reduction algorithm via singular value decomposition. In *Proceedings of the 12th Asilomar Conference on Circuits, Sytems and Comp.*, pages 705–714, 1978.

[93] P.L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(5):365–377, 2000.

[94] W.E. Larimore. Canonical variate analysis in identification, filtering and adaptive control. In *Proceedings of the 26th Conference on Decision and Control (CDC90), Hawaii, US*, pages 594–604, 1990.

[95] A. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Transactions on Automatic Control*, 24:913–921, 1979.

[96] A. Lindquist and G. Picci. Canonical correlation analysis, approximate covariance extension, and identification of stationary time series. *Automatica*, 32(5):209–233, 1996.

[97] K. Liu. Identification of Multi-Input and Multi-Output systems by observability range space extraction. *Proceedings of the 31st Conference on Decision and Control, Tucson, Arizona, USA*, pages 915–920, 1992.

[98] L. Ljung. Initialization aspects for subspace and output-error identification methods, Technical Report, presented at the European Conference on Control (ECC-03), Cambridge, England, September 2, 2003.

[99] L. Ljung. *System Identification, Theory for the User.* Prentice Hall, 1987.

[100] L. Ljung. *System Identification - Theory for the User.* Prentice-Hall, Upper Saddle River, NJ, 2nd edition, 1999.

[101] J. Marí and A. Dahlén. A covariance extension approach to identification of time series. *Automatica*, 36:379–398, 2000.

[102] J. Marí, P. Stoica, and T. McKelvey. Vector ARMA estimation: A reliable subspace approach. *IEEE Transactions on Signal Processing*, 48:2092–2104, 2000.

[103] I. Markovsky, J.C. Willems, P. Rapisarda, and B. De Moor. Algorithms for deterministic balanced subspace identification. *Automatica,* Accepted for publication, 2004.

[104] T. McKelvey and C. Hanner. On identification of Hammerstein systems using excitation with a finite number of levels. *Proceedings of the* 13^{th} *International Symposium on System Identification (SYSID2003)*, pages 57–60, 2003.

[105] R. R. Mohler. *Nonlinear systems, Volume II: Applications to Bilinear Control.* Englewood-Cliffs, New Jersey: Prentice-Hall, 1991.

[106] M. Moonen and B. De Moor. Comments on 'state-space model identification with data correlation'. *International Journal of Control*, 55(1):257–259, 1992.

[107] M. Moonen, B. De Moor, L. Vandenberghe, and J. Vandewalle. On- and off-line identification of linear state-space models. *International Journal of Control*, 49:219–232, Jan. 1989.

[108] M. Moonen, B. De Moor, and J. Vandewalle. SVD-based subspace methods for multivariable continuous time system identification. chapter in *Identification of continuous-time systems,* G.P.Rao, N.K. Sinha (Eds.), Kluwer Academic Publications, pages 473–488, 1991.

[109] M. Moonen and J. Vandewalle. A QSVD approach to on- and off-line state space identification. *International Journal of Control*, 51(5):1133–1146, 1990.

[110] K.S. Narendra and P.G. Gallman. An iterative method for the identification of nonlinear systems using the Hammerstein model. *IEEE Transactions on Automatic Control*, 11:546–550, 1966.

[111] Y. Oono. Introduction to pseudo-positive-real functions. In *Proceedings of the International Symposium on Circuits and Systems, Chicago*, pages 469–472, 1981.

[112] G. Pajunen. Adaptive control of Wiener type nonlinear systems. *Automatica*, 28:781–785, 1992.

[113] D. Pal. Balanced stochastic realization and model reduction. Master's thesis, Washington State University, Electrical Engineering, 1982.

[114] M. Pawlak. On the series expansion approach to the identification of Hammerstein systems. *IEEE Transactions on Automatic Control*, 36:736–767, 1991.

[115] B. Peeters. Stochastic subspace system identification of a steel transmitter mast. In *Proceedings of the International Modal Analysis Conference (IMAC 16), Santa Barbara, USA*, pages 130–136, 1998.

[116] K.M. Pekpe, G. Mourot, and J. Ragot. Subspace identification method using FIR modeling. In *Proceedings of the 16th IFAC world congress, Prague*, 2005, *Submitted for publication.*

[117] K. Pelckmans, M. Espinoza, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Primal-dual monotone kernel regression. Technical Report 04-108, Department of Electrical Engineering, K.U.Leuven, Leuven, Belgium, 2004, *submitted for publication*, available online at `http://www.esat.kuleuven.ac.be/sista/lssvmlab/`.

[118] K. Pelckmans, I. Goethals, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Componentwise least squares support vector machines. chapter in *Support Vector Machines: Theory and Applications,* L. Wang (Ed.), Springer, 2005, *in press.*

[119] R. Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1995.

[120] K. Peternell. *Identification of linear dynamic systems by subspace and realization-based algorithms.* PhD thesis, T.U. Wien, Vienna, 1995.

[121] K. Peternell, W. Scherrer, and M. Deistler. Statistical analysis of novel subspace identification methods. *Signal Processing*, 52:161–177, 1996.

[122] G. Picci and T. Katayama. A simple subspace identification algorithm with exogeneous inputs. In *Proceedings of the Triennial IFAC Congress, San Francisco, CA*, 1996.

[123] G. Picci and T. Katayama. Stochastic realization with exogeneous inputs and 'subspace-methods' identification. *Signal Processing*, 52:145–160, 1996.

[124] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[125] G.P. Rao and H. Garnier. Numerical illustration of the relevance of direct continuous-time model identification. In *Proceedings of the 15th International IFAC Triennal World Congress, Barcelona*, 2002.

[126] J.A. Rice. *Mathematical Statistics and Data Analysis.* Duxbury Press, Belmont, California, 2nd edition, 1975.

[127] B. Schölkopf and A. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

[128] B. Schölkopf, A. Smola, and K.R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[129] M. Scionti, J. Lanslots, I. Goethals, A. Vecchio, H. Van der Auweraer, B. Peeters, and B. De Moor. Tools to improve detection of structural changes from in-flight flutter data. In *Proceedings of the Eight International Conference on Recent Advances in Structural Dynamics (ISVR), Southampton*, Jul. 2003.

[130] J. Sjöberg, T. McKelvey, and L. Ljung. On the use of regularization in system identification. In *Proceedings of the 12th IFAC World Congress, Sydney, Australia, 7*, pages 381–386, 1993.

[131] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

[132] A. Smola, B. Schölkopf, and K.R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11:637–649, 1998.

[133] G.W. Stewart. On the perturbation of pseudo-inverses, projections and linear least squares problems. *SIAM Reviews*, 19:634–662, 1977.

[134] P. Stoica. On the convergence of an iterative algorithm used for Hammerstein system identification. *IEEE Transactions on Automatic Control*, 26:967–969, 1981.

[135] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines.* World Scientific, Singapore, 2002.

[136] J.A.K. Suykens, T. Van Gestel, J. Vandewalle, and B. De Moor. A support vector machine formulation to pca analysis and its kernel version. *IEEE Transactions on Neural Network*, 14(2):447–450, 2003.

[137] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[138] A. Swindlehurst, R. Roy, B. Ottersten, and T. Kailath. Subspace identification via weighted subspace fitting. *Proceedings of the American Control Conference*, pages 2158–2163, 1992.

[139] R.J. Vaccaro and T. Vukina. A solution to the positivity problem in the state-space approach to modeling vector-valued time series. *Journal of Economic Dynamics and Control*, 17:401–421, 1993.

[140] A. van der Sluis. Stability of the solution of linear least squares problems. *Numerical Mathematics*, 23:241–254, 1975.

[141] A. van der Sluis and G. W. Veltkamp. Restoring rank and consistency by orthogonal projection. *Linear Algebra Applications*, 28:254–278, 1979.

[142] T. Van Gestel, J. Suykens, P. Van Dooren, and B. De Moor. Identification of stable models in subspace identification by using regularization. *IEEE Transactions on Automatic Control*, 46(9):1416–1420, 2001.

[143] P. Van Overschee and B. De Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, 1993.

[144] P. Van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica,* Special Issue on Statistical Signal Processing and Control, 30(1):75–93, 1994.

[145] P. Van Overschee and B. De Moor. Choice of state-space basis in combined deterministic-stochastic subspace identification. *Automatica,* Special Issue on Trends in System Identification, 31(12):1877–1883, 1995.

[146] P. Van Overschee and B. De Moor. A unifying theorem for three subspace system identification algorithms. *Automatica,* Special Issue on Trends in System Identification, 31(12):1853–1864, 1995.

[147] P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications.* Kluwer Academic Publishers, 1996.

[148] P. Van Overschee, B. De Moor, and J. Suykens. Subspace algorithms for system identification and stochastic realization. In *Proceedings of the Conference on Mathematical Theory for Networks and Systems, MTNS, Kobe, Japan, Mita-press*, pages 589–595, 1991.

[149] T.H. van Pelt and D.S. Bernstein. Nonlinear system identification using Hammerstein and nonlinear feedback models with piecewise linear static maps - part i: Theory. *Proceedings of the American Control Conference (ACC2000)*, pages 225–229, 2000.

[150] V.N. Vapnik. *Statistical Learning Theory.* Wiley and Sons, 1998.

[151] V. Verdult, J.A.K. Suykens, J. Boets, I. Goethals, and B. De Moor. Least squares support vector machines for kernel CCA in nonlinear state-space identification. In *Proceedings of the 16th international symposium on Mathematical Theory of Networks and Systems (MTNS2004), Leuven, Belgium*, July 2004.

[152] M. Verhaegen. Subspace model identification, part III: analysis of the ordinary output-error state space model identification algorithm. *International Journal of Control*, 58:555–586, 1993.

[153] M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 30(1):61–74, 1994.

[154] M. Verhaegen and P. Dewilde. Subspace model identification, part I: the output-error state space model identification class of algorithms. *International Journal of Control*, 56:1187–1210, 1992.

[155] M. Verhaegen and P. Dewilde. Subspace model identification, part II: analysis of the elementary output-error state space model identification algorithm. *International Journal of Control*, 56:1211–1241, 1992.

[156] M. Verhaegen and D. Westwick. Identifying MIMO Hammerstein systems in the context of subspace model identification methods. *International Journal of Control*, 63:331–349, 1996.

[157] G. Wahba. *Spline models for Observational data.* SIAM, 1990.

[158] P. A. Wedin. Perturbation theory for pseudo-inverses. *BIT*, 13:217–232, 1973.

[159] D. Westwick and M. Verhaegen. Identifying MIMO Wiener systems using subspace model identification methods. *Signal Processing*, 52(2):235–258, 1996.

[160] N. Wiener. Collected works I, 1976; II, 1979; III, 1981; IV, 1984. P. Masani (Ed.), M.I.T. Press, Cambridge, MA, 1967–1984.

[161] T. Wigren. Convergence analysis of recursive identification algorithms based on the nonlinear Wiener model. *IEEE Transactions on Automatic Control*, 39:2191–2206, 1994.

[162] J.C. Willems. From time series to linear systems, part I. *Automatica*, 22(5):561–590, 1986.

[163] J.C. Willems. From time series to linear systems, part II. *Automatica*, 22(6):675–694, 1986.

[164] J.C. Willems. From time series to linear systems, part III. *Automatica*, 23(1):87–115, 1986.

[165] H. Zeiger and A. McEwen. Approximate linear realizations of given dimension via Ho's algorithm. *IEEE Transactions on Automatic Control*, 19:390–396, Apr. 1974.

[166] Y. Zhu. Estimation of an N-L-N Hammerstein-Wiener model. *Automatica*, 38:1607–1614, 2002.

Curriculum Vitae

Ivan Goethals was born June, 21 1978 in Wilrijk, Belgium.

He received his Master's degree in Physics from the Katholieke Universiteit Leuven, Belgium in July 2000. His main subject was Nuclear Physics and his Master's thesis was devoted to the 'study of the di-neutron transfer between Helium-6 and Helium-4 in an elastic scattering experiment' and was finished at the Institute of Nuclear and Radiation Physics (IKS) under the supervision of Prof. Dr. Marc Huyse.

In September 2000 he started working as a researcher at the Electrical Engineering Department ESAT in the Lab of Signals, Identification, System Theory and Automation SCD/SISTA at the Katholieke Universiteit Leuven. After first working on a KULeuven grant he became a Fellow with the Fund for Scientific Research Flanders (FWO-Vlaanderen), which offered a PhD grant from October 2001 until September 2005.

Publications by the author

Journal papers

- I. Goethals, T. Van Gestel, J.A.K. Suykens, P. Van Dooren, and B. De Moor. Identification of positive real models in subspace identification by using regularization. *IEEE Transactions on Automatic Control*, 48(10):1843–1847, 2003.

- I. Goethals, K. Pelckmans, J.A.K. Suykens, and B. De Moor. Identification of MIMO Hammerstein models using least squares support vector machines. Technical Report 04-45, ESAT-SISTA, K.U.Leuven (Leuven Belgium), *Accepted for publication in Automatica*, 2004.

- I. Goethals, K. Pelckmans, J.A.K. Suykens, and B. De Moor. Subspace identification of Hammerstein systems using least squares support vector machines. Technical Report 04-114, ESAT-SISTA, K.U.Leuven (Leuven Belgium), *Submitted for publication*, 2004.

- L. Hoegaerts, L. De Lathauwer I. Goethals, J.A.K. Suykens, J. Vandewalle, and B. De Moor. Efficiently Updating and Tracking the Dominant Kernel Principal Components. Technical Report 05-01, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), *Submitted for publication*, 2005.

- I. Goethals, L. Hoegaerts, J.A.K. Suykens, V. Verdult, B. De Moor. Hammerstein-Wiener subspace identification using kernel Canonical Correlation. Technical Report 05-30, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), *Submitted for publication*, 2005.

Book chapters

- K. Pelckmans, I. Goethals, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Componentwise least squares support vector machines. *Support Vector Machines: Theory and Applications,* L. Wang (Ed.), Springer, 2005, pp. 77-98.

Conference papers

- I. Goethals and B. De Moor. Model reduction and energy analysis as a tool to detect spurious modes. In *Proceedings of the International Conference on Noise and Vibration Engineering (ISMA), Leuven, Belgium*, Jun. 2002.
- M. Scionti, J. Lanslots, I. Goethals, A. Vecchio, H. Van der Auweraer, B. Peeters, and B. De Moor. Tools to improve detection of structural changes from in-flight flutter data. In *Proceedings of the Eigth International Conference on Recent Advances in Structural Dynamics (ISVR), Southampton*, Jul. 2003.
- I. Goethals and B. De Moor. Subspace identification combined with new mode selection techniques for modal analysis of an airplane. In *Proceedings of the 13th IFAC Symposium on System Identification (SYSID 2003), Rotterdam, The Netherlands*, pages 695–700, Sep. 2003.
- I. Goethals, T. Van Gestel, J.A.K. Suykens, P. Van Dooren, and B. De Moor. Identifying positive real models in subspace identification by using regularization. In *Proceedings of the 13th System Identification Symposium (SYSID2003), Rotterdam, The Netherlands*, pages 1411–1416, Aug. 2003.
- I. Goethals, L. Mevel, A. Benveniste, and B. De Moor. Recursive output-only subspace identification for in-flight flutter monitoring. In *Proceedings of the 22nd International Modal Analysis Conference (IMAC-XXII), Dearborn, Michigan*, Jan. 2004.
- V. Verdult, J.A.K. Suykens, J. Boets, I. Goethals, and B. De Moor. Least squares support vector machines for kernel CCA in nonlinear state-space identification. In *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTNS2004), Leuven, Belgium*, Jul. 2004.
- I. Goethals, K. Pelckmans, J.A.K. Suykens, and B. De Moor. NARX identification of Hammerstein models using least squares support vector machines. In *Proceedings of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2004), Stuttgart, Germany*, pages 507–512, Sep. 2004.
- I. Goethals, B. Vanluyten, and B. De Moor. Reliable spurious mode rejection using self learning algorithms. In *Proceedings of the International Conference on Modal Analysis Noise and Vibration Engineering (ISMA 2004), Leuven, Belgium*, pages 991–1003, Sep. 2004.
- T. Coen, N. Jans, P. Van de Ponseele, I. Goethals, J. De Baerdemaeker, and B. De Moor. Modelling the relationship between human perception and sound quality parameters using LS-SVMs. In *Proceedings of the International Conference on Modal Analysis Noise and Vibration Engineering (ISMA 2004), Leuven, Belgium*, pages 3749–3763, Sep. 2004.

- T. Coen, N. Jans, P. Van de Ponseele, I. Goethals, J. De Baerdemaeker, B. De Moor, Engine sound comfortability: relevant sound quality parameters and classification, Technical Report 04-165, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2004, *Submitted for publication.*
- T. Van Herpe, I. Goethals, B. Pluymers, F. De Smet, P. Wouters, G. Van den Berghe, and B. De Moor. Challenges in data-based patient modeling for glycemia control in ICU-patients. In *Proceedings of the IASTED International Conference on Biomedical Engineering* (IASTED - Biomed), Innsbruck, Austria, Feb. 2005, pp. 685-690.
- I. Goethals, K. Pelckmans, L. Hoegaerts, J.A.K. Suykens, B. De Moor, Subspace intersection identification of Hammerstein-Wiener systems, Technical Report 05-46, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2005, *Submitted for publication.*
- T. Coen, I. Goethals, J. Anthonis, B. De Moor, J.D. De Baerdemaeker, Modelling the propulsion system of a combine harvester, Technical Report 05-47, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2005, *Submitted for publication.*
- K. Pelckmans, I. Goethals, J.A.K. Suykens, B. De Moor, On model complexity control in identification of Hammerstein Systems, Technical Report 05-48, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2005, *Submitted for publication.*

Technical reports

- I. Goethals, L. Hoegaerts, J.A.K. Suykens, V. Verdult, and B. De Moor. Hammerstein-Wiener subspace identification using kernel Canonical Correlation Analysis. Technical Report 05-30, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2005.
- I. Goethals I., B. De Moor, First order perturbation analysis of data-driven subspace algorithms, Technical Report 05-31, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2005.
- I. Goethals I., B. De Moor, Some comments on the definition of a Total Least Squares condition number as, Technical Report 05-32, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2005.