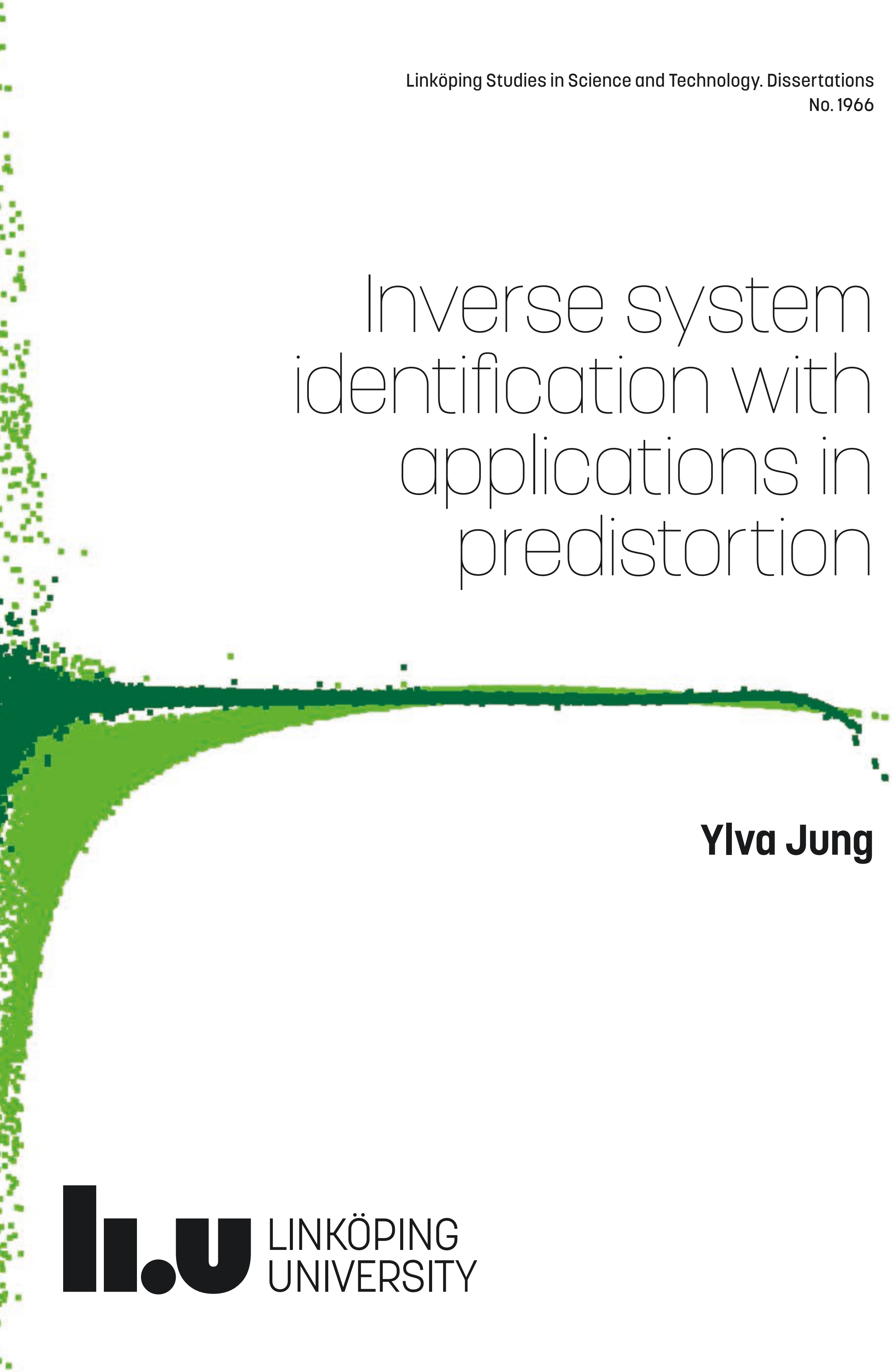
Linköping Studies in Science and Technology. Dissertations
No. 1966

Inverse system identification with applications in predistortion



Ylva Jung

LINKÖPING
UNIVERSITY

Linköping Studies in Science and Technology. Dissertations
No. 1966

Inverse system identification with applications in predistortion

Ylva Jung

LINKÖPING
UNIVERSITY

Cover illustration: Measured AM-PM characteristics (phase change vs. input amplitude) with predistortion (dark green) and without (light green).

Linköping studies in science and technology. Dissertations.
No. 1966

Inverse system identification with applications in predistortion

Ylva Jung

ylvju@isy.liu.se
www.control.isy.liu.se
Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE–581 83 Linköping
Sweden

ISBN 978-91-7685-171-5 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2018

Till Daniel, Hannes och Kerstin

Abstract

Models are commonly used to simulate events and processes, and can be constructed from measured data using system identification. The common way is to model the system from input to output, but in this thesis we want to obtain the inverse of the system.

Power amplifiers (PAs) used in communication devices can be nonlinear, and this causes interference in adjacent transmitting channels. A prefilter, called predistorter, can be used to invert the effects of the PA, such that the combination of predistorter and PA reconstructs an amplified version of the input signal. In this thesis, the predistortion problem has been investigated for outphasing power amplifiers, where the input signal is decomposed into two branches that are amplified separately by highly efficient nonlinear amplifiers and then recombined. We have formulated a model structure describing the imperfections in an outphasing PA and the matching ideal predistorter. The predistorter can be estimated from measured data in different ways. Here, the initially nonconvex optimization problem has been developed into a convex problem. The predistorters have been evaluated in measurements.

The goal with the inverse models analyzed in this thesis is to use them in cascade with the systems to reconstruct the original input. It is shown that the problems of identifying a model of a preinverse and a postinverse are fundamentally different. It turns out that the true inverse is not necessarily the best one when noise is present, and that other models and structures can lead to better inversion results.

To construct a predistorter (for a PA, for example), a model of the inverse is used, and different methods can be used for the estimation. One common method is to estimate a postinverse, and then using it as a preinverse, making it straightforward to try out different model structures. Another is to construct a model of the system and then use it to estimate a preinverse in a second step. This method identifies the inverse in the setup it will be used, but leads to a complicated optimization problem. A third option is to model the forward system and then invert it. This method can be understood using standard identification theory in contrast to the ones above, but the model is tuned for the forward system, not the inverse. Models obtained using the various methods capture different properties of the system, and a more detailed analysis of the methods is presented for linear time-invariant systems and linear approximations of block-oriented systems. The theory is also illustrated in examples.

When a preinverse is used, the input to the system will be changed, and typically the input data will be different than the original input. This is why the estimation for preinverses is more complicated than for postinverses, and one set of experimental data is not enough. Here, we have shown that identifying a preinverse in series with the system in repeated experiments can improve the inversion performance.

Populärvetenskaplig sammanfattning

Tänk dig att du är på plats A och vill ta dig till plats B. Du frågar tre olika personer om vägen, och får tre olika svar. Den första pekar dig i rätt riktning och förklarar att det är skyltat, och bara att följa skyltningen. Den andra berättar vilka gator och vägar du ska köra för att komma fram. Den tredje ger dig en karta. Alla tre sätten gör att du kommer fram till plats B utan problem. Sedan vill du åka tillbaka – är alla vägbeskrivningar lika bra nu?

Matematiska beskrivningar, kallade modeller, används i många tekniska tillämpningar. Ett exempel är utveckling av bilar, där man med simuleringar kan utvärdera olika designval på ett kostnadseffektivt sätt. Ett annat är flygtillämpningar där riktiga tester på flygplanet skulle kunna leda till fara för piloten. Modellerna kan skattas med hjälp av uppmätt data från systemet, vilket kallas systemidentifiering. Ett system är den avgränsade del av världen som vi är intresserade av, i exemplen ovan bilen och flygplanet. I systemidentifiering är målet att finna en matematisk modell som så bra som möjligt beskriver systemets beteende.

I denna avhandling undersöks hur inversa modeller kan skattas. Här menas med invers att vi bildligt sett ska gå baklänges genom systemet. I bilen är gaspådraget något vi kan påverka, och beroende på många olika faktorer (såsom växel, lutning på vägbanan och vind) så kommer detta att resultera i att bilen får en viss hastighet. Om vi istället vill ha inversen, skulle man kunna utgå från att vi vill ligga i 70 km/h, och därifrån beräkna vilket gaspådrag som behövs. I vägbeskrivningsexemplet är inversen en mer bokstavlig tolkning, där vi faktiskt vill åka tillbaka längs samma väg. Det är tydligt att en bra modell/beskrivning hänger ihop med hur den ska användas.

Skattning av inversa system kan göras på flera sätt. Inversen kan exempelvis baseras på en modell av systemet som sedan inverteras, eller skattas direkt som en invers. Hur inversen skattas påverkar modellen genom att olika egenskaper hos systemet fångas, och detta kan därför ha en stor inverkan på slutresultatet. De olika metoderna analyseras i avhandlingens första del. Även ordningen på systemet och inversen spelar roll för hur lätt det är att hitta en invers. Det visar sig vara mer rättframt att skatta en invers som ska användas efter systemet än då inversen skall ligga före systemet, som en förinvers.

Linjärisering av effektförstärkare är ett exempel där inversa modeller utnyttjas. Effektförstärkare används i många tillämpningar, bland annat mobiltelefoni, och deras uppgift är att förstärka en signal vilket är ett steg i överföringen av information. I exemplet med mobiltelefoner kan det exempelvis vara en persons röst som är signalen, vilken ska överföras från telefonen via luften och vidare till mottagaren. Om effektförstärkaren inte är perfekt kan detta medföra att den sprider effekt till närliggande frekvensband. För den som ska använda dessa frekvensband uppfattas detta som en störning, och det finns därför gränser för hur mycket spridning som får ske. För att uppfylla dessa krav på spridning krävs att man förändrar signalen på något sätt. Genom att modellera vad som händer i förstärkaren och invertera detta kan man få ett system som inte sprider effekt i angränsande frekvensband. I detta sammanhang säger man att en förkompensering, även kallad fördistorsion, används.

I outphasing-förstärkare, som har en olinjär effektförstärkarstruktur, delas signalen upp i två delar och varje del förstärks separat för att sedan adderas. Fördelen med denna uppdelning är att dessa effektförstärkare kan göras väldigt effektsnåla, vilket direkt speglas i exempelvis batteritiden för en mobiltelefon. Om denna uppdelning och addition inte är perfekt uppstår olinjäriteter, och fördistorsion krävs. I avhandlingen presenteras flera olika metoder för att ta fram fördistorsion för outphasingförstärkare. En första metod baseras på en ny modellstruktur som fångar förstärkarens beteende väl och sedan kan användas för fördistorsion. Denna metod är dålig ur beräkningssynpunkt och har därför vidareutvecklats, och vi visar hur de nya metoderna baseras på en teoretiskt ideal förinvers. Metoderna har utvärderats på fysiska förstärkare, och resultaten visar att en förbättring uppnås vid användning av fördistorsion.

Acknowledgments

Life is not easy. Finishing a PhD is not easy. One of the good things about the PhD is that you get a chance at the end to say thank you to the people who have helped you, which you might not get in life. This is my attempt to thank the ones who have helped me go through with this project. I'll start and end these acknowledgments with people without whom I think there would be no thesis (or not this, *my* thesis at least, others might still be able to write theirs).

To my supervisor Dr. Martin Enqvist: I think I would have given up a long time ago if it wasn't for your encouragements. I am really impressed with how you always have time (or rather, take the time) to answer questions or concerns, and without feeling stressed. I know you have lots of other things to do and I really appreciate it. I also like how you can turn things around for a positive spin. Thank you for pushing me through this!

Prof. Lennart Ljung and Prof. Torkel Glad, thanks for being co-supervisors and providing input. I should have talked, asked and learned more!

I have had the pleasure of having three different bosses, who have all helped me sort out the work situation when life gets in the way. Prof. Lennart Ljung, Prof. Svante Gunnarsson and Dr. Martin Enqvist, you all seem to be able to guide the group forward as well as see the people and how to help. Lennart, thanks for letting me join the group! Svante, without your flexibility and willingness to adapt I could not have finished this thesis. Thank you! Martin, thanks for making every day *bring-your-baby-to-work*! I am also grateful for the administrative help from Ninna Stensgård and her predecessor Åsa Karmelind.

The upside of being on the slow side finishing your PhD is that you get to meet a lot of colleagues. I have really appreciated the Automatic control group and the amazing people in it! It is a group full of brilliant, fun, and hard working people, thank you all! I think my current office mate Kerstin is my favorite, but Patrik Leissner, Maryam Sadeghi Reineh and Gustaf Hendeby, you tie for second place (and you were all far less demanding)! Many colleagues I also consider friends and hope I will still see a lot of you in the future. Special shout outs to Manon Kok (for staying in touch and being a good friend), Daniel Petersson (for pepp and believing in me), Patrik Leissner (for friendship and helpfulness), Michael Roth, Daniel Simon, Sina Khoshfetrat Pakazad, Johan Dahlin, Clas Veibäck, Zoran Sjanic, Niklas Wahlström, Martin Skoglund, Jonas Linder, Gustaf Hendeby, Gustav Lindmark, Erik Hedberg, Christian Lyzell, Roger Larsson, Jonatan Olofsson, Per Boström-Rost, Kristoffer Bergman, Christian Andersson Naesseth and Magnus Malmström. Thanks for the nice breaks in the fika room with discussions on every topic, lunch walks, bike excursions, conferences and beer nights.

Writing this thesis was made easier with the excellent thesis template by LaTeX gurus Dr. Gustaf Hendeby and Dr. Henrik Tidefelt, and all my questions and troubles were solved by Dr. Daniel Petersson and (again) Gustaf. Thanks a lot! I would also like to say thank you to my lovely proof readers, who have provided constructive comments and improved the thesis. Dr. Daniel Jung, Dr. Patrik Leissner, Dr. Daniel Petersson, Lic. Roger Larsson, M.Sc. Angela Fontan, M.Sc. Magnus Malmström, and my supervisors Martin and Lennart, thank you for your time!

Dr. Jonas Fritzin and Prof. Atila Alvandpour brought me into this field of research, and I appreciate the nice cooperation and collaboration.

I also got a chance to spend time at the Center for Automotive Research (CAR) at The Ohio State University (OSU) in Columbus, Ohio, and would like to express my gratitude to Prof. Giorgio Rizzoni for welcoming me to the group and lending me a desk. And to the group, Jen, Matilde, Greg, Qadeer, Bharat & Raj, Simon, Avi, Anna, Alex, Leo, Ruochen, Pradeep, Nancy, Adithiya, Meg, John, Shreshta, Marcello. Thanks for welcoming us, we miss you guys! (I know Hannes was the main reason, but you hid it well ;))

This work has been supported by the Excellence Center at Linköping-Lund in Information Technology (ELLIIT), the Center for Industrial Information Technology at Linköping University (CENIIT) and the Swedish Research Council (VR) Linneaus Center CADICS, which is gratefully acknowledged.

As big a part of life as work is, the outside life is so important to keep me up. Thanks for taking my thoughts off of work and for talking about everything else! I am so happy and blessed to have you in my life! To my parents Karin and Einar: Thanks for all the help and for swooshing down to the rescue! Tora, Magnus, Veronica, Ruben & Sofia: Thanks for being there! Hanna: Thanks for support and commiseration! Johanna, Andreas, Linda, Charlotte, Fredrik, Linda, Emma, Henrik: Nice timing! I am glad to have you around during the småbarnsåren and hope the tiny humans won't keep us too busy to meet up more! Kristofer, Claire, Hedvig, Gustaf, Frida, Sofie, Anders: Thanks for not forgetting us and keeping us a part of the outside world! Michaela & Erik: Stay! Annika & Kenny: For being amazing! Malin & Louise: Who knew a pelvic girdle pain class could have such a happy bonus?! Janna, Danne, Daniel, Malin, Karolina, Ulrika, Dan, Sofia, Martin: For old times! Maria, Anna, Petra: For welcoming me back home!

Daniel. I sortof blame you for getting me started on a PhD, but your encouragements are also one of the reasons that made me go through with it to the end. There is no one else I would rather go through life's ups and downs with! I look forward to being a happier and more positive version of myself soon (although probably still as tired). Hannes and Kerstin, you are the biggest distraction from work and the best! Jag är så glad att ni är mina! Jag älskar er!

Linköping, November 2018

Ylva Jung

Contents

I Estimation of inverse systems

II Power amplifier predistortion

Notation

Outphasing Amplifiers

Notation	Meaning
$\Delta_\psi(s_1, s_2)$	$\arg(s_1) - \arg(s_2)$, angle difference between outphasing signals, defined on page 129
Δ_ψ	same as $\Delta_\psi(s_1, s_2)$
$\Delta_\psi(s_{1,P}, s_{2,P})$	angle difference between predistorted outphasing input signals
$\Delta_\psi(y_{1,P}, y_{2,P})$	angle difference between predistorted outphasing output signals
ξ_k	angle difference between $\tilde{s}_k$ and s_k, defined in (9.6)-(9.7), page 131, and Figure 9.1, page 130
f_k	phase distortion in the amplifier branch k, defined in (9.9)
g_1, g_2	gain factors of each branch in PA, should ideally be $g_1 = g_2 = g_0$
h_k	phase predistorter functions in the amplifier branch k, defined in (10.1)
s_k	outphasing input signals, decomposed in standard way (8.11)
$s_{k,P}$	predistorted outphasing input signal in branch k, decomposed with identical gain factors using (8.11)
$\tilde{s}_k$	outphasing input signal in branch k, decomposed with nonidentical gain factors using (9.3)
y_k	outphasing output signal in branch k, decomposed with nonidentical gain factors using (9.3)
$y_{k,P}$	predistorted outphasing output signal in branch k, decomposed with nonidentical gain factors using (9.3)
$\hat{x}$	an estimate of the value of x

Power Amplifier Glossary

Notation	Definition
ACLR, ACPR	adjacent channel leakage (power) ratio, a linearity measure that describes the amount of power spread to neighboring channels, page 114.
AM-AM, AM-PM	amplitude modulation to amplitude modulation or phase modulation, respectively, a plot mapping the output amplitude (or phase distortion) to the input amplitude to determine the distortion induced by the circuit, for example a power amplifier, page 114.
combiner	the circuit that handles the addition of signals in, for example, Figure 8.13, page 121.
dBc	decibel to carrier, the power ratio of a signal to a carrier signal, expressed in decibels.
dBm	power level expressed in dB referenced to one milliwatt, so that zero dBm equals one mW and one dBm is one decibel greater (about 1.259 mW).
DE, PAE	drain efficiency and power added efficiency are efficiency measures for power amplifiers, page 113.
DLA, ILA	direct and indirect learning architectures are two approaches to estimate a power amplifier predistorter, see METHOD B and METHOD C on page 58.
DPD	digital predistortion, a linearization technique for power amplifiers that modifies the input to counteract power amplifier distortion from nonlinearities and dynamics, page 123.
DR	dynamic range, defining the ratio of the maximum and minimum output amplitudes an amplifier can achieve, page 122.
IQ	a signal separation into an imaginary part (quadrature, Q) vs real part (in-phase, I), page 110.
LO	local oscillator, a circuit that produces a continuous sine wave. Usually drives a mixer in a transmitter/receiver, page 110.
mixer	translates the signal up or down to another frequency, page 110 and Figure 8.2.
outphasing, LINC	an outphasing amplifier, also called linear amplification with nonlinear components, is a nonlinear amplifier structure.
PA	power amplifier, used to increase the power of a signal, so that the output is a magnified replica of the input.
RF	radio frequency, ranging between 3 kHz and 300 GHz.
SCS	signal component separator, (here) decomposes the signal into outphasing signals according to (8.11).

Abbreviations A-O

Abbreviation	Meaning
AC	Alternating current
ACLR	Adjacent channel leakage ratio
ACPR	Adjacent channel power ratio
AM	Amplitude modulation
AM-AM	Amplitude modulation to amplitude modulation
AM-PM	Amplitude modulation to phase modulation
BJT	Bipolar junction transistor
CMOS	Complementary metal-oxide-semiconductor
DAC	Digital-to-analog converter
DB	Digital baseband
DC	Direct current
DE	Drain efficiency
DLA	Direct learning architecture
DPD	Digital predistortion or predistorter
DR	Dynamic range
EDGE	Enhanced data rates for GSM evolution
EVM	Error vector magnitude
FPGA	Field programmable gate array
FET	Field-effect transistor
FIR	Finite impulse response
FM	Frequency modulation
GSM	Global system for mobile communications
GPRS	General packet radio service
IIR	Infinite impulse response
ILA	Indirect learning architecture
ILC	Iterative learning control
IQ	in-phase component (I, real part) vs quadrature component (Q, imaginary part)
IV	Instrumental variables
LINC	Linear amplification with nonlinear components
LMS	Least mean squares
LO	Local oscillator
LS	Least squares
LTE	Long term evolution
LTI	Linear time invariant
LUT	Look-up table
MIMO	Multiple-input multiple-output
MOSFET	Metal-oxide-semiconductor field-effect transistor
MSE	Mean square error
NMOS	N-channel metal-oxide-semiconductor

Abbreviations P-Z

Abbreviation	Meaning
PA	Power amplifier
PAE	Power added efficiency
PAPR	Peak-to-average power ratio
PD	Predistortion or predistorter
PEM	Prediction-error (identification) method
PM	Phase modulation
PMOS	P-channel metal-oxide-semiconductor
PVT	Process, voltage and temperature
PWM	Pulse-width modulated
RBW	Resolution bandwidth
RF	Radio frequency
RLS	Recursive least squares
RMS	Root mean square
RX	Receiver
SCS	Signal component separator
SISO	Single-input single-output
SLS	Separable least-squares
TX	Transmitter
WCDMA	Wideband code-division multiple access

1 Introduction

Modeling of inverse systems might seem like a very narrow field of research, because when would you really need it? The answer is *Quite often actually!*

Inverse systems and models thereof show up in numerous applications, more or less visibly. This results in a need for methods to estimate the models and evaluate the performance. The concept of building models based on measured data is called system identification, and there are many theoretical results concerning the properties of the estimated models. However, when the goal is to estimate an inverse model, less work has been done. There are different options to estimate such an inverse model, and the resulting model and its properties will be impacted by the choice.

In this chapter, a short research motivation will be given, followed by an outline of the thesis. Then follows an overview of the contributions of the thesis, and some clarifications of the author's role in the work.

1.1 Research motivation

Power amplifiers (PAs) are often used in communication devices, such as mobile phones and base stations. In a hand-held device (such as a mobile phone), the power efficiency is an important property as it will reflect directly on the battery time. Higher demands on high efficiency has pushed the development towards nonlinear devices, which are more power efficient, but also introduce new problems. A nonlinear device will not only transmit power in the frequency band where the input signal is, but also risks spreading power to neighboring transmitting channels. For anyone transmitting in these frequency bands, this will be perceived as noise. Therefore, there are standards describing the amount of power that is allowed to be spread to adjacent frequencies. This nonlinear spreading of energy can be reduced by linearization of the power amplifier, limiting the interference in the neighboring channels. Since it is preferable to work with the non-amplified signal, this is often done by adding an extra prefilter in series with the amplifier. This block is called a *predistorter*. More on power amplifier predistortion can be found in the second part of this thesis.

The problem in loudspeaker linearization is similar to that of power amplifier predistortion. Classical loudspeakers are large to allow for a large movement of the cone, to be able to produce sound of different frequencies. Today, there is a large demand for smaller loudspeakers, both for aesthetic reasons (they should not be visible and big as old loudspeakers) and a demand for better loudspeakers in smartphones, tablets and laptops. Small loudspeakers, in mobile phones for example, can show a nonlinear behavior due to limitations in the movement of the cone. This will distort the sound and make listening to music less agreeable [Björk and Wilhelmsson, 2014]. Cheaper material and components in combination with a smaller size make it harder to produce sound in the whole audible frequency range. The goal here is to create a better sound using digital signal processing, to reduce the effects of the nonlinearities introduced by the smaller size. For this application, the output is air pressure in the form of sound waves and once the signal has been converted to sound, it cannot be altered. It is of course possible to use microphones in the tuning of this linearizing block, but having a setup using a feedback loop with a microphone in daily use is not a desirable option. Hence, the need for a preinverter is clear.

The need for calibration is also relevant in other applications, for example sensors. One type of sensor is the *analog-to-digital converter* (ADC) where an analog (continuous) input signal is converted to a digital output, which is limited to a number of discrete values. A small error in the analog input risks causing a larger error in the output, since the discrete signal is limited to certain values. There are different implementation techniques for the ADC, and similarly to the PAs, the demand for higher speed has inspired new techniques. This has reduced the linearity, and increased the need for linearization. Also other types of sensors can be dynamic or nonlinear which will distort the measurement, and if we know how, we can obtain a better estimate of the original (measured) signal. Feedback in this setup would perturb the original signal we want to measure. So for sensor calibration, a postinverse is desired.

Inversion of systems also appear in other areas, not directly connected to pre- or postinversion. One application where models of both the system $\mathcal{S}$ and its inverse $\mathcal{S}^{-1}$ are used is robotics. The forward kinematics, describing how to compute the robot tool pose as a function of the joint variables, are used for control as well as the inverse kinematics, how to compute joint configuration from a given tool pose. In feedforward control, a common choice for the controller is a modification of the plant inverse (where the modification could be a softening of the behavior). The idea with feedforward control is that the feedforward controller should counteract the future effects of the plant, and it is often combined with feedback control to be able to handle model errors and disturbances. See for example Boeren et al. [2014] where feedback and feedforward control are combined with input shaping.

The same idea can be used when you want an internal signal that is impossi-

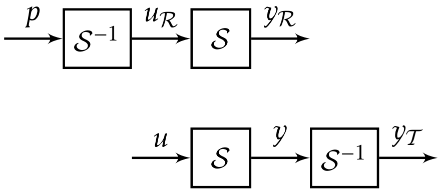


Figure 1.1: *An inverse $\mathcal{S}^{-1}$ is used to undo the effects of the system $\mathcal{S}$. The top figure shows a preinverse, where the inverse $\mathcal{S}^{-1}$ is applied before the system $\mathcal{S}$, and the bottom figure shows a postinverse where the system is followed by an inverse. For a preinverse, the preinverted signal $u_{\mathcal{R}}$ should make the output $y_{\mathcal{R}}$ the same as the reference p, $y_{\mathcal{R}} = p$. For a postinverse, the output $y_{\mathcal{T}}$ should be altered to be the same as the input u, $y_{\mathcal{T}} = u$.*

ble or hard to obtain or measure. This could be medical applications where some substances are hard to measure, and the concentration of one substance will tell you about the value of another. Kawato et al. [1987] explains voluntary movements in the brain with a full feedback loop and then model the inverse dynamics to reduce the time and the need for a longer feedback loop. The similiarities to robotics has been explored in Tavan et al. [2011]. Difficulties to measure the real value of course also occur in other areas, for example in the process industry where sensors in very harsh environments such as hot or acid places can be hard to use. The sensor needs to be placed somewhere else, and the better the connection (forwards or backwards) is modeled, the better. One way to improve the estimation of ship dynamics, where a lot of input signals/disturbances are unknown (such as wind and water conditions) is to use alternative measurements, which also includes using inverse systems [Linder and Enqvist, 2017].

In all of the above applications, the question is how to find an inverse $\mathcal{S}^{-1}$ to the system $\mathcal{S}$. The application will determine if it is a *pre*inverse or a *post*inverse that is desired. In Figure 1.1, the two different utilizations are illustrated.

One common way to find or construct a model is through model estimation using data. This opens up for questions regarding this inverse estimation. Different methods can be applied. For example it can be based on an inverted model of the system itself, or the method can estimate the inverse directly. That the choice of estimation method matters is motivated by Example 1.1. Example 1.2 illustrates that a model that is good for a forward purpose is not necessarily useful in the inverse case.

Example 1.1: Introductory example

Consider a *linear time-invariant* (LTI) system. The goal is to reconstruct the input by modifying the input signal. When the structure of the inverse is set, in this case to a *finite impulse response* (FIR) system, what is the best way to estimate it? Should the inverse be estimated directly or should an inverted model of the system itself be used? These two approaches have been applied to noise-free data, and the results are presented in Figure 1.2. We see here that the two models, both descriptions of the system inverse, capture very different aspects of the system,

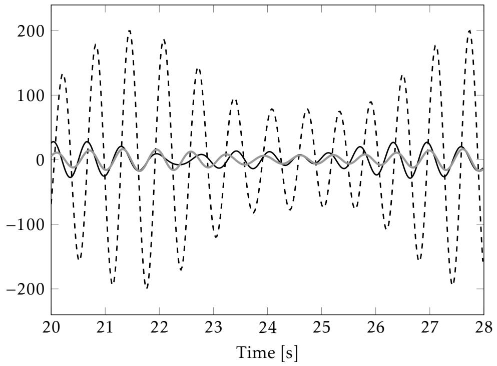


Figure 1.2: *The input u (black solid line), and the reconstructed input $y_{\mathcal{R}}$ using an inverted estimated forward model (black dashed line) and the inverse model estimated directly (gray solid line) in Example 1.1. The estimation of the inverse (gray) cannot perfectly reconstruct the input (black solid), but is clearly better than the inverted forward model (dashed).*

and that the method chosen can have a large impact. This example is described in more detail in Example 6.1, page 78.

Example 1.2: Driving instructions

You are at the Town square in Granville, Ohio. You have tickets to see the Buckeyes play football at the Shoe (The Ohio Stadium) at The Ohio State University in Columbus, Ohio. You ask someone for directions and you get one of the following driving instructions.

1. Take S Main Street, then follow signs for Columbus/OSU/Ohio Stadium…
2. Take S Main Street south, turn west on OH-16 W/OH-37 W and continue on OH/37 W for 21 miles …
3. They give you a map.

Either option will get you to the game in time. Now if you want to go back – which one would you prefer?

The first one gives you the information needed, but nothing more. The second one can be made more or less explicit (the distance traveled on each road, the exit number, etc.) and will then be easier or harder to follow/invert on the way back. The third one, the map, is the most complex, but you can use it in any situation (going back, traveling to a different location, …) A map of the area with a suggested itinerary is presented in Figure 1.3.

It is thus clear that a forward model that works great in that setting might not be optimal once you try to go backwards.

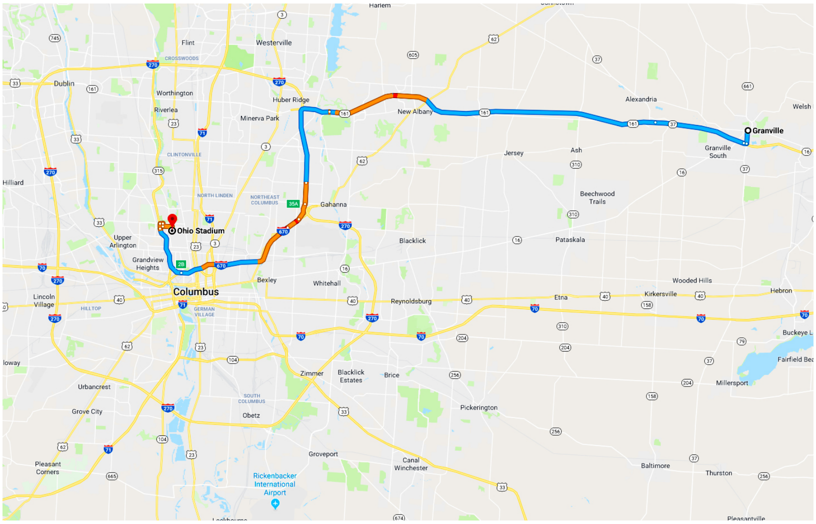


Figure 1.3: A map of a suggested itinerary from Granville Town square to the Shoe at The Ohio State University, Columbus, OH, USA, as discussed in Example 1.2. Map data ©2018 Google.

1.2 Outline

The thesis is divided into two parts. The first introduces system inversion and the estimation of inverse models. The second part concerns using estimated inverse models for power amplifier predistortion. Outside of the two parts of the thesis containing mainly new results are this chapter (with problem formulation and contributions) and Chapter 2 with a short introduction to model estimation using system identification.

Part I – Estimation of inverse systems contains results on the identification of inverse systems. Chapter 3 presents methods of inversion, used throughout literature. In Chapter 4, the inversion is expanded to include stochasticity such as noise. It also includes notation, a discussion on optimality and some examples. The identification of these inverse models is discussed in Chapter 5 along with method descriptions and analysis of some special cases: linear, time-invariant systems and block-oriented systems. The discussion is followed by examples in Chapters 6 and 7. In Chapter 6 model approximations in a noise-free setting are presented and in Chapter 7 a small case study with process noise and measurement noise is evaluated using different identification methods. Chapter 7 also contains an example of identification of a Hirschorn preinverse and a discussion on inverse system identification, concluding the first part of the thesis.

In *Part II – Power amplifier predistortion*, the estimation of inverse models is applied to outphasing power amplifiers. Here, the goal is to find an inverse such that the output of the power amplifier is an amplified replica of the input, counteracting the distortion caused by the amplifier. An introduction to power amplifier functionality and characterization is given in Chapter 8 as well as an overview of common predistortion methods. This chapter also contains a description of the outphasing power amplifier, which is a nonlinear amplifier structure that needs predistortion, and for which the predistorter methods in this thesis were produced. Modeling approaches for the power amplifier are presented in Chapter 9 and methods for finding a predistorter in Chapter 10. The predistortion methods are evaluated in measurements on real power amplifiers in Chapter 11.

The thesis is concluded by Chapter 12 where some conclusions and a discussion on ideas for future research are presented. Some additional information about the power amplifiers used is given in the appendix.

1.3 Contributions

The contributions in this thesis are in two areas; model estimation of inverse systems and the application thereof in power amplifier predistortion. The main contributions are highlighted here.

1. A formulation of the estimation goals for preinversion and postinversion and an explanation why they are principally different. When there is noise present, the true inverse will not be the best inverse, since the noise contributions need to be taken into consideration.

2. A classification of different identification methods for inverse system identification and the description of an iterative method that uses the system during repeated experiments to construct a preinverse.

3. The analysis of inverse identification methods for the special cases linear time-invariant systems and block-oriented systems.

4. A model structure that can describe both an outphasing power amplifier and a predistorter, and that only changes the phases of the outphasing signals.

5. The description of an ideal predistorter for outphasing amplifiers and different convex approaches to obtain an approximation of the predistorter based on measured data.

The contributions are further discussed below.

Inverse system identification Estimation of inverse models (inverse system identification), treats the problem of finding a good model when the end goal is to use not a model of the system itself, but the inverse. The inverse could be used as a preinverse or a postinverse. The first contribution is the formulation of the estimation goals for preinversion and postinversion in Chapter 4 and showing that they are principally different. For the estimation of the postinverse, measured data of input and output are sufficient to find the optimal inverse. For the estimation of a preinverse in a general setting, this is no longer the case and multiple measurements are needed, since the preinverse will change the input signal to the system. In system identification, it is common to use the idea of a *true system*, which is assumed to have produced the data, and when the data is noise-free and the system is invertible, the true inverse will be the best pre- or postinverse. However, when there is noise present, we have shown that the true inverse will no longer be the best inverse, since the noise contributions need to be taken into consideration. This is valid for a preinverse and a postinverse.

There are multiple ways to estimate a preinverse or postinverse. Inverse models can, for example, be estimated directly as a preinverse or postinverse, or based on a model of the forward system. The second contribution is the different approaches, discussed in Chapter 5, along with a classification of the different model estimation approaches used in literature. In this thesis we investigate the different methods to improve the knowledge of the inverse estimation methods.

The third contribution is the analysis of the two special cases *linear time-invariant* (LTI) systems and linear approximations of block-oriented systems. Inverse modeling of LTI systems was presented at the 52nd IEEE Conference on Decision and Control (CDC) in

> Ylva Jung and Martin Enqvist. Estimating models of inverse systems. In *52nd IEEE Conference on Decision and Control (CDC)*, pages 7143–7148, Florence, Italy, December 2013. ©2013 IEEE

For linear systems, the frequency weighting of the identified models differ depending on whether the inverse is based on a forward model or an inverse is estimated directly, and the models capture quite different properties of the system. The theory is presented in Chapter 5 and an example in Chapter 6. In this paper a postinverse application of Hirschorn's method presented in Section 3.3.2 is also shown.

A common way to represent nonlinear systems is to use block-oriented systems, consisting of static nonlinear blocks and linear dynamic blocks. The modeling of this type of systems is a well-explored field, but the inverse estimation has not been done before, to the authors' knowledge. The results were presented at the 17th IFAC Symposium on System Identification (SYSID) in

> Ylva Jung and Martin Enqvist. On estimation of approximate inverse models of block-oriented systems. In *17th IFAC Symposium on System Identification (SYSID)*, pages 1226–1231, Beijing, China, October 2015

describing the estimation of approximate inverse models of block-oriented systems. For a Hammerstein system with a white input signal, estimating a forward model and inverting it will result in the same model as if the inverse is estimated directly. For a colored input or a Wiener system, this is not true. The theory is presented in Chapter 5 and examples in Chapter 6.

As mentioned above, multiple measurements should be used for the estimation of a preinverse. A modification of the predistortion estimation method *direct learning architecture* (DLA) is formulated in which the modeling is performed with a predistorter present in the measurement collection. In the standard DLA, the system is replaced by a model thereof. Since the preinverse will change the characteristics of the input signal to the system and the system contains noise, repeated measurements should be used. This expansion is denoted METHOD B2. A simple implementation of the iterative method is shown to improve the preinversion results. METHOD B2 is presented in Chapter 5 and evaluated in simulations in Chapter 7.

Power amplifier predistortion A preinverse, which is also called a predistorter, can be used to counteract the imperfections of a power amplifier. The outphasing power amplifier predistortion described in Chapter 10 was first presented in

> Jonas Fritzin, Ylva Jung, Per N. Landin, Peter Händel, Martin Enqvist, and Atila Alvandpour. Phase predistortion of a Class-D outphasing RF amplifier in 90nm CMOS. *IEEE Transactions on Circuits and Systems-II: Express Briefs*, 58(10):642–646, October 2011a ©2011 IEEE

where a novel model structure for outphasing power amplifiers was used. The contribution here is the model structure that works for the PA and a predistorter that changes only the phases of the outphasing signals and was shown to successfully reduce the distortion introduced by the power amplifier. Measurements and evaluation are presented in Chapter 11. The proposed model and predistorter structures were produced in close collaboration between the paper's first three authors. The theoretical motivation of the predistorter model has been developed by the author of this thesis.

The nonconvex predistortion method presented in the publication above was developed into a method that explores the structure of the outphasing power amplifier, which is also discussed in Chapters 10 and 11. It basically consists of solving least-squares problems, which are convex, and performing an analytical inversion, and it is suitable for online implementation. This is presented in

> Ylva Jung, Jonas Fritzin, Martin Enqvist, and Atila Alvandpour. Least-squares phase predistortion of a +30dbm Class-D outphasing RF PA in 65nm CMOS. *IEEE Transactions on Circuits and Systems-I: Regular papers*, 60(7):1915–1928, July 2013. ©2013 IEEE

The derivation of this least-squares predistortion method has mainly been done by the author of this thesis, whereas the paper's second author has been responsible for the power amplifier and hardware issues. In addition to the reformulation of the nonconvex problem, the paper provides a theoretical description of an ideal outphasing predistorter, that is, one that does not change neither the amplitude nor the phase of the output. This involves a mathematical description of the branch decomposition and the impact of unbalanced amplification in the two branches. This is described in more detail in Chapter 10 with measurement results in Chapter 11. The fifth large contribution in this thesis is the description of the ideal predistorter and the different approaches to obtain an approximation of it based on measured data.

The contents of Appendix A are included here for the sake of completeness and are not part of the contributions of this thesis. The power amplifiers and the characterization thereof were done at the Division of Electronic Devices, Department of Electrical Engineering at Linköping University, Linköping, Sweden, by Jonas Fritzin, Christer Svensson and Atila Alvandpour.

The author was also involved in other publications, unrelated to the main research interests. The work in

> André Carvalho Bittencourt, Patrik Axelsson, Ylva Jung, and Torgny Brogårdh. Modeling and identification of wear in a robot joint under temperature uncertainties. In *18th IFAC World Congress*, pages 10293–10299, Milan, Italy, August 2011

is based on a project work carried out jointly by the first three authors. The first author came up with the idea, related to his research, and has continued to work with the results. Discussions regarding teaching aspects of an applied control systems course are presented in

> Svante Gunnarsson, Ylva Jung, Clas Veibäck, and Torkel Glad. Io (implement and operate) first in an automatic control context. In *12th International CDIO Conference*, pages 238–249, Turku, Finland, June 2016

where the author has contributed through discussions during development and teaching of the course.

2

Introduction to system identification

In many cases it is costly, tedious or dangerous to perform real experiments on a physical phenomenon, but we still want to extract information somehow about its behavior. The limited part of the world that we are interested in is called a *system*. This system can be pretty much anything. It can for example be interesting for a car manufacturer to know how the car will react to a change in the accelerator, depending on different design choices in the engine. Or in a paper mill, how the moist content of the wood will affect the quality of the paper. For a diabetic it is essential to know how the blood sugar (glucose) level depends on food intake, exercise and insulin doses. A pilot needs to know how an airplane reacts to the control of different rudders, and in economics it is necessary to know how a change in the interest rate will influence the customers' willingness to borrow or save money. What we see as a system depends on the application. In the car analogy, the system can be only the engine, or the whole car. For the blood sugar levels we can for example be interested only in how food intake affects the blood glucose, or how exercise contributes.

In many of these applications one does not want to perform experiments directly, but instead start the evaluation using simulations. This leads to a need for *models* of the systems. One way is to use *physical modeling* where the models are based on what we know of the system by using the knowledge of, for example, the forces, moments, flows, etc. In the engine example, it is possible to calculate the output and the connection between the accelerator and the engine torque. Another modeling approach is to gather data from the system and construct a model based on this information. This approach is called *system identification* and will be presented in this chapter.

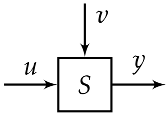


Figure 2.1: A system S with input u, output y, and disturbance v. For the blood glucose example, the system S is the patient, or rather a part of the body's metabolism system, the input u could represent food intake, the output y is the measured blood glucose level and the disturbance v is for example an infection that affects the body's insulin sensitivity.

2.1 System identification

System identification deals with the problem of identifying properties of a system. More specifically, it treats the problem of using measured data to extract a mathematical model of a system we are interested in. The introduction and notation presented here is based on Ljung [1999], but other standard references include Pintelon and Schoukens [2012] and Söderström and Stoica [1989]. Since we are dealing with sampled data, t will be used to denote the time index. Also, for notational convenience, the sample time T_s will be assumed to be one time unit, so that $y(tT_s) \triangleq y(t)$ and $y((t+1)T_s) \triangleq y(t+1)$ is the measurement after $y(t)$, but this can of course easily be adapted to other choices of T_s.

The observable signals that we are interested in are called *outputs*, denoted $y(t)$, and in the examples above this can be the car speed/engine velocity, or the glucose level in the blood for a diabetic. The system can also be affected by different sources that we are in control of – the accelerator or the food intake – called *inputs*, $u(t)$. Other external sources of stimuli that we cannot control or manipulate are called *disturbances*, $v(t)$, – such as a steep uphill affecting the car or a fever or infection which affect the insulin sensitivity. Some disturbances are measurable and for others the effects can be noted, but the signal itself cannot be measured. The different concepts are illustrated in a block-diagram in Figure 2.1.

A system has a number of properties connected to it. A system is *linear* if its output response to a linear combination of inputs is the same linear combination of the output responses of the individual inputs. That is

$$f(\alpha x_1 + \beta x_2) = f(\alpha x_1) + f(\beta x_2) = \alpha f(x_1) + \beta f(x_2),$$

with x and y independent variables and α and β real-valued scalars. The first equality makes use of the additivity (also called the superposition property), and the second the homogeneity property. A system that is not linear is called nonlinear. Since this includes "everything else", it is hard to do a classification and come to general conclusions. Most results in system identification are therefore developed for linear systems, or some limited subset of nonlinear systems. The system is *time invariant* if its response to a certain input signal does not depend on absolute time. A system is said to be *dynamical* if it has some memory or history, i.e., the output does not only depend on the current input but also previous inputs and outputs. If it depends only on the current input, it is *static*.

In system identification, the goal is to use the known input data u and the measured output data y to construct a model of the system S. Here, only *single-input single-output* (SISO) systems are considered, but the ideas can most of the time be adapted to *multiple-input multiple-output* (MIMO) systems. It is usually neither possible nor desirable to find a model that describes the whole system and all its properties. Instead, one wants to construct a model which captures and can describe some interesting subset thereof, which is needed for the given application. It is up to the user to define such criteria as to what needs to be captured by the model.

2.2 Transfer function models

One way to present a *linear time invariant* (LTI) system is via the *transfer function model*

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \tag{2.1}$$

where q is the shift operator, such that $qu(t) = u(t + 1)$ and $q^{-1}u(t) = u(t - 1)$, and $e(t)$ is a white noise sequence. $G(q, \theta)$ and $H(q, \theta)$ are rational functions of q and the coefficients in θ, where θ consists of the unknown parameters that describe the system. Depending on the choice of polynomials in $G(q, \theta)$ and $H(q, \theta)$, different structures can be obtained. A quite general structure is

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \tag{2.2}$$

where the polynomials are described by

$$X(q) = 1 + x_1 q^{-1} + \cdots + x_{n_x} q^{-n_x} \quad \text{for} \quad X = A,\ C,\ D,\ F,$$

and n_x is the order of the polynomial. There is a possible delay n_k in $B(q)$,

$$B(q) = b_{n_k} q^{-n_k} + \cdots + x_{n_k+n_b-1} q^{-(n_k+n_b-1)},$$

such that there can be a delay between input and output. This structure is often too general, and one or several of the polynomials will be set to unity. Depending on the polynomials used, different commonly used structures will be obtained. When the noise is assumed to enter directly at the output, such as white measurement noise, or when we are not interested in modeling the noise, the structure is called an *output error* (OE) model, which can be written

$$y(t) = \frac{B(q)}{F(q)}u(t) + e(t),$$

i.e., the polynomials $A(q)$, $C(q)$ and $D(q)$ have all been set to unity. Many such structures exist (see Ljung [1999] for more examples) and are called *black-box models*, since the model structure reflects no physical insight but acts like a black box on the input, and delivers an output. One strength of these structures is that they are flexible and, depending on the choice of $G(q, \theta)$ and $H(q, \theta)$, they can cover many different cases.

Physical models are sometimes called *white-box models* to highlight that they are see-through and can be built upon physical knowledge about the system. A model which does not belong to the black-box model structure, and is not completely obtained from physical knowledge of the system is called a *gray-box model*. This can for example be a physical structure with unknown parameters, such as an unknown resistance in an elsewise known circuit. It can also be a some properties of the data that can be explored in the choice of model structure. The latter is done in the power amplifier modeling in Chapter 9.

2.3 Prediction error method

One way to say something about the system, is to use a model that can predict what will happen next. At the present time instant t, we have collected data from previous time instants $t-1$, $t-2, \dots$, and this can be used to predict the output. The *one-step-ahead predictor* of (2.2) is

$$\hat{y}(t) = \frac{D(q)B(q)}{C(q)F(q)} u(t) + \left[1 - \frac{D(q)A(q)}{C(q)}\right] y(t), \tag{2.3}$$

and depends only on previous output data. The unknown parameters in the polynomials $A(q)$, $B(q)$, $C(q)$, $D(q)$ and $F(q)$ are gathered in the *parameter vector* θ,

$$\theta = [a_1 \dots a_{n_a} \; b_{n_k} \dots b_{n_k+n_b-1} \; c_1 \dots c_{n_c} \; d_1 \dots d_{n_d} \; f_1 \dots f_{n_f}]^T.$$

The predictor $\hat{y}(t)$ is often written $\hat{y}(t|\theta)$ to point out the dependence on the parameters in θ.

By defining the *prediction error*

$$\varepsilon(t) = y(t) - \hat{y}(t|\theta), \tag{2.4}$$

a straightforward modeling approach is to try to find the parameter vector $\hat{\theta}$, that minimizes this difference,

$$\hat{\theta} = \arg\min_{\theta} V(\theta), \tag{2.5a}$$

$$V(\theta) = \frac{1}{N} \sum_{t=1}^{N} l(\varepsilon(t)) \tag{2.5b}$$

where $l(\,\cdot\,)$ is a scalar valued, usually non-negative, function. Finding the parameters by this minimization is called a *prediction-error (identification) method* (PEM). This idea is illustrated in Figure 2.2.

Except for special choices of the model structures $G(q, \theta)$ and $H(q, \theta)$ and the function $l(\varepsilon)$ in (2.5b), there is no analytical way of finding the minimum of the minimization problem (2.5a). Numerical solutions have to be relied upon, which means that a local optimum might be found instead of the global one if

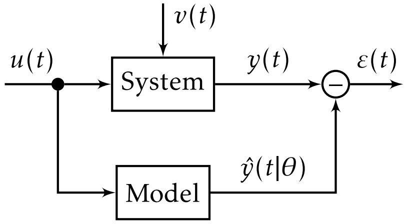


Figure 2.2: *An illustration of the idea behind the prediction error method in system identification. The goal is to minimize the prediction error $\varepsilon(t)$.*

the cost function is nonconvex and has more than one minimum. For results on the convergence of the parameters and other properties of the estimate, such as consistency and variance, see Ljung [1999].

Sometimes, the concept of a *true system* will be used, and the idea is that this true system exists and produces the data. The concept is interesting as it makes it possible to do analytical calculations and gain insight into convergence and other properties of the model, but it might be very hard to describe it mathematically for a real system.

2.4 Linear regression

A common way to describe the relationship between input and output of an LTI system is through a *linear difference equation* where the present output, $y(t)$, depends on previous inputs, $u(t-n_k), \ldots, u(t-n_k-n_b+1)$, and outputs, $y(t-1), \ldots, y(t-n_a)$, as well as the noise and disturbance contributions. This can for example be done for (2.2) when $C(q)$, $D(q)$ and $F(q)$ are set to unity, so that $G(q,\theta)$ and $H(q,\theta)$ in (2.1) correspond to

$$G(q,\theta) = \frac{B(q)}{A(q)}, \quad H(q,\theta) = \frac{1}{A(q)}$$

with

$$\begin{aligned} A(q) &= 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a} \\ B(q) &= b_{n_k} q^{-n_k} + \cdots + b_{n_k+n_b-1} q^{-(n_k+n_b-1)}. \end{aligned}$$

The linear difference equation is then

$$y(t)+a_1y(t-1)+\cdots+a_{n_a}y(t-n_a) = b_{n_k}u(t-n_k)+\cdots+b_{n_k+n_b-1}u(t-n_k-n_b+1)+e(t),$$

and we can write

$$A(q)y(t) = B(q)u(t) + e(t). \tag{2.6}$$

This particular structure is called *auto-regressive with external input* (ARX). Another special case is when the output only depends on past inputs, such that $n_a = 0$ in (2.6). This is called a *finite impulse response* (FIR) structure.

The predictor for an ARX model is

$$\hat{y}(t|\theta) = -a_1 y(t-1) - \cdots - a_{n_a} y(t-n_a) + \\ b_{n_k} u(t-n_k) + \cdots + b_{n_k+n_b-1} u(t-n_k-n_b+1). \tag{2.7}$$

By gathering all the known elements into one vector, the *regression vector*,

$$\phi(t) = [-y(t-1), \ldots, -y(t-n_a)\ u(t-n_k), \ldots, u(t-n_k-n_b+1)]^T$$

and the unknown elements into the parameter vector,

$$\theta = [a_1 \ldots a_{n_a}\ b_{n_k} \ldots b_{n_k+n_b-1}]^T,$$

the predictor (2.7) can be written as a *linear regression*

$$\hat{y}(t|\theta) = \phi^T(t)\theta, \tag{2.8}$$

that is, the unknown parameters in θ enter the predictor linearly.

2.5 Least-squares method

With the function $l(\,\cdot\,)$ in (2.5b) chosen as a quadratic function,

$$l(\varepsilon) = \frac{1}{2}\varepsilon^2,$$

and the predictor described by a linear regression, as in (2.8), we get

$$V(\theta) = \frac{1}{2N}\sum_{t=1}^{N}\left[y(t) - \phi^T(t)\theta\right]^2, \tag{2.9}$$

called the *linear least-squares* (LS) criterion. A good thing about this criterion is that it is quadratic in θ, which means that the problem is convex and the minimum can be calculated analytically. The minimum is obtained for

$$\hat{\theta}^{LS} = \left[\frac{1}{N}\sum_{t=1}^{N}\phi(t)\phi^T(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\phi(t)y(t), \tag{2.10}$$

and is called the *least-squares estimator*. See for example Draper and Smith [1998] for a more thorough description of the LS method and its properties.

Apart from the guaranteed convergence to the global optimum, a benefit with LS solutions is that there exist many efficient numerical methods to solve them. The *recursive least-squares* (RLS) method can be used to solve the numerical optimization recursively. Another option is the *least mean square* (LMS) method, which can make use of the linear regression structure of the optimization problem in (2.8). These methods are described in, for example, Ljung [1999, Chapter 11].

2.6 Nonlinear and nonconvex system identification

For many model structures, both linear and nonlinear, it is not possible to write the model in such a way that it can be put in the linear regression form (2.8).

2.6.1 Separable least-squares

For some model structures, the parameter vector can be divided into two parts, $\theta = [\rho^T \; \eta^T]^T$, so that one part enters the predictor linearly and the other nonlinearly, i.e.,

$$\hat{y}(t|\theta) = \hat{y}(t|\rho, \eta) = \phi^T(t, \eta)\rho.$$

Here, for a fixed η, the predictor is a linear function of the parameters in ρ. The identification criterion is then

$$V(\theta) = V(\rho, \eta) = \frac{1}{2N} \sum_{t=1}^{N} \left[y(t) - \phi^T(t, \eta)\rho \right]^2$$

and this is an LS criterion for any given η. Often, the minimization is done first for the linear ρ and then the nonlinear η is solved for. The nonlinear minimization problem now has a reduced dimension, where the reduction depends on the dimensions of the linear and nonlinear parameters. This method is called *separable least-squares* (SLS) as the LS part has been separated out, leaving a nonlinear problem of a lower dimension, see Ljung [1999, p. 335-336].

2.6.2 Nonlinear system linear in the parameters

There are also nonlinear model structures where the parameters enter linearly. One example is the model where

$$y = \sum_{k=1}^{K} \alpha_k f_k(\phi) \tag{2.11}$$

and the $f_k, k = 1, \ldots, K$ are nonlinear functions of the regression vector ϕ. Since the parameters $\alpha_k, k = 1, \ldots, K$ enter the equation linearly, this system is still *linear in the parameters* and the least-squares formulation can be used. This can also be seen as a redefinition of ϕ where instead of the original signals, we use nonlinear transformations of the same.

2.6.3 No least-squares formulation?

In many cases, we end up in a problem formulation that does not fit into the least-squares formulation. There are many numerical optimization algorithms to find the minimum of a function, when it is not possible to find an analytical solution. These include the gradient descent method that uses the gradient of the function to find a minimum and Newton's method that uses curvature information of the Hessian to find the minimum faster. The Gauss-Newton method is a modification of Newton's method that does not require second derivatives, which can be hard to compute.

In the case where no analytical solution is available, the optimization problem is often not convex, and there can be multiple local optima. Many different methods and heuristics have been developed to solve the optimization, see for example Ljung [1999, Chapter 10] and the references therein. One option is to use local solutions that expand the search region in different ways. This can be done by allowing the optimization some steps uphill (in a minimization criterion) to pass local maxima in search of a better local minimum, ideally the global one. Yet another option is to add some stochasticity to the solution. The results of these methods often depend on the initial guess of the parameter vector – how close it is to the global optimum. If the initial estimates are very poor, the methods will have a hard time finding the global optimum and will get stuck in a local optimum. Examples of these heuristics are simulated annealing, particle swarm optimization and evolutionary algorithms. The Nelder-Mead simplex method has been used in this thesis. Another idea is to use gridding where the whole domain, or a subset thereof, is evaluated within a predetermined accuracy, and all possible parameter combinations are evaluated. This can be very heavy numerically, and the number of cost function evaluations depends on the size of the domain and the precision for each parameter.

The trade-off here is the number of calculations vs. the search region, and how close to the global optimum we want to get. In nonconvex optimization, no guarantees can be made regarding the optimality of the solution.

2.7 Instrumental variables

So far the goal has been to minimize the prediction error. A different approach to system identification is to use a correlation approach. In the *instrumental variables* (IV) method we want to find *instruments* or *instrumental variables* $\zeta(t)$ that are correlated with the regression vector but uncorrelated with the noise. In this overview of the method we will assume the model is a linear regression (2.8) [Ljung, 1999].

In the IV method, the covariance between the prediction error (2.4), $\varepsilon(t) = y(t) - \phi^T(t)\theta$ and the instruments $\zeta(t)$ should be zero,

$$\hat{\theta}^{IV} = \text{sol}\left\{\frac{1}{N}\sum_{t=1}^{N}\zeta(t)\left[y(t) - \phi^T(t)\theta\right] = 0\right\}. \tag{2.12}$$

This can also be written as

$$\hat{\theta}^{IV} = \left[\frac{1}{N}\sum_{t=1}^{N}\zeta(t)\phi^T(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\zeta(t)y(t), \tag{2.13}$$

provided the inverse exists. Of course, the choice of instruments heavily impacts the performance of the method. Possible choices of instruments are simulated outputs (for example based on an LS estimate) and shifted inputs and outputs (assuming the orders of the system and the noise model are such that it is possible).

2.8 The system identification procedure

The process of constructing a model from data consists of a number of steps, which often have to be performed multiple times before a suitable model can be obtained. See for example Ljung [1999] for a more thorough discussion of the different steps.

1. A data set is needed, usually containing input and output data. The data should be "rich enough", so that it excites the desired properties of the system. This is called persistency of excitation.

2. Different model structures should be examined, to evaluate which structure best captures the properties of the data. These structures should fulfill certain demands, such that two sets of parameters do not lead to the same model. This property is called *identifiability*.

3. A measurement of "goodness", such as the criterion (2.5), has to be selected to decide which models best describe the data.

4. The model estimation step is where the parameters in θ are determined. In the LS method, this would consist of inserting the data into (2.10), and in the PEM case, the minimization of (2.5) for a certain choice of predictor structure $\hat{y}(t|\theta)$ in (2.3).

5. Model validation. In this step, different models should be evaluated to determine if the models obtained are good enough. The evaluation should be done on a new set of data, *validation data*, to ensure that the model is useful not only for the data for which it was estimated. Two important components of the model validation are the comparisons between measured data and model output as well as the residual analysis, where the statistics of the unmodeled properties of the data are evaluated.

Some of these steps contain a large user influence, whereas others might be set or rather straightforward. The choice of model structure and model order, such as n_a and n_b in (2.7), is often hard and needs to be repeated a number of times before a suitable model can be found.

Part I

Estimation of inverse systems

3

Introduction to system inversion

Inverse systems are used in many applications, more or less visibly. One application example is power amplifiers in communication devices, which are often nonlinear, causing interference in adjacent transmitting channels [Fritzin et al., 2011a]. This interference will be noise to anyone that transmits in these neighboring channels, and there are measures describing the amount of power that is allowed to be spread to adjacent frequencies. If the inverse of the nonlinearity can be found and applied to the signal, the noise should be canceled. However, one does not want to work with the amplified signal, but rather with the input signal to the system, that is, before the signal is amplified. A prefilter that inverts the nonlinearities, called a *predistorter*, is thus desired.

In sensor applications it is rather a *postdistortion* that is needed. If the sensor itself has dynamics or a nonlinear behavior, the sensor output is not the true signal but will also contain some sensor contamination. This would have to be handled at the sensor output, since this is where the user can get access to the signal.

In the area of robotics, there is a need for control such that the robot achieves the demands on precision. Smaller and lighter robots reduce the need for large motors, as well as the cost and wear of the robot. However, this also introduces new problems such as larger oscillations and increases the demands on the control performance. In robotic control applications, a common strategy is to use feedback to control the joint positions. The last part of the robot, however, connecting the tool to the robot, is often controlled using open-loop control. Models of both the forward and the inverse kinematics are used for control.

In the above applications, finding the inverse of the system is a crucial point; how should the input to or the output from the system be modified to obtain the desired dynamics from input to output? Each application entails its own restrictions and special conditions to attend to, and in this chapter, some aspects of system inversion are discussed. For a nonlinear system, the inversion is nontrivial, and different approaches can be used. A selection of methods is presented here.

When an inverse exists, there is a one-to-one relation between input and output and this property is called bijectivity. If the system or function is not bijective, finding an inverse is not possible. However, an approximate inverse can still be useful. In parts of this thesis we assume that the system and the inverse can both be written analytically, see Example 3.1 for a case when this is not valid. Both the system and the inverse are assumed to be stable and causal (see for example Rugh [1996]). The systems considered in this thesis are *single-input single-output* (SISO). In this chapter, the main focus is on inversion, and a model of the system is supposed to be known, either by physical modeling or by system identification. Different approaches to estimate inverse models will be presented in Chapter 5.

Example 3.1: Nonexisting analytical inverse

Consider the system

$$y(t) = e^{x(t)} + \sin(x(t))$$

for $|x(t)| < 0.5\pi$. The function $e^x + \sin(x)$ is monotonic on $[-\frac{\pi}{2}\ \frac{\pi}{2}]$, and thus also invertible. However, no analytic expression of the inverse exists, and a numerical inverse will have to be used.

Here, the methods are described in either continuous or discrete time. Different frameworks are usually most easily described in one domain or the other, hence the mixed use in this chapter. Also, the systems are often continuous whereas the controllers are implemented in discrete time. The explicit dependency on time will sometimes be left out for notational convenience.

This chapter mostly contains an overview of methods of system inversion already present in the literature. The new contribution is the adaptation of the Hirschorn method to a postinverse, presented in Section 3.3.2.

3.1 Inversion by feedback

The behavior of a system can be modified in a multitude of ways, often with the goal of making the output follow a desired trajectory, called *reference signal*, r. In the automatic control society the main choices are feedback and feedforward control. For the linear case many different control strategies exist, perhaps the most common of which is the PID, consisting of a proportional (P), an integral (I) and a derivative (D) part. The P, I and D parameters of the controller can be trimmed to obtain a desired behavior of the controlled system. The concept of controllers using PID has been used since the 18th century, but a recent contribution is for example Åström and Hägglund [2005].

In this section, a few feedback strategies will be introduced. An iterative control approach that can be used for linear and nonlinear systems is the *iterative learning control* (ILC). ILC works on systems with a repetitive input signal, such as a robot performing the same task over and over again. It makes use of the output from the last repetition and tries to improve this so that the output better follows the reference signal. Another feedback solution for nonlinear systems is the *exact input-output linearization*, which makes use of a known model of the system to obtain overall linear dynamics, determined by the user.

Though the classical view of feedback control is not that of system inversion, this is indeed one interpretation; the feedback system produces the input that leads to a desired output. This is also the goal of an inverse system, to produce an input by use of an output. We will start this chapter by covering a few control strategies.

3.1.1 Feedback and feedforward control

Feedback control refers to using a measured output of a system to determine the input to said system. A standard solution is to look at the difference between the reference $r(t)$ and the output $y(t)$, called *control error* $e(t) = r(t) - y(t)$, and this signal can be used for control of the system. For example, if the control error is negative, a conclusion can be that the input $u(t)$ is too small, and should be increased, and vice versa. Many control strategies based on this idea have been constructed and form the basis of the control laws used in industry. The idea is presented in Figure 3.1 where a feedback controller F is applied to the system G.

On the other hand, if we know something about how the system will transform the input, we might want to use this to counteract later effects. This is the concept of *feedforward control*, where the reference signal is altered and sent to the system, or fed forward. Often, feedforward and feedback control are used together to get the advantages of both approaches. Figure 3.2 shows a block diagram where the feedback loop in Figure 3.1 has been expanded to include a feedforward loop with the feedforward controller F_f. A common requirement is that the output should have a softer behavior than the reference, and this can be achieved via the filter G_m, denoting the desired dynamics. The ideal choice of the feedforward controller is $F_f = G_m/G$. If feedforward control is used alone, with no feedback loop, it is often called *open-loop control*.

Feedback control can handle phenomena like disturbances and model uncertainties, since it is based on the true output. It can also handle unstable systems, which is not possible for a pure feedforward (open-loop) control, however a bad feedback loop may cause instability.

Feedforward control has the advantage of not needing any measurement but the drawback is that ideal feedforward control (using $F_f = G_m/G$) requires perfect knowledge of the system, and that both G and G_m/G are stable. Also, there is no possibility to compensate for disturbances. However, if the disturbances are perfectly known or measurable, feedforward control from the disturbances can be applied and the disturbances compensated for. These are of course limiting assumptions. A benefit with feedforward control is that two cascaded stable systems will always be stable, hence a bad controller can not destabilize the system.

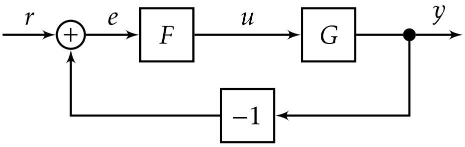


Figure 3.1: A feedback controller F applied to the system G.

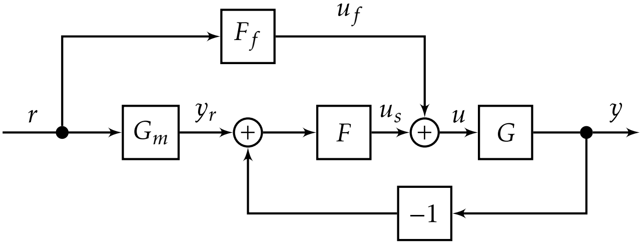


Figure 3.2: Feedforward controller F_f and feedback controller F applied to a system G. G_m is used here to describe the desired dynamics between reference and output.

3.1.2 Iterative learning control

As discussed in the introduction, *iterative learning control* (ILC) can be seen as an iterative inversion method [Markusson, 2001, Markusson and Hjalmarsson, 2001]; the goal is to find the input leading to the desired output. In this section, the basic concepts of ILC will be described, but for a more thorough analysis see for example Wallén [2011], Moore [1993] and the references therein. The ILC concept comes from the industrial robot application, where the same task or motion is performed repeatedly. The idea is to use the knowledge of how the controller performed in the last repetition and improve the performance in each iteration.

The system S in this setting is described by the input u, the output y, and the reference r over a finite time interval. The task is assumed to be repeated, so that the reference r and the starting point are the same for each iteration. The time index is t, where $t \in [0, N-1]$ for each repetition, and each repetition is of length N. A basic first order ILC algorithm is described by

$$u_{k+1}(t) = Q(q)\left(u_k(t) + L(q)e_k(t)\right) \tag{3.1}$$

where

$$e_k(t) = r(t) - y_k(t)$$

and k is the iteration index, indicating how many times the task has been repeated. Here, q is the shift operator, that is $q^{-1}u(t) = u(t-1)$, while $Q(q)$ and $L(q)$ denote linear or nonlinear operators, chosen by the user. It is important that this choice leads to convergence to an input where the output achieves the desired performance. Also, the learning should be fast enough. There are structured ways to determine $Q(q)$ and $L(q)$, which can be based on a model of the system. The concepts of stability and convergence of ILC systems are treated in, for example, Wallén [2011]. It can be shown that ILC is robust to model errors, such that for a linear system, a relative model error of 100% can be tolerated [Markusson, 2001]. Even a rather simple model can therefore perform well.

One of the limits of ILC is that it can only be used on tasks that are performed repeatedly, and the performance can deteriorate significantly if the tasks are slightly different. One solution to this is the use of basis functions in the ILC, which enables ILC use for slightly varying tasks [Boeren et al., 2015].

Iterative methods are used in many applications, also outside the control community. The common factor is that the information found in the output y is used to improve the input, but the algorithm is not necessarily similar to (3.1). One application where iterative solutions are often used is *analog-to-digital converter* (ADC) correction, such as in Soudan and Vogel [2012].

3.1.3 Exact linearization

In *exact linearization* (also called input-output linearization) [Sastry, 1999], the output from a nonlinear system S,

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x), \end{aligned} \tag{3.2}$$

which is affine in u, is differentiated enough times to obtain a relation between the differentiated output $y^{(n)}$ and the input, u. Differentiating y with respect to time, we obtain

$$\begin{aligned} \dot{y} &= \frac{\partial h}{\partial x} f(x) + \frac{\partial h}{\partial x} g(x)u \\ &= L_f h(x) + L_g h(x)u, \end{aligned}$$

where $L_f h(x)$ and $L_g h(x)$ are the Lie derivatives of h with respect to f and g, respectively. If $L_g h(x) \neq 0$, a relation between the differentiated output $\dot{y}$ and the input u has been obtained, then using the input,

$$u = \frac{1}{L_g h(x)}(-L_f h(x) + r)$$

leads to a linear relation between output and reference, $\dot{y} = r$. If $L_g h(x) = 0$, a second differentiation can be performed,

$$\begin{aligned} \ddot{y} &= \frac{\partial L_f h}{\partial x} f(x) + \frac{\partial L_f h}{\partial x} g(x)u \\ &= L_f^2 h(x) + L_g L_f h(x)u, \end{aligned}$$

from which a control law can be calculated if $L_g L_f h(x) \neq 0$. In this manner, one can continue until there is a direct relation between $y^{(\gamma)}$ and r through the control law

$$\begin{aligned} u &= \frac{1}{L_g L_f^{\gamma-1} h(x)} (-L_f^{\gamma} h(x) + r) \\ &= \alpha(x) + \beta(x) r. \end{aligned} \tag{3.3}$$

Here, γ is the smallest integer for which $L_g L_f^i h(x) \equiv 0$ for $i = 0, 1, \ldots, \gamma - 2$ and $L_g L_f^{\gamma-1} h(x) \neq 0$, and it is called the relative degree of the system.

The system (3.2) with control input (3.3) now describes a system with linear dynamics. Hence, linear theory can be used to obtain the desired dynamics chosen by the user, G_m, and the linear feedback loop can be combined with the nonlinear one. The overall system from r to y (the nonlinear system with the nonlinear and linear feedback) will thus be linear, and the dynamics will be described by the transfer function G_m.

Exact linearization requires knowledge of all the states, and is therefore often used in combination with a nonlinear observer. This can lead to a complicated feedback loop. Here, it is assumed that any zero dynamics present are stable. The above system and the derivation of the feedback loop are described in continuous time. A discrete-time description can also be done, as presented in Califano et al. [1998].

3.2 Analytic inversion

In the above feedback loops, only the system itself, or a model thereof, is used to produce an inverse. No explicit inversion is done. Another approach is to perform an analytic inversion of the system, which can be applied at the input to, or the output from, the system, see Figure 1.1. The output from this cascaded system should have the desired dynamics. If the goal is to make the output exactly the same as the reference in a noise-free setting, a "true" inverse has to be found. But even for other situations, the inversion can be seen as a case where the undesired nonlinear and linear dynamics have been inverted. For example, in the exact linearization case, the nonlinear and dynamical behavior of the system are inverted, and in the end a system with some user-defined linear dynamics is obtained. This approach has already been used in the feedforward controller $F_f = G_m/G$, where the system G is inverted.

There are several ways to find a system inverse. One method for finding an inverse to dynamic systems uses Volterra series, which is a nonlinear extension of the impulse response concept for the linear case. This approach leads to an analytical inverse. Other systems that might be analytically invertible are block-oriented systems, which consist of a static nonlinearity and a linear dynamic system. A brief overview of Volterra series will be presented here together with a short discussion on the use of preinverse and postinverse and problems that occur with inversion.

3.2.1 Problems occurring with system inversion

Here, the goal is not to cover all problems with the inversion of systems, but to give some insights to the problems that can occur.

For a stable and minimum-phase LTI system G, it is rather straightforward to find an inverse G^{-1}. However, if these conditions are not fulfilled, we quickly run into problems, even for linear systems. Any nonminimum-phase zeros of the original system will become unstable poles of the inverse system. However, if the system is nonminimum phase, the inverse can be used if noncausal filtering is allowed. If a delay can be allowed, time-reversed input and output sequences can be used together with a matching, stable inverse [Markusson, 2001].

Another aspect with inverse systems concerns whether the system is proper or not. A proper transfer function is one where the order of the denominator is greater than or equal to that of the numerator. A strictly proper transfer function is one where the order of the denominator is greater than that of the numerator. The amplification of a proper system always approaches a value as the frequency goes to infinity. If the transfer function is strictly proper, the amplification will approach zero at high frequencies. For a transfer function that is not proper, however, the amplification will approach infinity when the frequency approaches infinity. That is, high frequency contents will be amplified. This means that the inverse of a strictly proper system will be improper.

3.2.2 Postinverse and preinverse

Loosely speaking, a *postinverse* to a system $\mathcal{S}$ is a system $\mathcal{T}$ such that the system $\mathcal{T}\mathcal{S}$ behaves approximately as a unit mapping, while a *preinverse* $\mathcal{R}$ makes $\mathcal{S}\mathcal{R}$ act like a unit mapping. The concepts will be more formally defined in Section 4.1, where it will also be clear why it is important to distinguish between them.

As is commonly known, the ordering of two linear systems does not matter, i.e., the output from $A \cdot B$ equals the output from $B \cdot A$ when A and B are linear dynamical systems. This property is called commutativity. However, this does not apply to nonlinear systems, as shown in Example 3.2.

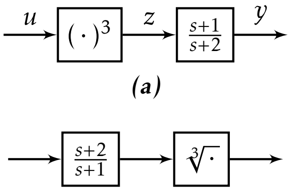
Example 3.2: Noncommutativity of nonlinear systems

Consider the two functions

$$f_1(x) = 2x \quad \text{and} \quad f_2(x) = x^2.$$

If the order of the systems is f_1, f_2, the output is $y_{12} = 4u^2$ and with the reversed order, the output is $y_{21} = 2u^2 \neq y_{12}$.

Thus, for nonlinear systems, the output depends on the order of the systems. There are some exceptions where this is not true and the systems can change order without changing the output. One example where two nonlinear systems commute, is where one of the systems is the inverse of the other, as in Example 3.3 for a Hammerstein-Wiener system. When an exact inverse exists, the preinverse



(a)

(b)

Figure 3.3: *(a) A Hammerstein system with invertible static nonlinearity followed by a linear, stable minimum-phase dynamical system. For such a system, an analytical inverse exists, as shown in (b).*

and the postinverse are the same in a noise-free case. However, it is often not possible to determine the exact inverse, and an approximate inverse has to be used. This approximate function does not necessarily commute with the system.

Example 3.3: Analytical inversion

Consider the Hammerstein system with the static nonlinearity

$$f_H(x) = x^3$$

which is invertible for all x, followed by the minimum-phase linear dynamic system

$$G_H(s) = \frac{s+1}{s+2},$$

as shown in Figure 3.3a. For this system, an analytical inverse exists, namely the Wiener system

$$G_W(s) = \frac{s+2}{s+1}, \quad f_W(x) = \sqrt[3]{x},$$

see Figure 3.3b. This inverse is also an example of where a nonlinear system and its inverse are commutative, that is, the two systems can be placed in whichever order. This Wiener system can thus be used as a *pre*inverse or a *post*inverse.

Another example of nonlinear systems that commute are the Volterra series and the p-th order Volterra inverse that will be described in the next section. But, in general, the commutative property does not apply to nonlinear systems, see Mämmelä [2006] for an extended discussion on commutativity in linear and nonlinear systems.

In Chapter 4 it will be shown that the optimal preinverse and postinverse are not necessarily the same. Various modeling approaches will be further considered in Chapter 5, leading to either a preinverse or a postinverse. Which one is requested is often closely connected to the application, for instance for power amplifier linearization a preinverse is desired, and for sensor calibration a postinverse. In power amplifier predistortion, the commutativity property is often considered approximately valid, and the pre- and postinverses are used interchangeably without further consideration [Abd-Elrady et al., 2008, Paaso and Mämmelä, 2008].

3.2.3 Volterra series

In the linear systems theory, a common way to describe the output $y(t)$ of the system affected by the input $u(t)$, is by the impulse response $g(\cdot)$,

$$y(t) = \int_{-\infty}^{\infty} g(\tau)u(t-\tau)\mathrm{d}\tau, \tag{3.4}$$

usually with the added constraints that the system is causal and the input zero for $t < 0$, so that the integral is limited to $[0, t]$. It can also be described by the corresponding Laplace relation

$$Y(s) = G(s)U(s), \tag{3.5}$$

where $Y(s)$ and $U(s)$ are the Laplace transformed versions of $y(t)$ and $u(t)$, respectively, and $G(s)$ is the transfer function. This is not possible for nonlinear systems. However, if the nonlinear system is time invariant with certain restrictions, an input-output relation can be determined. These conditions include convergence of the infinite sums and integrals that occur [Sastry, 1999], but will not be further considered here. The input-output relation can be described by

$$\begin{aligned} y(t) = &\int_{-\infty}^{\infty} h_1(\tau_1)u(t-\tau_1)\mathrm{d}\tau_1 + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)u(t-\tau_1)u(t-\tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2 + \ldots \\ &+ \int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty} h_n(\tau_1,\ldots,\tau_n)u(t-\tau_1)\ldots u(t-\tau_n)\mathrm{d}\tau_1\ldots\mathrm{d}\tau_n + \ldots \end{aligned} \tag{3.6}$$

where

$$h_n(\tau_1,\ldots,\tau_n) = 0 \quad \text{for any } \tau_j < 0, \quad j = 1,2,\ldots,n.$$

The relation (3.6) is called a Volterra series (sometimes Volterra-Wiener series) and the functions $h_n(\tau_1,\ldots,\tau_n)$ are called the *Volterra kernels* of the system. The expression (3.6) can also be written as

$$y(t) = \mathbf{H}_1[u(t)] + \mathbf{H}_2[u(t)] + \cdots + \mathbf{H}_n[u(t)] + \ldots \tag{3.7}$$

where

$$\mathbf{H}_n[u(t)] = \int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty} h_n(\tau_1,\ldots,\tau_n)u(t-\tau_1)u(t-\tau_2)\ldots u(t-\tau_n)\mathrm{d}\tau_1\ldots\mathrm{d}\tau_n \tag{3.8}$$

is called an n-th order *Volterra operator.*

When considering an LTI *single input-single output* (SISO) system, the Volterra series reduces to the standard form, and the kernel $h_1(\cdot)$ in (3.6) corresponds to $g(\cdot)$ in (3.4). See for example Schetzen [1980] for a more thorough description of Volterra series. The counterpart of the transfer function is based on the multivariable Fourier transform,

$$H_p(j\omega_1,\dots,j\omega_p) = \int\limits_{-\infty}^{\infty} \dots \int\limits_{-\infty}^{\infty} h_p(\tau_1,\dots,\tau_p) e^{-j(\omega_1\tau_1+\dots+\omega_p\tau_p)} \mathrm{d}\tau_1 \dots \mathrm{d}\tau_p \tag{3.9}$$

called the p-th order kernel transform. The inverse relation is

$$h_p(\tau_1,\dots,\tau_p) = \frac{1}{(2\pi)^p} \int\limits_{-\infty}^{\infty} \dots \int\limits_{-\infty}^{\infty} H_p(j\omega_1,\dots,j\omega_p) e^{j(\omega_1\tau_1+\dots+\omega_p\tau_p)} \mathrm{d}\omega_1 \dots \mathrm{d}\omega_p. \tag{3.10}$$

In analogy to the linear case, these functions are sometimes referred to as higher order transfer functions. The discrete counterpart of the Volterra operators (3.8) is [Tummla et al., 1997]

$$\mathbf{H}_n[u(t)] = \sum_{i_1=-\infty}^{\infty} \dots \sum_{i_n=-\infty}^{\infty} h^{(n)}_{i_1,i_2,\dots,i_n} u(n-i_1)\dots u(n-i_n). \tag{3.11}$$

This version is often used in data-based modeling, where the models are based on sampled data.

p-th order Volterra inverse

A p-th order inverse, $\mathbf{H}^{-1}_{(p)}$, is defined as a system that, when connected in series with the nonlinear system $\mathbf{H}$, results in a system, $\mathbf{Q}$, in which the first-order Volterra kernel is a unit pulse and the other Volterra kernels are zero up to order p, $q_k = 0, k = 2,\dots,p$. The Volterra kernels for $k > p$ might however be nonzero but are generally considered to be negligible [Zhu et al., 2008]. The inverse, $\mathbf{H}^{-1}_{(p)}$, can be determined by using the Volterra series (assumed known) of the system, and the desired output. This is done in a sequential way by first finding the first order Volterra operator, $\mathbf{H}^{-1}_{(1)}$, and then solving for the higher order Volterra operators $\mathbf{H}^{-1}_{(n)}$, $n = 2,\dots,p$, which then only depend on the system $\mathbf{H}$ and lower order operators of the inverse, see Schetzen [1980, Chapter 7] for a thorough discussion.

The ordering of the system $\mathbf{H}$ and the inverse $\mathbf{H}^{-1}_{(p)}$ will affect the output, but it can be shown [Schetzen, 1980] that the first p Volterra operators of the connected systems are the same. The order of the system $\mathbf{H}$ and the inverse $\mathbf{H}^{-1}_{(p)}$ can thus be interchanged and the postinverse $\mathbf{H}^{-1}_{(p)}$ can also be used as a preinverse, if only the nonlinearities up to order p are of interest.

3.3 Inversion by system simulation

Some approaches to avoid the explicit inversion of a system are based on a simulation of the true system, without including any feedback from the actual system.

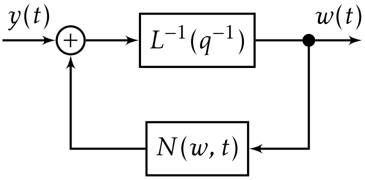


Figure 3.4: *An inversion method that only uses the inverse of the linear part L of a nonlinear system $S = L + N$.*

The exact linearization described in Section 3.1.3 can be modified such that it uses a simulated output in the feedback loop. Another approach is to decompose the original system to avoid the explicit inversion of the nonlinear system. The idea with these inversion methods is that the inverse can be used as a preinverse or a postinverse, thus avoiding a feedback loop.

3.3.1 Separation of a nonlinear system

A way to avoid the explicit inversion of a nonlinear system is presented in Markusson [2001, p. 51]. There, the nonlinear system S is separated into a linear part, L, and a nonlinear part, N, where operator notation is used. The inverse of $S = L+(S-L) = L+N = L(I+L^{-1}N)$, can then be written as $S^{-1} = (I+L^{-1}N)^{-1}L^{-1}$. We have thus obtained a postinverse S^{-1} such that

$$S^{-1}S = (I + L^{-1}N)^{-1}L^{-1}(L + N) = (I + L^{-1}N)^{-1}(I + L^{-1}N) = I.$$

This can also be used as a preinverse, since

$$SS^{-1} = (L + N)(I + L^{-1}N)^{-1}L^{-1} = L(I + L^{-1}N)(I + L^{-1}N)^{-1}L^{-1} = I.$$

The inverse $(I + L^{-1}N)^{-1}L^{-1}$ can be obtained in a feedback loop, with the nonlinear part N in the feedback and the linear inverse L^{-1} in the forward path (compare to the sensitivity function for LTI systems), see Figure 3.4. It follows that the nonlinear part N does not have to be explicitly inverted, and that only the linear part L is to be inverted. The output from the inverted system is denoted $w(t)$ to separate it from the true input $u(t)$, since different initial conditions of the true system and the model will produce an output that is not exactly equal to the input. Unknown initial states are discussed in Markusson [2001, p. 45], in a *maximum likelihood* (ML) setting.

3.3.2 Hirschorn's method

Another approach to invert nonlinear systems is Hirschorn's method, where exact linearization is used in order to construct a linear system [Hirschorn, 1979]. Given that the model is good enough, it should be possible to use it not only in the feedback to construct u as in (3.3), but also as a simulation model.

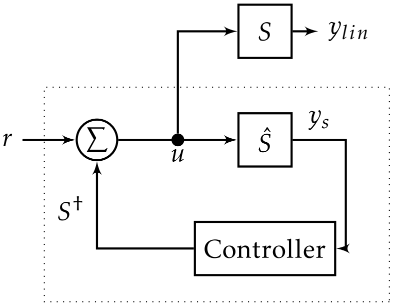


Figure 3.5: *A block diagram of Hirschorn's method, where the system S is replaced by a model $\hat{S}$ in the exact linearization feedback loop. The input signal calculated in this way is then also applied to the real system S. The simulation system and feedback loop that leads to an overall linear behavior between r and y_s is denoted $S^{\dagger}$. The input to $S^{\dagger}$ is the reference r and the output is the control signal u.*

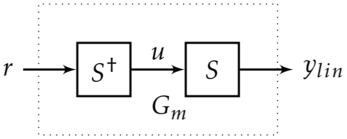


Figure 3.6: *The predistortion block $S^{\dagger}$ obtained using Hirschorn's method in series with the real system leads to an overall linear behavior between r and y_{lin}.*

Preinversion

If instead of the measured output from the system, the output from the simulated model is fed back to the controller, see Figure 3.5, the overall system (from reference r to output y_s) will by construction be linear with the dynamics G_m. Also, the input calculated for this (simulated) system leads to the desired dynamics, and the same input signal can be used also for the true system. The system from r to u will be denoted $S^{\dagger}$. A pure open loop controller is thus obtained, as in Hirschorn [1979], see Figure 3.6, and this is called *Hirschorn's method*. The simulated feedback can also be interpreted as an observer with no measurement inputs.

Hirschorn's method in combination with a small feedback loop has been shown to give promising results in attenuating harmonic distortion in loudspeakers [Arvidsson and Karlsson, 2012]. When a model of the system is available, the same idea could be able to use not only for exact linearization, but also for other complicated control feedback loops.

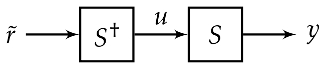


Figure 3.7: The (possibly fictitious) reference signal $\tilde{r}$ can be seen as input to the block $S^\dagger$, creating the input u to the nonlinear system S.

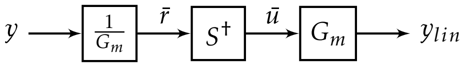


Figure 3.8: Hirschorn's method applied as postdistortion, when the output can be assumed to be created according to Figure 3.7. The block $S^\dagger$ cannot simply be applied at the output y, but has to be manipulated to obtain a linear behavior between u and y_{lin}.

Postinversion

Hirschorn's method describes a preinversion, which we have adapted and expanded into a postinversion method described here.

Let the nonlinear system be denoted S and the precompensation be denoted $S^\dagger$, since it is not really an inverse of S, but rather creates a system that, in series with S, will be linear. The dynamics of the overall linear system is G_m.

The method described above can be seen as an inversion of the nonlinearities of the system – the output from the overall system will be linear with dynamics G_m chosen by the user. This is based on the assumption that the model is accurate enough, of course. This is a setup where preinverse and postinverse are not interchangeable; Hirschorn's method tells us only how to determine the input to the nonlinear system such that the reference-to-output has the linear dynamics G_m, not how to manipulate the output to make it a linear response to the input. If it is this postinverse that is wanted, a different setup is needed.

It is known that $S^\dagger$ in cascade with S yields a linear system G_m, so that

$$y = G_m r \tag{3.12}$$

with r being the reference, cf. Figure 3.6. The goal is to obtain a linear response to u by using a postinverse on the output y. Assume that u was actually created by a prefilter, $S^\dagger$, with u as output and the fictitious signal $\tilde{r}$ as input, as in Figure 3.7. An estimate of this signal can then be obtained by

$$\bar{r} = \frac{1}{G_m} y, \tag{3.13}$$

where, if no transients or noise are present, $\bar{r} = \tilde{r}$. An estimate of the input u, called $\bar{u}$, can be obtained by filtering $\bar{r}$ by $S^\dagger$. Now, to obtain the desired dynamics, $\bar{u}$ must be filtered by the linear function G_m, see Figure 3.8. The cascade of these three blocks ($1/G_m$, $S^\dagger$ and G_m), thus makes up a postdistorter that leads to a linear response between u (not available for manipulation) and y_{lin} in Figure 3.9. This method is illustrated in Example 3.4.

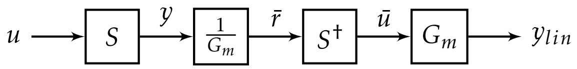


Figure 3.9: *Hirschorn's method used as postdistortion. The postinverse consists of the three blocks $1/G_m$, $S^{\dagger}$ and G_m. Used in this way, the overall behavior between u and y_{lin} will be linear.*

Example 3.4: Hirschorn's postinverse

Consider the nonlinear system

$$\begin{aligned} \dot{x}_1 &= -x_1^3 + x_2 + w_1 \\ \dot{x}_2 &= -x_2 + u + w_2 \\ y &= x_1 \end{aligned} \tag{3.14}$$

with process noise $w_i \in N(0, 0.05)$ and a multisine input. The nonlinear feedback

$$u = -3x_1^5 + 3x_1^2 x_2 + x_2 + \tilde{u}$$

leads to a linear system $\ddot{y} = \tilde{u}$. Now, linear theory can be applied and pole placement has been used to get an overall system response from reference r to output y corresponding to the one from

$$G_m(s) = \frac{1}{s^2 + 5s + 6}. \tag{3.15}$$

The output from the nonlinear system (3.14) is plotted in Figure 3.10 together with the output from the desired dynamics G_m.

A preinverse $S^{\dagger}$ has been constructed as in Figure 3.5. $S^{\dagger}$ has been used as a preinverse, as well as a postinverse for evaluation purposes. The results are shown in Figure 3.11. Here, it is clear that the desired preinverse and postinverse are not the same, and that $S^{\dagger}$ cannot straight away be used as a postinverse. If instead, the output y is filtered by the cascaded systems $1/G_m$, $S^{\dagger}$ and G_m, as in Figure 3.8, the result improves considerably, as shown in Figure 3.11. The remaining errors are primarily caused by the noise. For noise-free data, the preinverse performs perfectly whereas the postinverse has some minor errors.

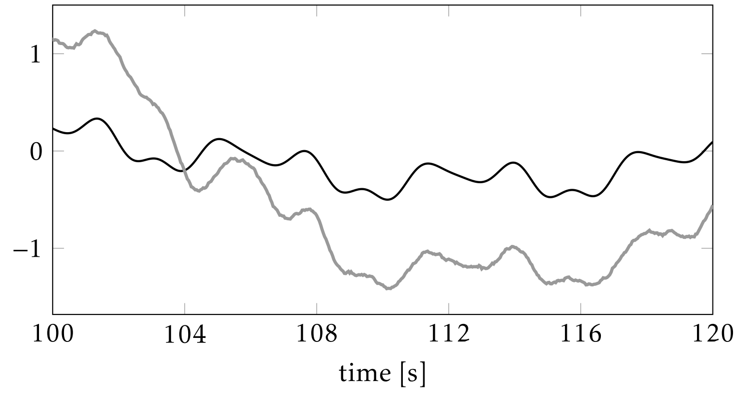


Figure 3.10: The output from the nonlinear system (3.14) *in gray and the desired dynamics from G_m* (3.15) *in black.*

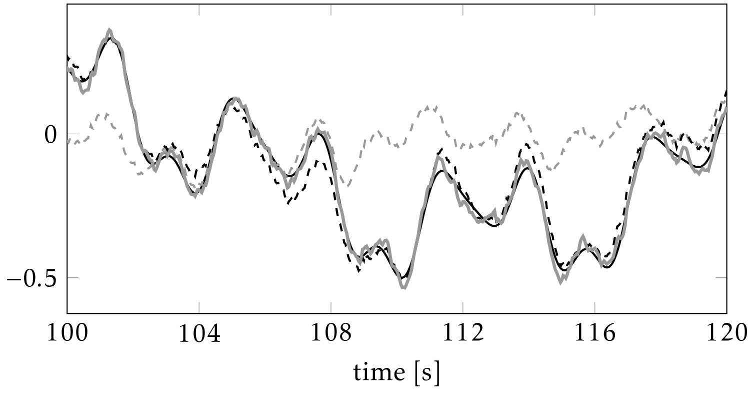


Figure 3.11: A Hirschorn postinverse applied to the system (3.14)*. The output from G_m* (3.15) *is plotted in solid black and the output when $S^{\dagger}$ was used as a preinverse in dashed black. The dashed gray line shows the output from $S^{\dagger}$ used as a postdistorter. When the postdistortion is constructed according to Figure 3.8, the result improves considerably. The solid gray line is the output from a postinverse consisting of three blocks, $1/G_m$, $S^{\dagger}$ and G_m. Note the scale difference from Figure 3.10.*

4 A stochastic approach to system inverses

In this chapter we will present what we mean by an inverse system, and a discussion on optimal inverses and the true system. The inverse models of interest here have the purpose of being used in cascade with the system itself, denoted $\mathcal{S}$, as an inverter, and a good inverse model in this setting would be one that reconstructs the original input, see Figure 4.1. The inverse can be applied at the input, making the inverse a *preinverse* $\mathcal{R}$, or after the system as a *postinverse* $\mathcal{T}$. The goal is to obtain $y_{\mathcal{R}} = p$ for a preinverse and $y_{\mathcal{T}} = u$ for a postinverse.

4.1 Definitions and notation

Here, notation and some definitions will be introduced. The terms preinverse and postinverse have already been used, but will be defined more clearly.

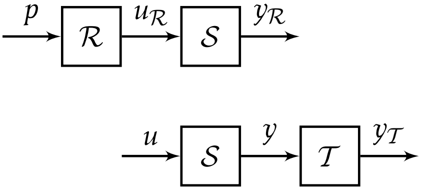


Figure 4.1: *The intended use of the estimated inverses. The top figure shows a preinverse $\mathcal{R}$, where the inverse is applied before the system $\mathcal{S}$. The lower shows a postinverse $\mathcal{T}$, where the order of the system and the inverse is reversed. For a preinverse, the preinverted signal $u_{\mathcal{R}}$ should make the output $y_{\mathcal{R}}$ the same as the reference p, $y_{\mathcal{R}} = p$. For a postinverse, the output $y_{\mathcal{T}}$ should be altered to be the same as the input u, $y_{\mathcal{T}} = u$.*

4.1.1 Signals

In this thesis both deterministic and stochastic signals will be used. When working with stochastic signals, stationarity is assumed, i.e., that the signals have the same properties regardless of the absolute time. With some abuse of notation, the use of the signals and whether they are deterministic or stochastic, will be given by the context.

The general notation will be that an input is denoted u, an output y, measurement noise v and process noise w. The reference signal will be denoted p. There is a dependency on time t when necessary, that is, $u(t)$ is the value of the input signal u at time instant t.

The whole signal x is denoted

$$X = \Big(x(t)\Big)_{t=-\infty}^{\infty} \tag{4.1}$$

and past measurements, up to the current time t,

$$X_t = \Big(x(t-k)\Big)_{k=0}^{\infty}. \tag{4.2}$$

Sometimes it will be assumed that the signal is equal to zero before time $t = 0$, which reduces the limits above to $X_t = \Big(x(t-k)\Big)_{k=0}^{t}$.

4.1.2 Systems

System $\mathcal{S}$ The system to be inverted is denoted $\mathcal{S}$. This can be a linear system, a nonlinear system, a static system, a dynamic system, or a combination thereof. When there is a need to stress the kind of system, a subscript can be used. In this way, $\mathcal{S}_f$ denotes a static nonlinearity f.

Preinverse $\mathcal{R}$ **and Postinverse** $\mathcal{T}$ An optimal preinverse is denoted $\mathcal{R}$, where the goal is to minimize the difference between the reference P and the predistorted output $\mathcal{S}(\mathcal{R}(P))$,

$$\ell\left(P - \mathcal{S}(\mathcal{R}(P))\right). \tag{4.3}$$

For a postinverse the goal is to minimize the difference between the input U and the postdistorted output $\mathcal{T}(\mathcal{S}(U))$,

$$\ell\left(U - \mathcal{T}(\mathcal{S}(U))\right), \tag{4.4}$$

where $\mathcal{T}$ is an optimal postinverse. ℓ is an arbitrary non-negative, real-valued function to be chosen by the user, and will be discussed below, in Section 4.2. Different notations will be used for postinverse and preinverse to stress that they might not be the same.

It can be helpful to remember the order by noting that $\mathcal{R}$ $\mathcal{S}$ $\mathcal{T}$ are in alphabetical order. Another memory rule is to see that p$\mathcal{R}$e contains the letter $\mathcal{R}$ and pos$\mathcal{T}$ contains the letter $\mathcal{T}$.

Inverse $\mathcal{S}^{-1}$ For the true inverse system, when it exists, the conventional notation $\mathcal{S}^{-1}$ will be used. Then $\mathcal{S}(\mathcal{S}^{-1}(x)) = \mathcal{S}^{-1}(\mathcal{S}(x)) = x$.

Commutativity In the case where an exact inverse does not exist, the system and its approximate inverse do not necessarily commute, so $\mathcal{R} \neq \mathcal{S}^{-1}$, $\mathcal{T} \neq \mathcal{S}^{-1}$ and $\mathcal{R} \neq \mathcal{T}$ in the general case.

Subscripts will be used when necessary to emphasize the original system, such that $\mathcal{T}_f$ is the optimal postinverse of a static nonlinearity f. Capital letters denote a model of the system or the inverse. That is, T is a particular model of a postinverse and $\hat{T}$ is the actual estimated postinverse. In the same way, a postinverse of a static nonlinearity f can be denoted T_f, with the estimated model $\hat{T}_f$.

4.1.3 Noise

In this thesis, cases with and without noise will be considered. The noise-free case is interesting since it answers questions like *What is the best possible result we could obtain?* and *Is there a difference in performance between different methods even under perfect conditions?* However, all applications in the real world are subject to some levels of noise, disturbances and uncertainties.

It is often assumed that noise corrupts the outcome of a system, such that not all behavior depends solely on the known input signal and the often unknown system. It might be possible to say something about the characteristics of the noise, such as the frequency content or where it enters the system, but it cannot be predicted exactly.

The noise is sometimes assumed to be additive, (Gaussian) noise that corrupts the output directly, and is then called measurement noise. It can also be assumed to enter the system earlier and it is then called process noise. If the noise enters along with the input, it is called input noise. For a linear system, these noise sources are interchangeable, which means that they can be pulled through the system and be assumed to enter at any other point (since linear systems are additive, (2.1)), perhaps with a different coloring. For a nonlinear system, this is not the case.

In Ljung [1999], it is shown how noise affects the estimation, but there the noise is assumed to act on the output. Noise on the input can be handled by, for example, input error (IE) methods, where the input is assumed to be corrupted by additive noise [Åström and Eykhoff, 1971]. This assumes that the noise signals are independent. In this thesis, when the goal is to use the inverse, the role of the input and the output in some sense change, and the inverse can be seen as a case with a perfectly known output u and an input y corrupted by noise. However, it is important to remember that the noise and y are dependent in this case.

To put the noise in a more formal setting, the following definition will be used. White noise consists of *independent, identically distributed* (i.i.d.) variables. Independency implies that two samples at two different time points are not correlated. Identically distributed means that the distribution of the noise does not change over time.

When there is no noise present, the system is described by $\mathcal{S}(U)$, and with noise present by $\mathcal{S}(U, W)$. Here, each element in W can be a vector. This is the most general formulation where the noise can enter anywhere. For an input error formulation, the output would be described by $Y = \mathcal{S}(U + W)$ and for output error, the output is described by $Y = \mathcal{S}(U) + W$.

Presence of noise in applications

In real applications, the noise levels could vary and be of considerable size. In mobile phone base stations and other costly parts of the communication chain, it could be a valid assumption that the noise can be seen as negligible. However, in cheaper parts such as mass-produced mobile phones this assumption is probably not true.

Sensors come in all shapes, sizes and price ranges, and the quality of the sensors (and the noise levels) vary. Also in loudspeaker linearization, the price can be assumed to be closely connected to the noise levels. The high-end loudspeakers use more expensive components, whereas the mass-produced loudspeakers used in tablets and smartphones use cheaper components.

4.2 The optimal preinverse and postinverse

To be able to say something is optimal, the goal needs to be determined. One common measure is the (nonlinear) *mean square error* (MSE) estimator,

$$\hat{g} = \arg\min_{g} E\left[(y(t) - g(U_t))^2\right] \tag{4.5}$$

which can be used to find the function that minimizes the difference between the output and a filtered input in the forward case. The function g that is the argument here is a function that takes U_t as input and gives an estimate of $y(t)$ as output. Often, the function g is of a set model structure and the minimization is done over the parameters θ in the given structure, but it could also contain different nonlinearities or other structures. The function g can be defined in many ways, which will be discussed later in the thesis.

The goal is to find the MSE (4.5), but this is a theoretical measure using expected values and infinite series of measurements and cannot be applied directly to measured data. Instead, the corresponding quadratic cost function that approximates MSE, or a finite data MSE, will be used

$$M' = \frac{1}{N}\sum_{t=1}^{N}(y(t) - g(u(t)))^2 . \tag{4.6}$$

In this case, the function $\ell(\,\cdot\,)$ in (4.3) and (4.4) is defined by

$$\ell(X) = \frac{1}{N}\sum_{t=1}^{N} x(t)^2.$$

There are of course a number of possible measures to evaluate the models and inverses. Depending on the choice, the results will be different, and it is relevant to discuss what the purpose of the inverse should be. To minimize the total error, the mean square error (4.6) could be used (depending on how the error is defined). The MSE is the L2-norm, but other norms are possible choices, such as the L1-norm. Another is to minimize the maximum error such that all errors are made fairly small, but there is no incentive to minimize the individual errors. The *min-max*-optimization is one such method which minimizes the largest (maximal) error. The choice of minimization criterion is up to the user. However, this should reflect the purpose of the inverse to be used.

In this section, there is noise corrupting the output, that is, the output is described by $Y = \mathcal{S}(U, W)$. The models considered are simulation models, meaning that the output is based solely on the input at past and present time instances.

4.2.1 Optimal forward model

In forward (standard) model estimation and system identification, the goal is to minimize the difference between the measured output and the modeled output as in (4.6). For forward models, the calculations are well-known, but they are presented here for the sake of completeness.

In the standard setup, the input u to the system is completely known, the output y from the system is measured, and the noise w is unknown but can be seen through the effects it has on the output. The goal is to estimate a model of the system.

In the forward modeling, the goal is to find the minimum mean-square error estimator (4.5). Let the conditional expected value $E\left[y(t)|U_t\right]$ be denoted

$$g_0(U_t) = E\left[y(t)|U_t\right], \tag{4.7}$$

and consider also an arbitrary estimator $g(U_t)$. The goal is to find the minimizing argument

$$\hat{g} = \arg\min_{g} E\left[(y(t) - g(U_t))^2\right]. \tag{4.8}$$

Then

$$\begin{aligned} E\left[(y(t) - g(U_t))^2\right] &= E\left[\Big((y(t) - g_0(U_t)) + (g_0(U_t) - g(U_t))\Big)^2\right] \\ &= E\left[(y(t) - g_0(U_t))^2\right] + 2E\Big[(y(t) - g_0(U_t))(g_0(U_t) - g(U_t))\Big] \\ &\qquad + E\left[(g_0(U_t) - g(U_t))^2\right] \\ &= E\left[(y(t) - g_0(U_t))^2\right] + E\left[(g_0(U_t) - g(U_t))^2\right] \\ &\geq E\left[(y(t) - g_0(U_t))^2\right] \end{aligned} \tag{4.9}$$

using

$$\begin{aligned} E\left[(y(t) - g_0(U_t))(g_0(U_t) - g(U_t))\right] &= E_U\left[E\left[(y(t) - g_0(U_t))(g_0(U_t) - g(U_t))\right]|U_t\right] \\ &= E_U\left[(g_0(U_t) - g(U_t))E\left[y(t) - g_0(U_t)|U_t\right]\right] \end{aligned}$$

and

$$E\left[y(t) - g_0(U_t) | U_t\right] = E\left[y(t)|U_t\right] - g_0(U_t) = 0.$$

The inequality in (4.9) comes from the fact that the first term on the row above does not depend on the minimizing argument g, and the best that can be done with the second term is to set $\mathcal{R} = g(U) = g_0(U)$.

4.2.2 Optimal postinverse

For a postinverse, the goal is instead to look at what can be done to the output of the system, to recover the original input. The input to and the noisy output from the system are unchanged by a postinverse.

Now, to find the optimal postinverse, the MSE is defined by

$$\hat{f} = \arg\min_{f} E\left[(u(t) - f(Y_t))^2\right] \tag{4.10}$$

which can be used to find the function that minimizes the difference between the postinverted output and the input. The corresponding finite-data mean square error MSE is

$$M = \frac{1}{N}\sum_{t=1}^{N}(u(t) - y_{\mathcal{T}}(t))^2. \tag{4.11}$$

Let the conditional expected value $E\left[u(t)|Y_t\right]$ be denoted

$$f_0(Y) = E\left[u(t)|Y_t\right], \tag{4.12}$$

and consider an arbitrary estimator $f(Y_t)$.

Ideally, a postinverse should recover the original input to the system. This will of course not be the case since there is noise present, but it is interesting to see what can be achieved. It turns out that f_0 is the minimum mean-square error estimator since

$$\begin{aligned}
E\left[(u(t) - f(Y_t))^2\right] &= E\left[\left((u(t) - f_0(Y_t)) + (f_0(Y_t) - f(Y_t))\right)^2\right] \\
&= E\left[(u(t) - f_0(Y_t))^2\right] + 2E\Big[(u(t) - f_0(Y_t))(f_0(Y_t) - f(Y_t))\Big] + \\
&\qquad E\left[(f_0(Y_t) - f(Y_t))^2\right] \\
&= E\left[(u(t) - f_0(Y_t))^2\right] + E\left[(f_0(Y_t) - f(Y_t))^2\right] \\
&\geq E\left[(u(t) - f_0(Y_t))^2\right]
\end{aligned} \tag{4.13}$$

using

$$\begin{aligned}
E\left[(u(t) - f_0(Y_t))(f_0(Y_t) - f(Y_t))\right] &= E_Y\left[E\left[(u(t) - f_0(Y_t))(f_0(Y_t) - f(Y_t))\right]|Y_t\right] \\
&= E_Y\left[(f_0(Y_t) - f(Y_t))E\left[u(t) - f_0(Y_t)|Y_t\right]\right]
\end{aligned}$$

and

$$E\left[u(t) - f_0(Y_t)|Y_t\right] = E\left[u(t)|Y_t\right] - f_0(Y_t) = 0.$$

In a similar way as for the forward case, the best postinverse is $\mathcal{T} = f_0(Y) = E\left[u(t)|Y_t\right]$.

4.2.3 Optimal preinverse

For the optimal preinverse, the input signal should be adapted in such a way that the output from the system will be as similar to the original input as possible. This is different compared to the earlier cases, since the actual input to the system will be changed.

To find the optimal preinverse, the goal is to minimize the difference between the reference p and the preinverted output

$$Y_{\mathcal{R}} = \mathcal{S}(R(P), W), \tag{4.14}$$

that is

$$\mathcal{R} = \arg\min_{R} E\left[(p(t) - y_{\mathcal{R}}(t))^2\right].$$

For the optimal forward model and the optimal postinverse, the measured data U_t and Y_t (corrupted by noise) contain all the information needed. Or rather, the data and their stochastic properties contain the information needed. In those two cases, the goal is to find the signal or system by determining what is interesting and what is noise, and how to use the degrees of freedom in the model to obtain the best performance.

For the optimal preinverse, the setup is a very different. Here, the input signal $U_{\mathcal{R}}$ is determined by the user, and the goal is to change this signal, which will then pass through the system (with noise W) with the overall goal that the output from the system should be similar to the original input.

Hence, the data with corresponding properties are not enough, since the preinverse could change the properties of the predistorted input significantly. It is therefore not sure that the experiments have captured the desired properties of the system with a different input signal. In general in system identification, a model is only valid for inputs with the same properties as the one used for the estimation, since different input signals could excite different parts of the system. Since the original input and the predistorted input could have significantly different properties, the two inputs could excite different properties in the system, and an inverse based on measurements using one input signal is not necessarily useful for the other. The same reasoning is valid for the inverse – an inverse estimated for one input signal could perform badly when used with a different input signal.

The concepts of *domain* and *range* of a signal is also coupled to this. For a forward model the domain is the same and the range of the model should be the same as for the system. For a postinverse, the goal is to reconstruct the input, and the domain of the inverse is the range of the system, and vice versa. However, a preinverse has the same domain as the system, but the range of the inverse could be outside of the domain of the system. Imagine a system that makes the output much smaller in amplitude; then the preinverse should amplify this signal, and there are no guarantees that this new input is within the domain of the system.

4.3 Is the exact inverse best?

Here, some examples to illustrate that the answer is *No* will be presented. The mean square error (4.11) is used as a criterion.

4.3.1 Difference between preinverse and postinverse

The following two examples illustrate that the best preinverse and the best postinverse are not necessarily the same.

Example 4.1: A signal in noise

Consider a system with process noise and no measurement noise. This system is described by

$$y = f(u + w)$$

where u is the input and w is the noise. The inverse $f^{-1}(\cdot)$ exists and $f^{-1}(f(x)) = f(f^{-1}(x)) = x$. This means that the true inverse applied to a noisy output is

$$f^{-1}(f(u + w)) = u + w.$$

That is, an exact postinverse would recreate the sum of the input and the process noise. However, the goal is not to recreate this sum but only the input u, so using another postinverse T_f here seems reasonable if that inverse model is better at predicting the original input u.

Using a specific function, with $f(u) = u$, the exact inverse is $f^{-1}(y) = y$. If the input and noise are band-limited and in separate frequency bands (for example a slowly varying input signal and a high-frequent noise signal), the optimal postinverse would be an ideal (low-pass) filter such that the noise is suppressed while the original input passes unchanged and the exact input can be regained. Here, the exact inverse f^{-1} will not affect the output at all, but $y_T = y = u + w$. The same reasoning works with other characteristics of the input signal and the noise, as long as they can be separated. However, a preinverse in this case will act on the input signal (with no noise present) so using this optimal postfilter as a prefilter will have no effect on the overall behavior of the system.

In the example above, it is clear that the preinverse and postinverse cannot be interchanged without further consideration. In the example below, some theoretical results are presented to support this conclusion.

Example 4.2: Analytical calculations of a cubic function

Consider a system that is described by a cubic function

$$f(u) = bu^3$$

where u is the input and $y = f(u)$ is the output. There is process noise w present, that enters along with the input

$$y = b(u + w)^3. \tag{4.15}$$

The input signal u and the noise signal w are independent white Gaussian noises with zero mean and variances σ_u^2 and σ_w^2, respectively.

A model that is the inverse of the true system will be used as a preinverse, to evaluate what can be achieved if the true structure of the system is known. The structure of the inverse is then a cubic root with a scaling parameter. For a postinverse (with subscript T), the input estimate is

$$y_T = \theta_T \sqrt[3]{y}$$

intended to be applied at the output of the system. For a preinverse (with subscript R) the predistorted signal is

$$u_R = \theta_R \sqrt[3]{u}$$

and the corresponding predistorted output is

$$y_R = b(u_R + w)^3.$$

When the cubic root $\sqrt[3]{\cdot}$ is used, the real branch of the root is assumed.

The cost function is the mean square error (4.11), see also Chapter 2, where the goal is to minimize the difference between the original input u and the preinverted or postinverted output, respectively. The inverted output is y_p with $y_p = y_T$ or $y_p = y_R$ depending on the case evaluated, and

$$\hat{\theta} = \arg\min_{\theta} E[(y_p - u)^2]. \tag{4.16}$$

The true inverse is $u_0 = \theta_0 \sqrt[3]{s}$ with $s = \{u, y\}$ for a preinverse and postinverse, respectively, and

$$\theta_0 = \frac{1}{b^{1/3}}. \tag{4.17}$$

To find the minimizing argument of (4.16), take the derivative and set it to zero

$$\frac{\partial}{\partial\theta} E[(y_p(\theta) - u)^2] = 0, \tag{4.18}$$

where y_p depends on θ. Since linear functions commute, it is possible to change the order of the expected value and the differentiation, and

$$\theta_T = \frac{1}{b^{1/3}} \frac{\sigma_u^2}{\sigma_u^2 + \sigma_w^2} = \theta_0 \frac{\sigma_u^2}{\sigma_u^2 + \sigma_w^2}$$

is the optimal value of θ for a postinverse. The optimal postinverse takes the noise variance into account, so only knowing the true value of b does not really help in constructing the optimal inverse. For the noise-free case ($\sigma_w^2 = 0$), the true inverse is optimal, and if the noise variances are known the optimal inverse model can be obtained.

The preinverse leads to a more complicated expression, and the equation to be solved is

$$\theta_R^5 6b^2\sigma_u^2 + \theta_R^3 60b^2\sigma_w^2 E[u^{4/3}] - \theta_R^2 6b\sigma_u^2 + \theta_R 90b^2\sigma_w^4 E[u^{2/3}] - 6b\sigma_w^2 E[u^{4/3}] = 0. \tag{4.19}$$

This is not a trivial equation to solve, but it is easy to check if the true inverse is one solution by inserting θ_0 in the equation. Using (4.17) in (4.19) leads to

$$\frac{\partial}{\partial\theta}E[(y_R - u)^2|\theta = \theta_0] = 54b\sigma_w^2 E[u^{4/3}] + 90b^{5/3}\sigma_w^4 E[u^{2/3}] > 0$$

where the last step comes from

$$E[u^{4/3}] > 0 \quad \text{and} \quad E[u^{2/3}] > 0,$$

and implicitly that $b \neq 0$ and $\sigma_w^2 \neq 0$. That $E[u^{4/3}] > 0$ and $E[u^{2/3}] > 0$ hold is clear when it is considered that the real branch $u^{1/3}$ is well defined along the real axis, and the expected value of a square of this stochastic variable ($(u^{1/3})^2$) will always be positive. The same holds for $(u^{1/3})^4$.

For the noise-free case, where $\sigma_w^2 = 0$, the calculations are simplified and

$$\theta_R = \frac{1}{b^{1/3}} = \theta_0,$$

just like for the postinverse.

This example shows that even in a simple case like the cubic nonlinearity, the true inverse is not the optimal inverse neither for a preinverse nor a postinverse when there is noise present. For the postinverse, knowing the true forward system (or inverse system) as well as the noise variances will lead to the optimal inverse, by using a scale factor in this case. For a preinverse the connection between the true inverse and the optimal one is not as straightforward.

In the examples above it can clearly be seen that the optimal preinverse and postinverse are not necessarily the same, and that the optimal inverse could be something else than the true inverse of the system.

4.3.2 Choice of preinverse structures

In the following two examples the system itself is a static nonlinearity, and two different types of preinverses will be applied.

Example 4.3: Piecewise linear system

In this example, look at a piecewise linear system where the slope is 1 between $[-a \quad a]$ and k outside, such that there is a jump in the derivative at a and $-a$, respectively. The function itself is continuous. Here, $a = 1$ and $k = 10$. The input u is uniform in $[-a \quad a] = [-1 \quad 1]$ and the process noise w is uniform in $[-a/10 \quad a/10] = [-0.1 \quad 0.1]$, see Figure 4.2 (a). The goal is to apply a preinverse.

If the true inverse is used as a preinverse, the input will not be affected at all since the input is within the linear range, and the precompensated output will be the same as the original output. If some other inverse is used the nonlinear behavior can be reduced, for example by simply saturating the input signal at $0.9a$, such that when it enters the nonlinearity with the added process noise, it is still within the linear region, see Figure 4.2(b). This will of course cause some

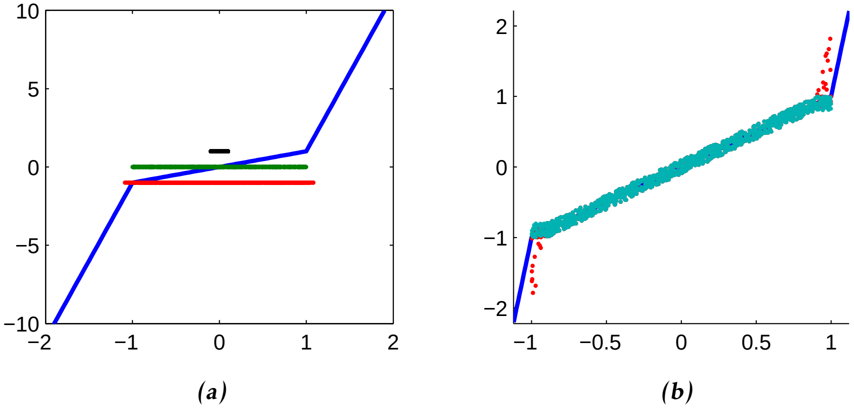


Figure 4.2: *(a) The nonlinearity in Example 4.3. The data distributions are also plotted, the input u (green) at 0, the noise w (black) at 1 and the noisy input $u + w$ (red) at -1. (b) The nonlinearity (blue solid line) and the output/predistorted output using the true preinverse (red dots). These will be the same since the true inverse does not affect the signal. The output using a saturating predistortion is plotted in light blue.*

Table 4.1: *Squared errors of the piece-wise linear example in Example 4.3.*

	Preinverse	
y	True	Saturation
7.97	7.97	3.77

distortion, but it will be smaller than in the original output. Other smoother functions that are not as harsh as a saturation, such as an arctangent function, could also be used that reduce the influence of the noise. This will not be further investigated here. The squared errors $\sum_{t=1}^{N} (y_{\mathcal{R}}(t) - p(t))^2$ are shown in Table 4.1 for the output, the true preinverse and a saturating preinverse.

The example above shows that a nonlinear function that is not the inverse to the true system can help improve the performance, when used as a preinverse. In the example below, a dynamical system is instead used in addition to a nonlinearity to improve the preinversion behavior of a purely nonlinear system.

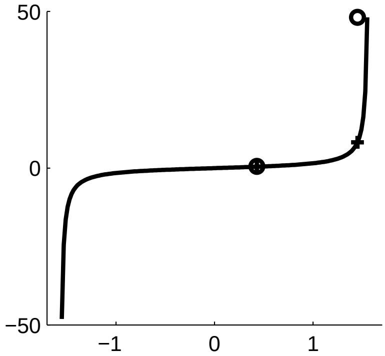


Figure 4.3: The tangent function $y = f(u, w) = \tan(u + w)$. The solid line shows the tangent function without process noise ($y = f(u) = \tan(u)$). The plus sign and the circles show the effect of process noise $w = 0.1$ for two different input values. The plus signs are the values without noise and the circles are with process noise. It is clear that the same noise level will have different effects depending on the input amplitude.

Example 4.4: A tangent function

In this example, the nonlinear system is a tangent function with input noise, $y = f(u, w) = \tan(u + w)$. The derivative of a tangent function is small close to zero and grows to infinity close to the edges of the function domain $]-\frac{\pi}{2}, \frac{\pi}{2}[$. For an input that is close to the edge, adding a bit of noise can have a big effect on the function output, see Figure 4.3.

Consider an input that is a multisine with an overshoot, see Figure 4.4(a), with $u_{\text{max}} = 1.18$. There is process noise w present, with w uniform, $w \in [-0.39, 0.39]$, such that $u + w \in \,]-\frac{\pi}{2}, \frac{\pi}{2}[$ and the system is invertible. The input to and the output from the system are shown in 4.4(b). The true inverse f^{-1} to the nonlinear function f is an arctangent function.

This true inverse $R_0 = f^{-1}$ will be compared to an inverse R_{H} where the nonlinearity is combined with a linear filter in a Hammerstein structure. In a Hammerstein structure, the static nonlinearity is followed by a linear dynamic block, see Figure 4.8 in Section 4.4 where block-oriented systems will be further discussed. The nonlinearity is the true inverse of the system (i.e., the same as the inverse R_0) and the linear dynamics is $G(z) = \frac{0.04603}{z-0.954}$. The two blocks in the Hammerstein inverse R_{H} were also swapped to create a Wiener structure preinverse R_{W}. The sample time is 0.1 seconds and the number of data points $N = 10\,000$.

The results are shown in Figures 4.5 and 4.6. Here only the Hammerstein inverse R_{H} is shown, which significantly reduces the spikes in the predistorted output signal. For the sake of comparison, the noise realization is the same for the three predistorters in the evaluation. The MSE values (4.11) of the three predistorters R_0, R_{H} and R_{W} and the original output y are presented in Table 4.2 and a box plot of Monte Carlo simulations in Figure 4.7.

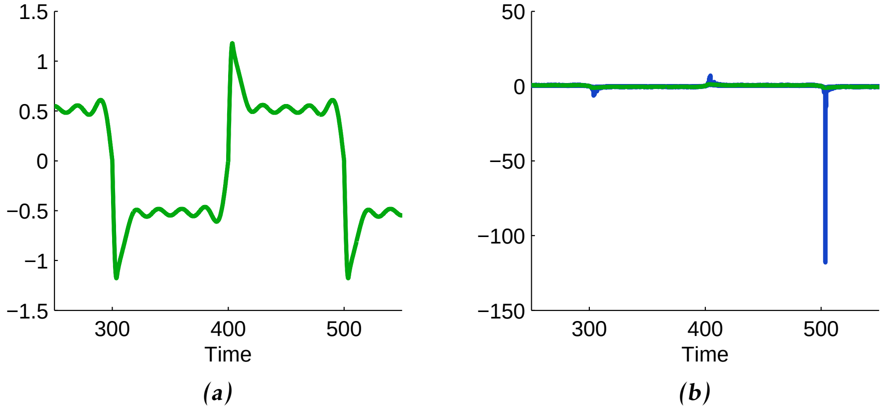


Figure 4.4: *(a) The input to the tangent function nonlinearity. (b) The input u (green) and the output y (blue) signals.*

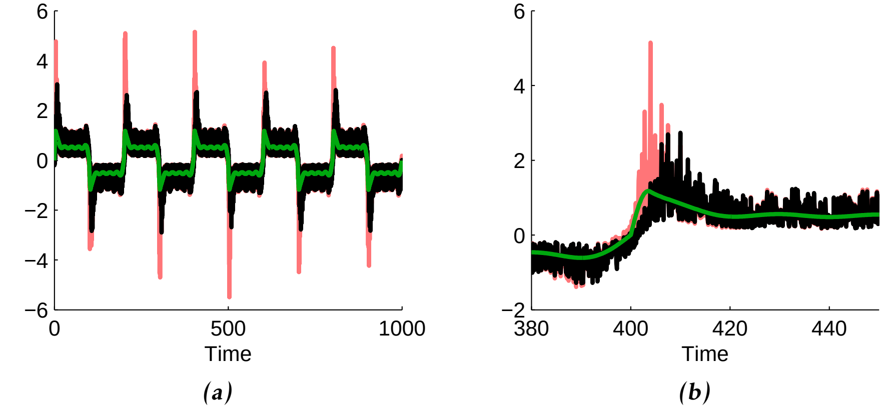


Figure 4.5: *(a) The pink line shows the predistorted output when the true inverse R_0 is applied, and the black line when a different inverse R_H is applied. The green line is the original input. As can be seen, using the true inverse R_0 results in larger spikes than the alternative Hammerstein inverse R_H. However, since the noise is significant, both predistorted outputs are rather noisy. (b) A zoomed in copy of (a) to better show the difference between the signals.*

Table 4.2: MSE (4.11) *of the tangent example in Example 4.4.*

y	True	Hammerstein	Wiener
	R_0	R_H	R_W
2.6707	0.1429	0.1273	0.1359

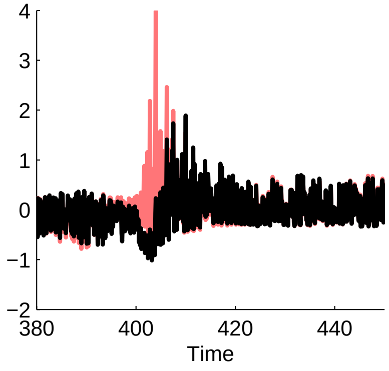


__Figure 4.6:__ The pink line shows the predistorted output error when the true inverse R_0 is applied, and the black line when a Hammerstein structured inverse R_H is applied. That is, this plot shows the deviations from the green line in Figure 4.5.

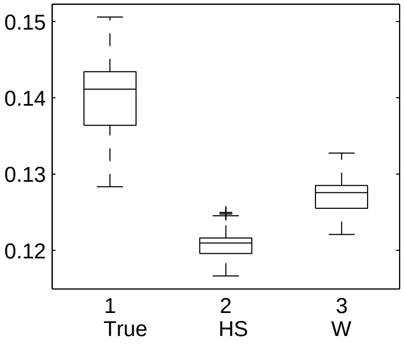


__Figure 4.7:__ A boxplot of the MSE of 40 Monte Carlo simulations of the system in Example 4.4. The left box represents the true inverse R_0, the middle one the Hammerstein inverse R_H and the right box is the Wiener inverse R_W.

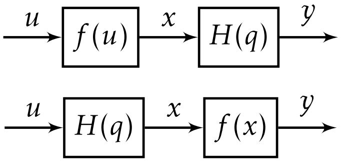


Figure 4.8: *Block-oriented systems consist of linear time-invariant dynamic systems $H(q)$ and static nonlinearities $f(\cdot)$. The top figure shows a Hammerstein system, where the nonlinearity is at the input, and the lower a Wiener system with a nonlinearity at the output.*

It is clear that using a predistorter reduces the distortion and the MSE but also that the structure of the predistorter affects the results. The Hammerstein inverse R_H performs best, followed by the Wiener inverse R_W. Both block-oriented inverses outperform the true inverse R_0. Hence, the true inverse, a pure nonlinearity, is not the best structure in this case, since it is beneficial to add a dynamic filter to the preinverse.

The examples above show that different approaches can be used to improve the performance of a preinverse or a postinverse, and that the optimal inverse is not necessarily the true inverse of the system. A common denominator in these examples is that there is noise present. Different structures (nonlinearities, dynamics or a combination thereof) can be added to, or used instead of, the true inverse system to improve the performance.

These examples do not tell us how to obtain a better inverse, but they do illustrate that it can be beneficial to explore different structures of the inverse, even if the structure of the system itself is known.

4.4 A background to linear approximations of block-oriented systems

Nonlinear systems can be challenging to model but one common way is to use block-oriented models. These are built-up by LTI systems and static nonlinearities. This is a reasonable assumption when there is, for example, a nonlinear actuator due to saturation in an otherwise linear control application. In this section, some background on linear approximations of block-oriented systems is presented which will be used in the derivation of theory presented in Section 5.3.2.

A Hammerstein system consists of a static nonlinear system followed by a linear dynamic system and in a Wiener system, the static nonlinearity is at the output of the linear dynamics, see Figure 4.8. The combination of the two, with a static nonlinearity at both sides of the linear dynamics, is called a Hammerstein-Wiener system. One way to broaden the use of the Hammerstein model is to use a more general *parallel* Hammerstein model, with multiple Hammerstein models in parallel branches [Schoukens et al., 2011].

4.4.1 Linear models of nonlinear systems

It can often be practical to work with a linear model instead of a nonlinear one. This can be done by either linearizing a nonlinear model via differentiation, or by fitting a linear model directly to measured input-output data. Here, the approach of finding a linear second-order equivalent that in some sense captures part of the behavior of the nonlinear system will be used. This can be useful for example in the PA predistortion case where the predistorter should be implemented in hardware. A reduction in complexity is reflected in a smaller chip area and a lower power consumption, so if a linear model can perform well, this is beneficial.

The concept of constructing linear models of nonlinear systems, often by using an LTI system with certain required properties, such as stability and causality, has been looked at before. The term *LTI second order equivalent* (LTI-SOE) here denotes the optimal stable and causal LTI system that approximates a nonlinear system in a mean-square error sense (4.6). The term *second order equivalence* refers to the property that the true, nonlinear system and the LTI-SOE will be equivalent if second order properties of inputs, outputs and model residuals are considered [Enqvist, 2005, Enqvist and Ljung, 2005, Ljung, 2001]. Similar concepts as the LTI-SOE are used under different names, such as the noncausal Wiener filter [Gardner, 1990], related dynamic system and *best linear approximation* (BLA) [Pintelon and Schoukens, 2012]. When the BLA is stable and causal, it equals the LTI-SOE [Pintelon and Schoukens, 2012]. The concepts of related dynamic systems and BLA, and the estimation in the frequency domain are discussed in Schoukens et al. [1998] and Schoukens et al. [2005]. A deterministic approach is investigated in Mäkilä and Partington [2003], where differentiation is used to obtain an LTI approximation.

It is well-known that the stable model G_0 that minimizes the mean-square error,

$$E\left[(y(t) - G(q)u(t))^2\right], \tag{4.20}$$

can be written

$$G_0 = \frac{\Phi_{yu}(z)}{\Phi_u(z)}, \tag{4.21}$$

where $\Phi(z)$ is the z-spectrum of $u(t)$, see for example Gardner [1990]. The (power) spectrum is defined as

$$\Phi_u(\omega) = \sum_{\tau=-\infty}^{\infty} E\left[u(t)u(t-\tau)\right] e^{-i\tau\omega}$$

and the cross spectrum is defined as

$$\Phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} E\left[y(t)u(t-\tau)\right] e^{-i\tau\omega}.$$

The model (4.21) is by construction stable, but not necessarily causal. When the model is required to be stable and causal and of *output error* (OE) structure [Ljung, 1999], it is called an *output error linear time-invariant second or-*

der equivalent (OE-LTI-SOE) and sometimes this model coincides with (4.21) [Enqvist and Ljung, 2005, Corollary 7]. Additive output noise, $y(t) = G(q)u(t) + e(t)$ is assumed in the OE structure. An introduction and overview of optimal linear models of nonlinear systems can be found in Enqvist [2005] or Pintelon and Schoukens [2012].

The models are defined by minimizing the mean-square error. A model of the forward behavior can be found by minimizing

$$E\left[(y(t) - G_f(q,\theta)u(t))^2\right], \tag{4.22}$$

and the stable model that minimizes this will be denoted $G_{0,f}$ and called a noncausal LTI-SOE. Here only cases where the inverted estimated forward model

$$G_{0,fi}(q,\theta) \triangleq 1/G_{0,f}(q,\theta) \tag{4.23}$$

is also stable will be looked at.

Here, the investigation concerns how the estimation of inverse systems can be done when the goal is to construct a linear approximation of a nonlinear system. In a similar way as the optimal forward model was defined in (4.22), an optimal inverse model is defined according to the following definition.

Definition 4.1. The term noncausal *inverse LTI-SOE* (I-LTI-SOE) will be used for models obtained by minimizing the MSE criterion

$$E\left[(u(t) - G_i(q,\theta)y(t))^2\right], \tag{4.24}$$

and this stable inverse model will be denoted $G_{0,i}$.

It follows from (4.21) that the optimal model that minimizes (4.24) is

$$G_{0,i} = \frac{\Phi_{uy}(z)}{\Phi_y(z)}, \tag{4.25}$$

where the input $u(t)$ and the output $y(t)$ are switched.

Hence, the notation is as follows. $G_{0,f}$ is used for the noncausal LTI-SOE, $G_{0,fi}$ for the inverted $G_{0,f}$, such as it will be used here, and $G_{0,i}$ is used for the noncausal I-LTI-SOE (4.25). For notational convenience, the arguments of the models will sometimes be omitted.

4.4.2 Application to experimental data

The discussion above concerns theoretical definitions of the optimal models, which are marked by a subscript 0. However, when measured input-output data are available, the ideas can be used to estimate models. A model structure then has to be chosen for the models to be estimated. To make comparisons fair, the inverse of the forward model is restricted to have the same structure as the inverse model, such that

$$G_{fi}(q,\theta_f) = G_i(q,\theta_f) \tag{4.26}$$

for a specified model structure, where θ_f are the parameter values that minimize the forward MSE (4.22) in a given model structure. The inverted estimated optimal forward model G_{fi} thus has the same structure as the estimated optimal inverse model G_i. Note also that there are no assumptions on the invertibility of the nonlinearity, since a stable linear approximation with a stable inverse is estimated.

The estimated inverses G_{fi} and G_i can be applied as preinverse or postinverse. They can be applied at the output as

$$y_{f\mathcal{T}}(t) = G_{fi}(q, \theta_f)\, y(t) \quad \text{or} \quad y_{i\mathcal{T}}(t) = G_i(q, \theta_i)\, y(t). \tag{4.27}$$

The inverses can also be used as preinverses,

$$u_{f\mathcal{R}}(t) = G_{fi}(q, \theta_f)\, u(t) \quad \text{or} \quad u_{i\mathcal{R}}(t) = G_i(q, \theta_i)\, u(t) \tag{4.28}$$

and then the signals $u_{f\mathcal{R}}(t)$ or $u_{i\mathcal{R}}(t)$ are passed through the nonlinear system. In general, this is an application-dependent choice that the user cannot affect. For example, in the sensor calibration application, the user does not have access to the input side, and only a postinverse is possible to use.

Another point that is important to discuss for the estimation of inverse systems is the scaling of the signals. The amplitude of the output of a precompensated versus a postcompensated system can be considerably different. To reduce the effects of the signal amplitudes, normalization can be used.

In the next chapter, estimation methods for inverse models will be discussed.

5

Estimation of a system inverse

In Chapter 4, what is meant by a preinverse and a postinverse was discussed, and what the optimal inverse is. However, nothing was said about how the inverses should be obtained. One way is to use the method of *system identification*, described in Chapter 2, where a model is based on measured data.

The estimation of system inverses is rather common, but there has not been much theoretical work on which approach is to be preferred in practical applications. In Paaso and Mämmelä [2008] and Abd-Elrady et al. [2008] two approaches have been evaluated on data. The approach that leads to better results in one paper leads to worse results in the other, leading to the conclusion that there might not be a method that is always preferred, and that further investigations are needed. However, in PA predistortion it seems to be more common to choose an estimation approach and evaluate the results, rather than to evaluate the different approaches themselves. This chapter contains some results regarding the differences between the approaches, and the goal is to improve the knowledge of the estimation of inverse systems.

In estimation, it is usually beneficial to estimate the system in the setting in which it should be used, concerning for example the choice of input and the experimental conditions [Ljung, 1999, Gevers and Ljung, 1986, Pintelon and Schoukens, 2012]. It is important to choose the input signal to capture the significant characteristics of the system. Another important topic in system identification is the choice of loss function, V in (2.5b). It should reflect the goal of the identification, and, depending on how it is chosen, different properties of the estimated model will be emphasized. Here, the goal is to make use of these degrees of freedom and the flexibility of the model to obtain an accurate input reconstruction. Parts of the contents are also presented in Jung and Enqvist [2013, 2015].

***Table 5.1:** Inputs and outputs to the identification procedure, using the different methods.*

Method	Input	Output	Requires	Model	In PA literature
A	u	y		$\mathcal{S}$	
B1	p	p	$\hat{\mathcal{S}}$	$\mathcal{R}$	DLA
B2	p	p	Experiments	$\mathcal{R}$	
C	y	u		$\mathcal{T}$	ILA

5.1 Classification of estimation methods

In system identification, the goal is to achieve a model as good as possible to explain the behavior of y by a prediction or simulation $\hat{y}(t|\theta)$, which depends on the estimated model parameters θ and the input u. This is done using measured data, usually input data $u(t)$ and output data $y(t)$, see also Chapter 2.

The inverse model is estimated with the purpose of using it in series with the system itself, as an inverter, see Figure 4.1. In this setup, the goal is to minimize the difference between the input u and the output from the cascaded systems, $y_{\mathcal{R}}$ or $y_{\mathcal{T}}$. A good model in this setting would be one that, when used in series with the original system, reconstructs the original input.

5.1.1 Method overview

There are three main approaches to the estimation of an inverse of a system $\mathcal{S}$, described in more detail below.

Method A In a first step, the forward model $\hat{\mathcal{S}}$ is estimated in the standard way, with input data u and output data y. Step two is to invert the resulting model to obtain an approximate inverse $\hat{\mathcal{S}}^{-1}$.

Method B The inverse model is estimated as a preinverse $\mathcal{R}$, in series with a model of the system $\hat{\mathcal{S}}$ (Method B1) or the system $\mathcal{S}$ itself (Method B2). For Method B1, the goal is to minimize the difference between the input u and the simulated, preinverted output $y_{\mathcal{R}}$. In Method B2 the difference between the input u and the measured output $y_{\mathcal{R}}$ is minimized iteratively with the system $\mathcal{S}$ in the loop.

Method C The identification is done in one step, by identifying the inverse directly, using input data y and output data u. This leads to a postinverse $\mathcal{T}$.

The inputs and outputs to the different approaches are summarized in Table 5.1.

Method A The identification method in the first approach, Method A, is the standard one, as described in, for example, Pintelon and Schoukens [2012] and Ljung [1999], and the inversion is discussed in Åström and Hägglund [2005] in the feedforward control application. One way to construct an inverse based on a forward model is to use Hirschorn's method, presented in Section 3.3.2. The use of feedforward control based on an inverse model of the system in the presence of plant uncertainty is discussed in Devasia [2002]. A good thing with METHOD A is that the identification uses standard methods. On the other hand, an inversion is required, and as mentioned before the model should be estimated in the setting in which it will be used which is not fulfilled here. The inverse can be used as either a preinverse or a postinverse.

METHOD B This method is developed for preinversion. An inverse estimated this way can of course be used as a postinverse, but this does not seem like a straightforward way. For METHOD B1, the quality of the inverse and the forward models are closely coupled, and multiple choices are available. Since it is often preferable to obtain a rather simple inverse model (for example in the predistorter case), this restriction can also be applied to the forward model, so that the same model structure is used for the forward and the inverse models. Another approach is to use a more complex forward model, making sure that as much as possible of the behavior of the system is captured, and then let the inverse model be less complex. The choice in the end comes down to the implementation – if the forward model has to be implemented, also this model needs to have a limited complexity.

Since the system output is obtained through simulations, the estimation of the inverse model is done with no noise present. This requires two, possibly non-convex, minimizations with the risk of obtaining local minima and the quality of the inverse clearly depends on the quality of the forward model. The preinverse problem is also harder numerically, since the signal will pass through the system. Even in the case where the both $\hat{\mathcal{S}}$ and $\mathcal{R}$ can be described by a linear regression, the cascaded system will have a more complicated structure that will make the optimization harder.

An alternative method where the inverse model is still estimated in a preinverse setting is to use the system itself in an iterative method. This method is denoted METHOD B2. The benefit of estimating the preinverse in series with the real system is that this is the actual setup in which it will be used. Also, it is possible to take the noise into consideration in a suitable way, which can improve the performance. The drawback is of course that multiple measurements are needed, as well as access to the physical device to perform experiments.

METHOD C METHOD C determines a postinverse, which can also be used as a preinverse. For METHOD C to be applicable for preinversion, it is assumed that a preinverse and a postinverse are interchangeable (commutativity), see also Section 3.2.2. An advantage with this method is that the model is estimated as an inverse, which is how it will be used. A drawback is that the measured output is used as input, which risks causing a biased estimate [Amin et al., 2012]. METHOD C can be an easier approach than METHOD A or METHOD B, since the estimation is done in one step. This makes it easy to try out different model structures and model orders, which is often needed to find a good model. Depending on the model structure, it can also be easier to find a convex formulation compared to METHOD B which is a nested problem. Also, there is no need to construct a model for the forward system that will be discarded later.

5.1.2 Predistortion application of the methods

In the second part of this thesis, the goal is to estimate a power amplifier predistorter. This section provides a discussion on how the different methods are used in predistortion. The boundary between the different methods is not always clear. For example, METHOD B1 could be seen as belonging to METHOD A, where the second estimation step is a numerical inversion. However, since it is a method designated to design a preinverse, we have separated them into different methods.

The two most common approaches in DPD applications are METHODS B1 and C. In this application, METHOD B1 is also called *direct learning architecture* (DLA) [Fritzin et al., 2011a, Abd-Elrady et al., 2008, Paaso and Mämmelä, 2008]. METHOD C is also called *indirect learning architecture* (ILA), or the postdistortion and translation method [Gilabert et al., 2005]. It is commonly used in power amplifier predistortion applications and has been evaluated in for example Abd-Elrady et al. [2008] and Paaso and Mämmelä [2008].

In power amplifier predistortion applications, METHOD C is more commonly used than METHOD B1, as investigated in Paaso and Mämmelä [2008]. This could be because METHOD C has a less complex structure and faster convergence [Chani-Cahuana et al., 2016] and this is considered a bigger benefit than the drawback that there is a risk of bias. In Paaso and Mämmelä [2008], comparisons indicate that METHOD B1 performs better, whereas METHOD C seems to perform slightly better in Abd-Elrady et al. [2008], both evaluated in simulations. Ghannouchi and Hammi [2009] state that the accuracies of pre- and postcompensation are equivalent, which we will show is not always the case.

Hussein et al. [2012] compares predistortion results of METHODS B1 and C for different noise settings, and their results show that the METHOD C is better than METHOD B1 in the noise-free case, but the opposite is true when there is noise present (that is, METHOD B1 outperforms METHOD C). The results are evaluated in a simulation study on a power amplifier. The errors in the PA modeling in METHOD B1 are used as an explanation that METHOD C performs better in the noise-free case. The deterioration of METHOD C in the presence of noise is attributed to the least-squares solution which is sensitive to noise entering in a non-standard way. The authors also stress that it is important both for both methods B1 and C to use an iterative approach with repeated measurements, since the predistortion changes the input to the PA and broadens the bandwidth of the signal. This is done by performing a reidentification of the PA after a new predistorter is produced, until the PA-DPD system converges. This iterative METHOD B1 is different from METHOD B2, which does not use a model of the PA.

In this thesis, we draw the same conclusion that it is important to reiterate the modeling approach using multiple measurements, but with a different reasoning. The errors from the PA modeling in METHOD B1 can be avoided by using METHOD B2 where the true system is used instead of a model. In Chapter 4 it was illustrated that using the noise in the construction of an inverse can improve the performance. Of course, the noise can also degrade the quality, and needs to be handled correctly. By using METHOD B2, both these points can be taken into account.

A modification to METHOD C, called *model-based indirect learning architecture* (MILA), is presented in Landin et al. [2014]. A model is created for the amplifier and a simulated output is used in the METHOD C instead of a measurement of the output, to reduce the risk of bias in the parameters due to the noise in the wrong place in the METHOD C setup. However, this also entails a second optimization and the quality of the predistorted output will depend on the structure and the quality of the PA model, just like in METHOD B1. In this classification, MILA could be interpreted as a METHOD A. The only noisy estimation performed is done on the forward system, and the second step of using the model to construct an inverse could be seen as a numerical inversion.

One of the problems of inverse system identification is that the noise enters the estimation in a nonstandard way. It is quite common to want to minimize the influence of the noise, such as in the METHOD B1 approach (where the system is replaced with a noise-free model) or in the METHOD C modification. However, as was illustrated in Chapter 4, it can be beneficial to include the noise in the inverse estimation.

The ILC-DPD is a rather new approach, presented in Chani-Cahuana et al. [2016], Schoukens et al. [2017], where the authors construct the desired input signal to the power amplifier using iterative learning control (ILC). ILC is an iterative method that can be used for repetitive tasks, where the input signal u_k to the system is changed between the iterations, see Section 3.1.2. Since the same task is performed over and over again, the output from the last iteration gives information about the performance, and instead of changing the controller, the input signal is adapted. The goal is to obtain an output that follows the reference perfectly.

This ILC is performed as a first step in the ILC-DPD. In a second step, a transfer function from original reference r to the new input signal u_P is estimated. This is an interesting approach since it looks at the intermediate signal u_P, between the predistorter and the power amplifier, which is not done in any of METHODS A-C. The benefit with this method is that the estimation in the second step can be done using standard forward methods, and that the inverse is estimated as a preinverse. This method does not perfectly fit into the classification of the methods in this chapter, since the ILC is a nonparametric method that does not result in a model of the system or its inverse directly. However, it has similarities with METHOD B2 in that it uses the system iteratively to obtain the inverse model, though this method has an additional step of getting the internal signal before constructing the preinverse.

5.1.3 In mathematical terms

In Chapter 2, the basics of system identification were covered and in Section 5.1.1, different approaches to use system identification for inverse systems were described. Here, these methods will be presented in more mathematical terms.

The parameter estimation is usually done by minimizing a cost function $V(\theta)$ such that

$$\hat{\theta} = \arg\min_{\theta} V(\theta) \tag{5.1}$$

with

$$V(\theta) = \frac{1}{N}\sum_{t=1}^{N} l\left(y(t), \hat{y}(t|\theta)\right) \tag{5.2}$$

where the model gives the output predictor $\hat{y}(t|\theta)$ and the minimization returns the parameter estimates $\hat{\theta}$ that best describe the data, within the given model structure. The function $l(y, \hat{y})$ describes how the measurements and model enter the equation.

There are many possible choices for the minimization criterion. Here, we have chosen to use the mean square error

$$\hat{\vartheta} = \arg\min_{\vartheta} V_i(\vartheta) \tag{5.3}$$

where

$$V_i(\vartheta) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon^2(t, \vartheta) \tag{5.4}$$

and

$$\varepsilon_{\mathcal{R}}(t, \vartheta) = p(t) - \mathcal{S}(\mathcal{R}(p(t), \vartheta)) \tag{5.5}$$

for a preinverse where $p(t)$ is the reference signal and

$$\varepsilon_{\mathcal{T}}(t, \vartheta) = u(t) - \mathcal{T}(\mathcal{S}(u(t), \vartheta)) \tag{5.6}$$

for a postinverse with input signal $u(t)$ to the system. That is, the goal is to minimize the difference between the original input and the pre- or postdistorted output.

For METHOD A, the above procedure corresponds to minimizing (5.2), and then use mathematical methods to invert the resulting forward model. METHOD B1 uses the same first step as METHOD A, but in a second step an inverse model is estimated using the prediction error (5.5) with $\mathcal{S}$ replaced by the corresponding model $\mathcal{M}$. METHOD B2 minimizes (5.3)-(5.5) iteratively, using repeated experiments with the system in the loop. METHOD C uses the prediction error (5.6) in the estimation to find $\hat{\vartheta}$ in (5.3)-(5.4).

5.2 Method descriptions

The methods described here will be evaluated on data later in Chapters 6 and 7. In this section, a step-by-step description is presented with the goal of estimating an inverse.

The system identification procedure will be used repeatedly in the method descriptions in this section, and will therefore be described in Procedure 5.1 for the prediction error method PEM and Procedure 5.2 for the instrumental variables IV method. See also Section 2.3 for more information on the PEM and Section 2.7 for IV. The system identification procedure is also presented in Section 2.8. For a clearer notation, the explicit dependence on time has been omitted in the method descriptions. There is some overlap in the method descriptions regarding the choice of model structure, etc., which makes the algorithms easier to read on their own, but gives redundancy when used together.

Prodedure 5.1 System identification using PEM

```
Require: Input data $x_i$ and output data $x_o$

1: Choose a model structure and model order and form $\hat{y}(x_i, x_o, \theta)$, which is a
   function of input data $x_i$ up to current time $t$, output data $x_o$ up to previous
   time $t-1$ and the unknown parameters $\theta$.
2: Formulate the prediction error (2.4), $\varepsilon = y - \hat{y}(x_i, x_o, \theta)$, using $x_o$ as output
   data $y$.
3: Find the minimizing argument $\hat{\theta}$ in (2.5)
4: if model is a linear regression (2.8) and $l(\varepsilon)$ in (2.5b) is a quadratic function
   (the problem is linear least-squares (LS) and can be solved analytically) then
5:    Use (2.10) to find an LS estimate.
6: else
7:    Solve the minimization problem (2.5) using your favorite numerical solver.
8: end if
9: Repeat with different model structures (linear and nonlinear), model orders,
   etc., until the required model performance is acquired.
```

5.2.1 Method A

Method A is a straightforward application of the standard identification method in the literature. A number of parameter estimation methods are possible. The most common and the one used if nothing else is mentioned, is the PEM. Another parameter estimation method possible is the instrumental variables.

For the analytical inversion, the nonlinearities are assumed to have a well-defined inverse, sometimes with a limitation on the range or domain. It is also interesting to see what can be done numerically with a nonlinearity that is *almost* invertible, for example a function that is monotonous except in a small part of the domain.

Prodedure 5.2 System identification using IV

```
Require: Input data $x_i$ and output data $x_o$

1: Choose a model structure and model order and form $\hat{y}(x_i, x_o, \theta)$, which is a function of input data $x_i$ up to current time $t$, output data $x_o$ up to previous time $t-1$ and the unknown parameters $\theta$.
2: Choose instruments $\zeta(t)$ that are correlated with the system output but uncorrelated with the noise.
3: Formulate the prediction error (2.4), $\varepsilon = y - \hat{y}(x_i, x_o, \theta)$, using $x_o$ as output data $y$.
4: Use (2.13) to find an IV estimate $\hat{\theta}$.
5: Repeat with different model structures (linear and nonlinear), model orders, instruments, etc., until the required model performance is acquired.
```

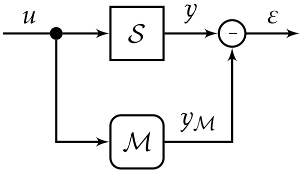


Figure 5.1: *An illustration of the identification part of* METHOD A*, using a prediction error method. The goal is to minimize (a function of) the difference $\varepsilon = y - y_{\mathcal{M}}$ where y is the output from the system $\mathcal{S}$ and $y_{\mathcal{M}}$ is the output from the model $\mathcal{M}$. The box with rounded corners shows the part with unknown parameters to be estimated.*

The dynamics are assumed to be inversely stable, such that both the system and its inverse are stable. LTI systems on the rational form

$$G(q, \theta) = \frac{B(q)}{A(q)}, \tag{5.7}$$

are straightforward to invert, and the inverse is

$$G^{-1}(q, \theta) = \frac{A(q)}{B(q)}. \tag{5.8}$$

The approach is described in Procedure 5.3 and illustrated in Figure 5.1.

5.2.2 METHOD B1

METHOD B1 refers to the standard use of DLA, where two models are estimated. The first estimation is a model of the system itself. This is a standard identification, which of course entails a number of choices like the model structure, the model order, etc., but the identification itself is straightforward. After a model

Prodedure 5.3 METHOD A

```
Require: Input data $u$ and output data $y$

   {Estimation of forward model}
1: Estimate a model $\mathcal{M}$ of the system $\mathcal{S}$ using Procedure 5.1 or Procedure 5.2,
   with $x_i = u$ and $x_o = y$.
   {Construction of inverse model}
2: Invert the forward model analytically or numerically to obtain an exact or
   approximate inverse.

Output: Inverse $\hat{S}^{-1}$
```

validation, where the model is judged to be good enough according to some measure of performance, this model will replace the system in the identification of the preinverse. The output from the inverse model is sent to the simulation model of the system, and the cascaded output should be equal (or similar) to the original input.

The identification in the second step of METHOD B1 is less straightforward. Even though the first step could be a convex optimization problem (depending on for example the model structure chosen), the second one will probably not be, with two cascaded systems. Since the goal is to find a preinverse, the parameters of the preinverse will affect the input to the system (model), so even if both models by themselves are linear in the parameters, the nested problem will not necessarily be.

There are a number of nonconvex optimization solvers in different softwares. In this thesis, the Matlab routine `fminsearch`, based on the Nelder-Mead simplex method, has been used (in Chapters 9 and 10). There are of course other routines and algorithms that can be used. One thing to remember when using a nonconvex solver is that there are generally no guarantees of global optimality, see also Section 2.6. To avoid the nonconvexity of the optimization, the PA problem in this thesis has been reformulated into a (separable) least squares problem, see Chapters 9 and 10.

The approach is described in Procedure 5.4, where each part (forward and inverse modeling) are done until the performance meets the specified criteria, and illustrated in Figure 5.2.

5.2.3 METHOD B2

The iterative METHOD B, called METHOD B2, is based on the idea that the system itself should be used in the estimation. The benefit with this method is that systematic noise contributions can and should be taken into account.

The difference compared to METHOD B1 is that in METHOD B2, the first part, the identification of the forward model, is removed and the system itself is kept in the loop. This means it is necessary to have access to the system that should be inverted, and the possibility to control and change the input. There are multi-

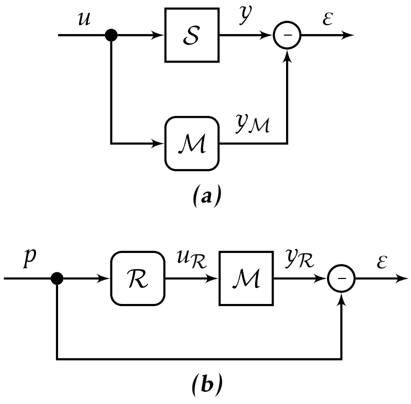


Figure 5.2: *An illustration of the identification part of* Method *B1, using a prediction error method. The modeling of a forward model is shown in (a), and is the same as in* Method *A. The inverse model estimation is shown in (b), where the model from the forward modeling is used in series with the preinverse. The goal in each step is to minimize (a function of) the difference ε, in (a), $\varepsilon = y - y_{\mathcal{M}}$ and in (b), $\varepsilon = y_{\mathcal{R}} - p$. The box with rounded corners shows the part with unknown parameters to be estimated.*

Prodedure 5.4 Method B1

Require: Input data u and output data y

{**Estimation of forward model**}

1: Estimate a model of the system using Procedure 5.1 or Procedure 5.2, with $x_i = u$ and $x_o = y$. This model will be denoted $\mathcal{M}$.

{**Estimation of preinverse model**}

2: Use the forward model $\mathcal{M}$ from Step 1 as a simulation model.

3: Choose a model structure and model order for the preinverse R. This preinverse has input p and output $u_{\mathcal{R}}(\theta, u)$, which is a function of the reference p and the unknown parameters θ.

4: Use the signal $u_{\mathcal{R}}$ as the input signal to the simulation model $\mathcal{M}$. The output from the model $\mathcal{M}$ is denoted $y_{\mathcal{R}}$.

5: Find the parameter values using Procedure 5.1 or Procedure 5.2, with $x_i = p$ and $x_o = y_{\mathcal{M}}$. The model structure in the identification is $\mathcal{M}(\mathcal{R}(\theta, u))$.

6: Repeat Steps 3-5 with different model structures (linear and nonlinear), model orders, etc., until the required model performance is acquired. If needed, repeat also Step 1.

Output: Preinverse $\mathcal{R}$

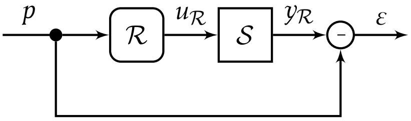


Figure 5.3: *An illustration of the minimization of* M*ETHOD B2, using a prediction error method. The goal is to minimize (a function of) the difference* $\varepsilon = y_{\mathcal{R}} - p$*. An iterative solution can be used to obtain new parameter values in the predistorter block* $\mathcal{R}$*. The difference between this setup and the one in Figure 5.2 (b) is that in this method, the system is still in the loop. The box with rounded corners shows the part with unknown parameters to be estimated.*

ple options to find the preinverse, different stochastic optimization methods are available, see for example Murphy [2012].

There are also multiple possibilities regarding the setup of the optimization. A brute-force method is presented in Section 5.2.5 and relies on a local numerical differentiation, where the parameter values are varied and a numerical differentiation is done. One possibility is to combine METHOD B2 with, for example, METHOD B1 or METHOD C (see below) to try out different model structures, etc, and also to obtain new parameter values between the experiments. This model (or these models) could then be refined to also account for the noise contribution using experiments with the real system in the loop. The disadvantage is that two methods are needed (B1/C and B2), but the advantage is that the number of experiments needed can be significantly reduced by evaluating different model structures, obtaining good initial values, etc., in an offline setting.

The outline of the approach is described in Procedure 5.5, where the inverse modeling is done until the performance meets the specified criteria, and illustrated in Figure 5.3.

5.2.4 METHOD C

METHOD C produces a postinverse, and when used for predistortion it is based on the assumption that the system and the inverse commute, and that a postinverse will work as a preinverse. In this method, the identification is straightforward, but the roles of the input and the output are reversed, compared to the standard identification procedure. Just like in any model estimation, the identification method is up to the user to choose, and the PEM and the IV methods are possible choices.

The approach is presented in Procedure 5.6, where the modeling is done until the performance meets the specified criteria, and illustrated in Figure 5.4.

5.2.5 An iterative solution for METHOD B2

The optimization problem in METHOD B2 is often nonlinear, leading to a nonconvex optimization which requires numerical techniques to solve. Many numerical

Prodedure 5.5 METHOD B2

Require: The system $\mathcal{S}$. The possibility to control the input u. Initial parameter values (could be obtained using for example METHOD B1 or METHOD C).

{**Estimation of preinverse model**}

1: Choose a model structure, model order and initial values for the preinverse R. This preinverse has input x_i and output $u_{\mathcal{R}}(\theta, u)$, which is a function of input data p and the unknown parameters θ.

2: Send $u_{\mathcal{R}}$ as the input signal to the system $\mathcal{S}$. The preinverted output from the system is denoted $y_{\mathcal{R}}$.

3: Use an online optimization that minimizes (a function of) the difference between the reference p and the precompensated output $y_{\mathcal{R}}$. One example of such a method is described in Section 5.2.5, which minimizes the MSE using repeated experiments with the system in the loop.

4: Repeat with different model structures (linear and nonlinear), model orders, etc., until the required performance is acquired.

Output: Preinverse $\mathcal{R}$

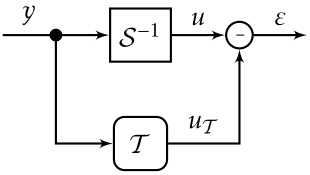


Figure 5.4: *An illustration of the identification part of METHOD C, using a prediction error method. The goal is to minimize (a function of) the difference $\varepsilon = u - u_{\mathcal{T}}$, where it is assumed that the system inverse $\mathcal{S}^{-1}$ exists. This is the inverse estimation problem compared to METHOD A, Figure 5.1, where the box with rounded corners shows the part with unknown parameters to be estimated.*

Prodedure 5.6 METHOD C

Require: Input data u and output data y

{**Estimation of postinverse**}

1: Choose a model structure and model order for the postinverse $\mathcal{T}$. This postinverse has input y and output $y_{\mathcal{T}}(\theta, u)$, which is a function of the input y and the unknown parameters θ.

2: Estimate a model of the system using Procedure 5.1 or Procedure 5.2, with $x_i = y$ and $x_o = u$.

{**Construction of a preinverse, when desired**}

3: Use estimated postinverse as a preinverse.

Output: Postinverse $\mathcal{T}$

optimization solvers use local methods to find an optimum of the cost function. A common choice is Newton's method, where information from the gradient and the Hessian of the cost function are used. The method will converge to a stationary point of the gradient. Depending on the starting point (initial value) of the optimization method, the solver will converge to different local optima, which can be close to or far away from the global optimum. If an analytical description of the gradient and Hessian are not available, numerical approximations can be used.

This can of course be achieved in many different ways, here one iterative solution will be shown, assuming that measurements are not too costly or time-consuming. This can be assumed to be the case in power amplifier predistortion applications, but might not be true in, for example, a robotic application. The method presented here is rather naïve and is just to show that a model can be achieved this way. It is based on local numerical differentiation and is very simple in terms of implementation. The method is non-convex which means it risks only reaching a local minimum. Evaluating different initial values of the method or using one of the other methods here (for example, METHOD B1 or METHOD C) to obtain the initial values can be useful. Other modeling methods can also be explored, depending on the complexity of the model, and the predistorter setup at hand.

The goal is to estimate a preinverse by minimizing (5.1) where

$$V(\theta) = \frac{1}{N}\sum_{t=1}^{N}(u_t - y_{\mathcal{R}t}(u,\theta,w,v))^2 \tag{5.9}$$

and θ comes from the parametrization of $\mathcal{R}(u,\theta)$, determined beforehand. Since the input signal to the system depends on the parameter value of θ, a new experiment is needed for each iteration of θ.

The basic idea is to use a local optimization by adding a small Δ_θ to θ, evaluate the new value of $V(\theta + \Delta_\theta)$ and follow the negative gradient. In higher dimensions, this method can be extended by taking a step in each dimension and evaluating V (that is, assuming that the two parameters are independent, which is of course not the case). First, $V\left(\theta + \begin{bmatrix}\Delta_\theta & 0\end{bmatrix}^T\right)$ is evaluated and then $V\left(\theta + \begin{bmatrix}0 & \Delta_\theta\end{bmatrix}^T\right)$ for a 2D case. This gives the direction of the optimization solver. The approach is described in Procedure 5.7.

5.3 Analysis

METHODS A-C can be applied to any kind of nonlinear dynamic system and to evaluate and analyze the different methods for a general nonlinear dynamic system is difficult. To be able to draw some more conclusions about the properties of the estimated models, three special cases will be looked at. Block-oriented systems are one special case with linear dynamics and static nonlinearities. Purely linear systems will also be analyzed, and IV.

Prodedure 5.7 Iterative preinverse

Require: The system $\mathcal{S}$. Possibility to control the input to the system. Initial values θ^0 (could be done using METHOD B1 or METHOD C for example)

```
{Iterative construction of preinverse}
Choose a model structure, model order for the inverse, and $\Delta_\theta$ and $N_{\mathrm{it}}$ for the
  approach. The inverse model has the reference $r$ as input and output $u_{\mathcal{R}}(\theta, u)$,
  which is a function of the input $u$ and the unknown parameters $\theta$.
Calculate $V^0$ using (5.9) with $\theta = \theta^0$.
for $i = 1 : N_{\mathrm{it}}$ do
  $\theta^i = \theta^{i-1}$
  for all $k = 1 : n$, where $n$ is the dimension of the parameter vector $\theta$ do
    $\bar{\theta} = \theta^{i-1} + \Delta_\theta e_k$, where $e_k$ is the $k$:th unit vector.
    Construct predistorted input signal $u_{\mathcal{R}}$ using parameter vector $\bar{\theta}$.
    Run experiment on system using $u_{\mathcal{R}}$ as input signal.
    Evaluate cost function $\bar{V}$ using (5.9).
    {Update parameter vector $\theta^i$}
    if $\bar{V} < V^i$ then
      $\theta^i = \theta^i + \Delta_\theta e_k$
    else
      $\theta^i = \theta^i - \Delta_\theta e_k$
    end if
  end for
  Construct predistorted input signal $u_{\mathcal{R}}$ using parameter vector $\theta^i$.
  Run experiment on system using $u_{\mathcal{R}}$ as input signal.
  Evaluate cost function $V^i$ using (5.9).
end for
```

In this section, the goal is to show that the results of inverse system identification using the methods introduced in this chapter will be different, depending on the choice of method. In some special cases, the forward and inverse models are the same, but in general this cannot be assumed.

5.3.1 Inverse PEM identification of LTI systems

In this section, the estimation of LTI systems will be investigated. Since linear systems commute, there is no principal difference between METHOD C and METHOD B, except numerical issues in the optimization. Therefore, only METHODS A and C will be discussed here. An example using the theory is presented in Section 6.1.

An LTI dynamical system will be looked at. The model estimation is done in open loop and assuming the output was created according to

$$y(t) = G_0(q)u(t) + H_0(q)e_0(t) \tag{5.10}$$

where G_0 is the true system, H_0 is the true noise dynamics and e_0 is a white noise sequence. Here, the causal case is considered. A kernel-based identification approach for non-causal inverse systems is covered in Blanken et al. [2018] for FIR models.

As shown before, in system identification, the goal is often to find the minimizing argument of a function of the prediction error $\varepsilon(t,\theta)$

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t,\theta)^2 = \arg\min_{\theta} \frac{1}{N}\sum_{t=1}^{N} [y(t) - \hat{y}(t|\theta)]^2, \tag{5.11}$$

where $y(t)$ is the measured output and $\hat{y}(t|\theta)$ is the predicted output given the model parameters θ. Here, a fixed noise model $H_* \equiv 1$ is used such that the prediction is described by $\hat{y}(t|\theta) = G(q,\theta)u(t)$. Looking at the identification from a frequency domain point of view, the minimization criterion in (5.11) can asymptotically be written as [Ljung, 1999, (8.71) p. 266]

$$\hat{\theta} = \arg\min_{\theta} \int_{-\pi}^{\pi} \left|G_0(e^{i\omega}) - G(e^{i\omega},\theta)\right|^2 \Phi_u(\omega)\mathrm{d}\omega \tag{5.12}$$

where $G(e^{i\omega},\theta)$ is the model and $\Phi_u(\omega)$ is the spectrum of the input signal. The estimation will thus be done in a way to emphasize the model fit in frequency bands where the transfer function and the input spectrum are large enough to have a significant impact on the total criterion. The minimization is done with respect to the product of model fit ($|G_0 - G|^2$) and input spectrum. If the input is white noise (flat spectrum), it is thus more important to obtain a good model fit at frequencies with a large transfer function magnitude.

If instead the goal is to estimate the inverse model to be used as described in Section 5.1, the minimization criterion in the time domain can be written

$$\hat{\vartheta} = \arg\min_{\vartheta} \frac{1}{N}\sum_{t=1}^{N} \left[u(t) - \frac{1}{G(q,\vartheta)}y(t)\right]^2 \tag{5.13}$$

and the frequency domain equivalent to (5.13), when y is noise-free, is

$$\hat{\vartheta} = \arg\min_{\vartheta} V_{\text{inv}}(\vartheta). \tag{5.14}$$

The loss function is

$$\begin{aligned}
V_{\text{inv}}(\vartheta) &= \int_{-\pi}^{\pi} \left| \frac{1}{G_0(e^{i\omega})} - \frac{1}{G(e^{i\omega}, \vartheta)} \right|^2 \Phi_y(\omega) \mathrm{d}\omega \\
&= \int_{-\pi}^{\pi} \left| \frac{1}{G_0(e^{i\omega})} - \frac{1}{G(e^{i\omega}, \vartheta)} \right|^2 |G_0(e^{i\omega})|^2 \Phi_u(\omega) \mathrm{d}\omega \\
&= \int_{-\pi}^{\pi} \left| 1 - \frac{G_0(e^{i\omega})}{G(e^{i\omega}, \vartheta)} \right|^2 \Phi_u(\omega) \mathrm{d}\omega \qquad (5.15) \\
&= \int_{-\pi}^{\pi} \left| G(e^{i\omega}, \vartheta) - G_0(e^{i\omega}) \right|^2 \frac{\Phi_u(\omega)}{|G(e^{i\omega}, \vartheta)|^2} \mathrm{d}\omega \qquad (5.16)
\end{aligned}$$

using $\Phi_y = |G_0(e^{i\omega})|^2 \Phi_u$ if no noise is present. The loss function in (5.16) is similar to the weighting for the input error case where $H = G$ so that $y(t) = Gu + Ge = G(u + e)$, that is, the error enters the system at the same place as the input [Åström and Eykhoff, 1971].

Comparing the minimization criterion for the forward estimation in (5.12) to the one for the inverse estimation in (5.15), the weighting is clearly different. In the forward case, a relative model error at a frequency where the system amplification is small, will affect the criterion much less than a model error at a frequency where the system amplification is large. In the inverse estimation case, a relative model error will have the same effect on the criterion for two frequencies with the same input spectral density, and does not depend on the system amplification at that frequency. The weighting, and thus the model fit, between the different frequencies will be shifted to better reflect the importance of a good fit also at frequencies with a small transfer function magnification.

The time domain criterion (5.13) thus leads to the frequency domain description (5.15), and the weighting is automatically done to match the use of the inverse model estimate. Here, only the case when the system and its inverse are both stable and causal will be investigated. See Section 3.2.1 for a brief discussion on the problems involved in system inversion. There are also methods where a forward and an inverse identification leads to the same results, see Ho and Enqvist [2018] for the IV case.

5.3.2 Linear approximations of block-oriented systems

Another common special case of general nonlinear dynamic systems, are block-oriented systems. In this section, the theory presented in Section 4.4 will be applied to Hammerstein structured systems. Examples using the theory in these sections will be presented in Section 6.2 and Section 6.3.

Before the new results, some background to linear and nonlinear filtering of stochastic processes is presented. Let $y(t)$ be a stationary stochastic process obtained by passing a zero-mean stationary process $u(t)$ through a stable LTI system with transfer function $H(z)$ such that $y(t) = H(q)u(t)$. Then

$$\Phi_y(z) = H(z)\Phi_u(z)H(z^{-1}) \tag{5.17}$$

and

$$\Phi_{yu}(z) = H(z)\Phi_u(z). \tag{5.18}$$

Furthermore, if $x(t)$ is jointly stationary with $u(t)$ and $y(t)$, it is also known that

$$\Phi_{xy}(z) = \Phi_{xu}(z)H(z^{-1}). \tag{5.19}$$

See for example Kailath [2000].

These results hold for linear systems only, but some properties can be shown for nonlinear systems. Here, Bussgang's theorem will be used, which is valid for static nonlinearities [Bussgang, 1952]. Assume a differentiable static nonlinearity f with a Gaussian input $u(t)$ and output $x(t)$, such that $x(t) = f(u(t))$. Assume also that both signals have zero-mean, that is $E[x(t)] = E[u(t)] = 0$, and that the covariance functions $R_{xu}(\tau)$ and $R_u(\tau)$ are well-defined for all τ and that $E[f'(u(t))]$ exist. Then

$$R_{xu}(\tau) = c\,R_u(\tau) \tag{5.20}$$

where $c = E[f'(u(t))]$. In the z-domain, Bussgang's theorem is thus equivalent to

$$\Phi_{xu}(z) = c\,\Phi_u(z), \tag{5.21}$$

and $\Phi_{xu}(z)$ and $\Phi_u(z)$ differ only by a scaling factor.

Hammerstein systems

Assume a system with a Hammerstein structure, with a nonlinearity followed by a stable linear system $H(z)$, and no measurement noise on the output is present, see Figure 4.8. The goal is to fit a linear model to the block-oriented system.

Assumption A1. Signals will here be considered with well-defined first and second order moments, z-spectra and canonical spectral factorization according to Assumptions A1 and A2 in Enqvist and Ljung [2005]. When white noise is used, this means *independent, identically distributed* (i.i.d.) variables.

The optimal forward linear approximation of the system with a Gaussian input is

$$\begin{aligned} G_{0,f}(z) &= \frac{\Phi_{yu}(z)}{\Phi_u(z)} = \frac{H(z)\Phi_{xu}(z)}{\Phi_u(z)} = \frac{H(z)\;c\,\Phi_u(z)}{\Phi_u(z)} \\ &= c\,H(z), \end{aligned} \tag{5.22}$$

that is, the noncausal LTI-SOE is equal to the linear subsystem times a constant.

For the inverse estimated directly, the following result is valid.

Lemma 5.1 (Inverse estimation). *Consider a Hammerstein system with a Gaussian input $u(t)$ and output $y(t)$ and intermediate signal (after the nonlinearity) $x(t)$ such that Assumption A1 is fulfilled. Then the noncausal* I-LTI-SOE *for a Hammerstein structure is*

$$G_{0,i}(z) = \frac{c}{H(z)} \frac{\Phi_u(z)}{\Phi_x(z)} \tag{5.23}$$

where $H(z)$ is the linear part of the Hammerstein model.

Proof:

$$\begin{aligned} G_{0,i}(z) &= \frac{\Phi_{uy}(z)}{\Phi_y(z)} = \frac{\Phi_{ux}(z)H(z^{-1})}{\Phi_y(z)} \\ &= \frac{\Phi_{ux}(z)H(z^{-1})}{H(z)\Phi_x(z)H(z^{-1})} = \frac{\Phi_{ux}(z)}{H(z)\Phi_x(z)} \\ &= \frac{\Phi_{xu}(z^{-1})}{H(z)\Phi_x(z)} = \frac{c\Phi_u(z^{-1})}{H(z)\Phi_x(z)} = \frac{c}{H(z)}\frac{\Phi_u(z)}{\Phi_x(z)}. \end{aligned}$$

□

That is, there is an extra dynamic factor $\Gamma(z) = \Phi_u(z)/\Phi_x(z)$. For an i.i.d. white noise input, it follows that Φ_x is constant if Φ_u is, so that only a scale factor differs and the result is

$$G_{0,i}(z) = \frac{\Phi_{uy}(z)}{\Phi_y(z)} = \frac{\tilde{c}}{H(z)}, \tag{5.24}$$

thus, the noncausal I-LTI-SOE is equal to the inverse linear subsystem times a constant.

It can also be seen that the inverse model is proportional to $1/H(z)$ in frequency regions where the spectral densities are flat. Typically, $\Gamma(z)$ has a low-pass characteristic, which means that the linear model will be a scaled version of the true linear system in these frequencies.

For a white noise input, the result from estimating a forward model and inverting it according to METHOD A is the same, up to a constant, as when estimating an inverse model directly as in METHOD C. However, if the input $u(t)$ is not white, (5.24) is not valid, and the weight factor $\Gamma(z) = \Phi_u(z)/\Phi_x(z)$ will affect the optimal linear inverse model.

Should a forward or inverse model of a Hammerstein system be estimated?

Whether or not it is better to estimate a forward model or an inverse model must depend on the application. Two viewpoints can be taken.

Forward: If the linear model is to be used as a part of a nonlinear model (for example as the linear block in a Wiener or Hammerstein model), it is better to estimate the forward model directly, which can then be inverted. For a Gaussian input, this estimation will lead to a model that is a scaled version of the linear system, and as a second step the nonlinearity can be estimated.

Inverse: If the goal is to find a linear approximation of the inverse, and the approximate inverse is to be used, it can be better to estimate the inverse directly. This inverse model will then take into account the use of the estimated model. This can for example be useful in power amplifier predistortion, where an inverse of low complexity is preferred, since it reduces the power consumption and chip area.

5.3.3 Inverse IV identification

As outlined in Section 2.7, the instrumental variable (IV) method is a way to use correlations to estimate a model. For a forward model, it is known that

$$\hat{\theta}^{IV} = \operatorname{sol}\left\{ \frac{1}{N} \sum_{t=1}^{N} \zeta(t) \left[y(t) - \phi^T(t)\theta \right] = 0 \right\}.$$

It has been shown in Ho and Enqvist [2018] for the LTI case that a for a basic IV method with process noise, it does not matter whether a model is estimated in a forward or inverse setting, and that the estimate and the variance are the same for the two models. This is valid even for finite data lengths.

Here, a cubic (nonlinear) example is examined (which was used in Example 4.2) and will be further analyzed in Chapter 7.

Example 5.1: Inverse IV estimation

The system output is described by

$$y = b(u + w)^3.$$

The input signal u and the noise signal w are independent white Gaussian with zero mean and variances σ_u^2 and σ_w^2, respectively. The dependence on time has been omitted for ease of notation.

The instruments of the inverse model are chosen as

$$Z = \begin{bmatrix} u & u^2 & \cdots & u^n & \cdots \end{bmatrix}^T,$$

and the regression vector ϕ

$$\phi = \begin{bmatrix} y & y^{1/3} \end{bmatrix}^T.$$

The inverse model with has a cubic and a linear term, as in

$$\hat{u} = \tilde{\vartheta}_1 y + \tilde{\vartheta}_3 y^{1/3} = \phi^T \vartheta$$

where ϑ is the parameter vector. This means the inverse problem $E\left[Z\left(u - \hat{u}\right)\right] = 0$ is

$$E\left[Z\left(u - \hat{u}\right)\right] = E\left[\begin{pmatrix} u \\ u^2 \\ \vdots \end{pmatrix}\left(u - \hat{u}\right)\right] = E\left[\begin{pmatrix} u \\ u^2 \\ \vdots \end{pmatrix}\left(u - \tilde{\vartheta}_1 y - \tilde{\vartheta}_3 y^{1/3}\right)\right]. \tag{5.25}$$

Since the elements in the vector are linearly dependent, this is equivalent to looking at the first element only. Using $y = b(u+w)^3$ and $y^{1/3} = b^{1/3}(u+w)$ gives

$$E\left[Z\left(u-\hat{u}\right)\right] = E\left[u^2 - \tilde{\vartheta}_1 bu(u^3 + 3u^2w + 3uw^2 + w^3) - \tilde{\vartheta}_3 b^{1/3}(u^2 + wu)\right].$$

Taking the expected value with respect to the noise w, and knowing that $E_w\left[3u^2w\right] = E_w\left[w^3\right] = E_w\left[wu\right] = 0$ and $E_w\left[w^2\right] = \sigma_w^2$ we get

$$E\left[Z\left(u-\hat{u}\right)\right] = E\left[u^2 - \tilde{\vartheta}_1 bu(u^3 + 3u\sigma_w^2) - \tilde{\vartheta}_3 b^{1/3}u^2\right] = 0.$$

The only way to assure that $\tilde{\vartheta}_1 b(u^4 + 3u^2\sigma_w^2)$ for an arbitrary $\sigma_w^2 \neq 0$ and $b \neq 0$ is that $\tilde{\vartheta}_1 b = 0 \quad \Rightarrow \quad \tilde{\vartheta}_1 = 0$. This leaves

$$E\left[u^2 - \tilde{\vartheta}_3 b^{1/3}u^2\right] = E\left[\left(1 - \tilde{\vartheta}_3 b^{1/3}\right)u^2\right] = 0$$

or

$$\tilde{\vartheta}_3 = \frac{1}{b^{1/3}}.$$

The result of the inverse estimation is thus that

$$\tilde{\vartheta}_{\text{IV}} = \begin{bmatrix} 0 & \frac{1}{b^{1/3}} \end{bmatrix}^T = \tilde{\vartheta}_0.$$

That is, the inverse IV method results in an unbiased estimate of the parameters.

For a forward model, the problem coincides with the LS problem, and there will be a bias in the linear term.

6

Examples of approximations in noise-free measurements

In an ideal world, every signal can be measured and all measurements are perfect. Even in this case, it is often the case that it is not possible or desired to find or use the same structure for the model as the system, and model approximations are needed. The system could be very complex making it impossible to find the corresponding model structure, or it could be a choice to use a simplified model structure (for example, a linear model for a nonlinear system) to make it easier to work with the model.

In this chapter, some of the theory presented in earlier chapters will be illustrated by examples. The data here is all noise-free, and so the examples present what happens in a best-case-scenario. In Chapter 7, examples with noisy data are presented.

The inverse can either be used as a preinverter or a postinverter, see Figure 6.1 (the same as Figure 4.1, duplicated here for easy access). Both preinversion and postinversion will be evaluated when possible (in the linear case, the system and its inverse commute and there is no difference between a preinverse and a postinverse). The goal here is then to regain the original input signal or reference signal, using the preinverse or postinverse, such that $y_{\mathcal{R}} = p$ or $y_{\mathcal{T}} = u$, respectively.

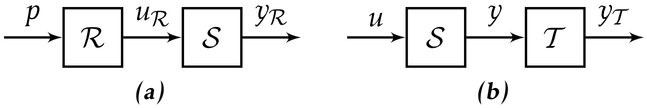


Figure 6.1: *The intended use of the estimated inverses. (a) shows a preinverse $\mathcal{R}$, where the inverse is applied before the system $\mathcal{S}$. (b) shows a postinverse $\mathcal{T}$, where the order of the system and the inverse is reversed.*

6.1 METHOD A and METHOD C for linear systems

In Section 5.3.1, a theoretical analysis of METHOD A and METHOD C was presented for linear systems and here, those results will be illustrated with an example.

Example 6.1: A linear dynamic system with lower order model

In this example, a linear, resonant system is considered. The two methods METHOD A (estimate a forward model and invert it) and METHOD C (estimate an inverse model directly) will be evaluated.

The goal is to obtain a system inverse to be used in series with the original system in order to retrieve the input, see Figure 6.1a. The input u and the noise-free output y are measured with no predistorter present. The system has two resonance frequencies, at $\omega = 1$ rad/s and $\omega = 10$ rad/s. The magnitudes of the two resonance peaks are very different, with the first one a hundred times larger than the second one. The true system, G_0 is described by

$$G_0(s) = \frac{10}{s^4 + 1.1s^3 + 101.1s^2 + 11s + 100} \tag{6.1}$$

and the Bode magnitude diagram is shown in Figure 6.2. The input consists of three sinusoids around each of the two resonance peaks such that the input power is concentrated in two bands, centered around the resonance frequencies, i.e.,

$$u(t) = \sum_{k=1}^{6} a_k \sin(\omega_k t + \phi_k) \tag{6.2}$$

with $a_k = 1$ for $k = 1, 2, 3$, $a_k = 10$ for $k = 4, 5, 6$, $\omega_k \in \big(0.9,\ 1,\ 1.1,\ 9,\ 10,\ 11\big)$ and $\phi_k \sim U[-\pi\ \ \pi]$. The input amplitude and the frequency points are illustrated by the stars in Figure 6.2. The sampling time is $T_s = 0.02$ s and $N = 10\,000$ simulated measurements have been collected.

With the goal of using an FIR model as a preinverse $\mathcal{R}$ to recover the input, two models have been estimated, using METHOD A and METHOD C. An FIR model depends only on previous input signals, as described in Section 2.4. As the system is linear, the ordering of the two systems does not matter, and the preinverse and postinverse are interchangeable.

First, a forward model has been estimated as an output error (OE) model using System Identification Toolbox in MATLAB [Ljung, 2003], with `[nb nf nk] = [1 3 0]`. This model has then been inverted resulting in an FIR model with 4 terms, according to METHOD A. The approximative inverse using METHOD C is an FIR model with 4 terms, i.e., `[nb nf nk] = [4 0 0]`, and will have a very different weighting. Hence, the two inverses will catch different behaviors of the system. The system G_0 in (6.1) is a fourth order system whereas the model is third order. Thus, the model cannot perfectly model the system but should be able to capture one resonance peak and the overall behavior of the system.

As can be seen in the Bode magnitude plot in Figure 6.3, the METHOD A model has a much better fit around $\omega = 1$ rad/s and almost perfectly models

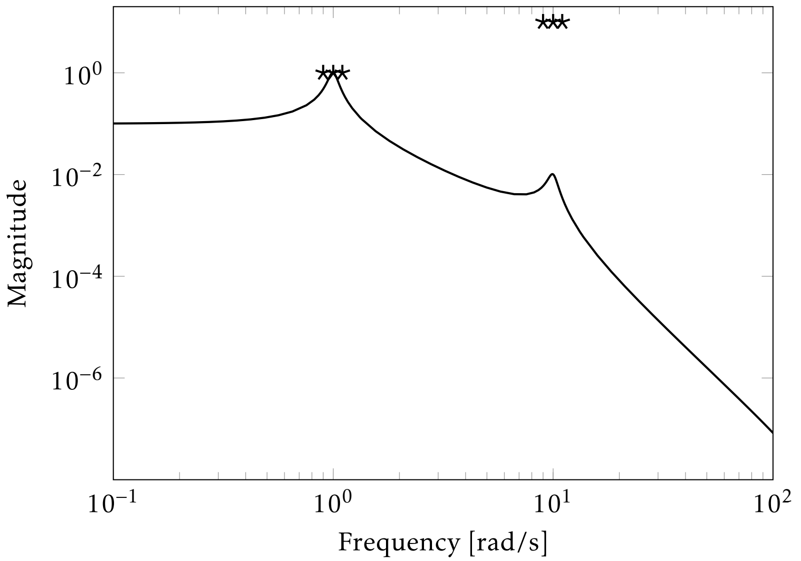


Figure 6.2: *The Bode magnitude plot of G_0 in* (6.1) *is marked by the solid line. The stars mark the amplitude of the multisine input (u in* (6.2)*) components at each frequency.*

the resonance peak, but completely misses the second resonance peak at $\omega = 10$ rad/s. The inverse estimate, the METHOD C model, on the other hand, does not manage to catch either of the resonance peaks in a satisfactory way but catches the amplitudes at both of the resonance frequencies. That is, the amplification at $\omega = 1$ and 10 rad/s is well captured, but not the behaviors around the peaks. Estimating the forward model in the standard way will clearly focus on the frequencies where the product of model error $|G_0 - G|^2$ and input spectrum is large. When this system approximation is inverted, according to METHOD A, the errors around $\omega = 10$ rad/s will become prominent.

The results in the time and frequency domains are presented in Figures 6.4 and 6.5. In the time domain plot in Figure 6.4, it is clear that the METHOD C model better reconstructs the input than the METHOD A model. In Figure 6.5, the periodograms of the reconstructed inputs are shown, zoomed in around the input frequencies. At the lower frequency around $\omega = 1$ rad/s, the METHOD A model captures the input almost perfectly, but around $\omega = 10$ rad/s, the reverse is true and the METHOD C model performs better.

As shown in this small example, there are clearly occasions when it is advantageous to estimate an approximate inverse directly as opposed to estimating the forward model and then inverting it.

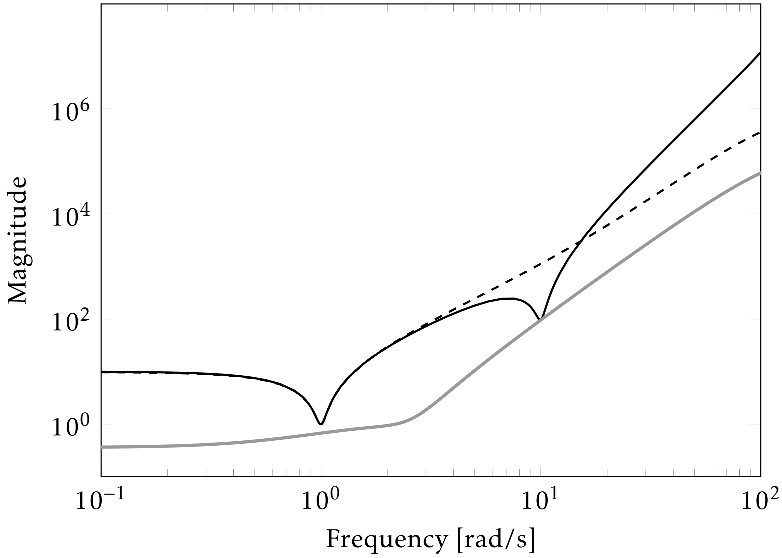


Figure 6.3: *The Bode magnitude response of G_0^{-1} (black solid line), the inverted estimated forward model from* Method *A, (black dashed line) and the inverse model estimate using* Method *C (gray solid line). The* Method *A model perfectly catches the resonance peak at $\omega = 1$ rad/s, whereas the* Method *C inverse does not model either of the resonance peaks in a satisfactory way. The* Method *C model instead has an accurate modeling of both peak frequency values, that is, it manages to accurately model the amplification at $\omega = 1$ and 10 rad/s, but not the resonance peaks.*

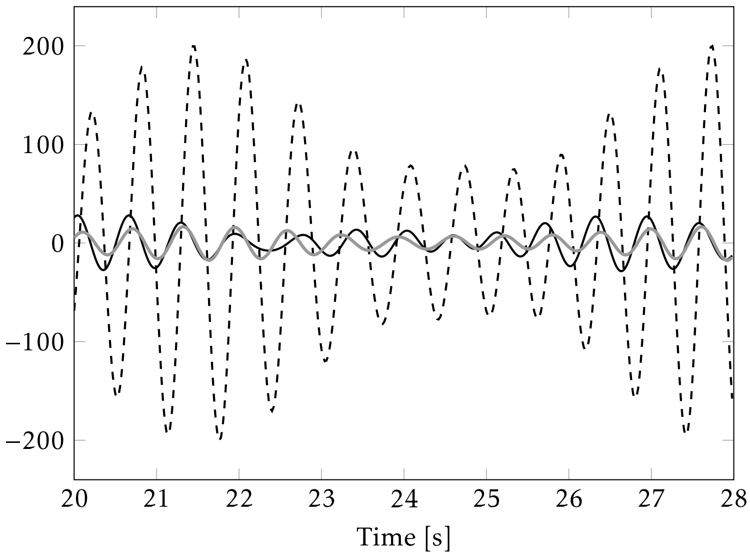


***Figure 6.4:** The input u (black solid line), and the reconstructed input y_u using METHOD A (black dashed line) and the METHOD C model (gray solid line). The estimation of the inverse cannot perfectly reconstruct the input, but is clearly better than the inverted forward model.*

6.2 Linear models of Hammerstein systems

The theoretical aspects of modeling a block-oriented system using a linear model are presented in Section 5.3.2 and here illustrated by two examples.

Example 6.2: Hammerstein system with white and colored inputs

In this example, the goal is to estimate a linear model to a nonlinear system, which is a Hammerstein system.

Two input signals $u(t)$ are used, one white noise sequence and one with colored noise, where a white noise sequence $e(t)$ has been passed through an FIR filter $L(q)$. The input $u(t)$ is passed through a static nonlinearity $x(t) = f(u(t))$ and then through an LTI filter $H(q)$. See also Figure 6.6. Here, the nonlinearity is

$$f(u(t)) = u^3(t) \tag{6.3}$$

and the LTI filter is

$$H(q) = \frac{1}{1 + 0.5q^{-1}}. \tag{6.4}$$

Two models have been estimated with the goal of using an FIR model as pre- or postfilter to recover the input $u(t)$; one forward (which has then been inverted) and one inverse. The first, a forward model, has been estimated as an output error (OE) model using System Identification Toolbox in MATLAB, with `[nb nf nk] = [1 1 0]`, and has then been inverted (resulting in an FIR model

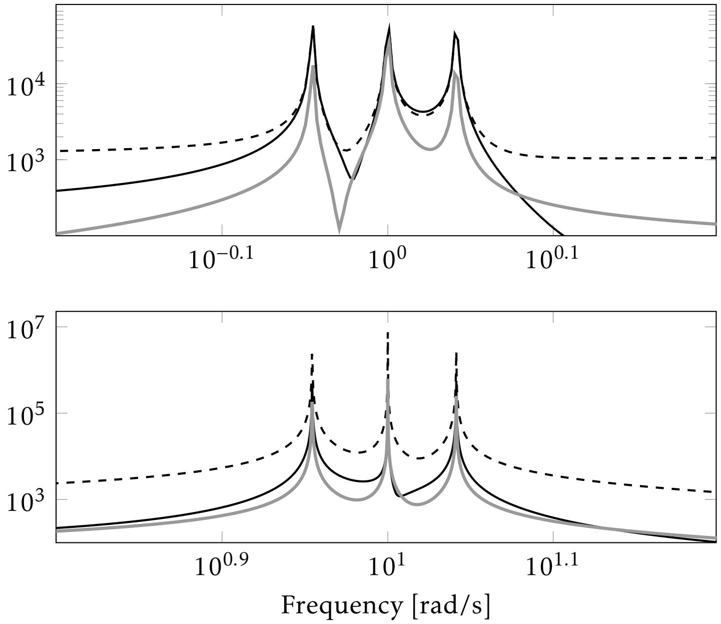


Figure 6.5: *Periodogram of the input u (black solid line), and the reconstructed input $y_{\mathcal{R}}$ using* Method *A (black dashed line) and* Method *C (gray solid line) around $\omega = 1$ rad/s (top) and $\omega = 10$ rad/s (bottom). It is clear also in the frequency domain that the forward model captures the behavior around $\omega = 1$ rad/s better than the inverse estimation, but the reverse is true around $\omega = 10$ rad/s.*

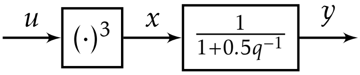


Figure 6.6: *The Hammerstein system used in the example. The input u (white noise or filtered white noise) is passed through a cubic nonlinearity and an* LTI *filter $H(q)$.*

with 2 terms) as in METHOD A. The directly estimated approximate inverse, from METHOD C, is an FIR model with 2 terms (`[nb nf nk] = [2 0 0]`).

The colored input is constructed by FIR-filtering a white noise sequence $e(t)$ with variance σ^2 and $E\left[e(t)\right] = 0$ as

$$u(t) = L(q)e(t) = (1 + l_1 q^{-1})e(t), \tag{6.5}$$

with $l_1 = 0.7$. Then

$$\begin{aligned} R_x(0) &= 15\sigma^6 \left(1 + 3l_1^2 + 3l_1^4 + l_1^6\right) \\ R_x(\pm 1) &= \sigma^6 \left(9l_1 + 24l_1^3 + 9l_1^5\right) \\ R_x(\tau) &= 0 \quad |\tau| > 1 \end{aligned} \tag{6.6}$$

such that

$$\begin{aligned} \Phi_x(e^{i\omega}) &= \sum_{\tau=-\infty}^{\infty} R_w(\tau)e^{-i\tau\omega} \\ &= (e^{i\omega} + e^{-i\omega})\sigma^6 \left(9l_1 + 24l_1^3 + 9l_1^5\right) + 15\sigma^6 \left(1 + 3l_1^2 + 3l_1^4 + l_1^6\right). \end{aligned} \tag{6.7}$$

The spectral density for u is

$$\begin{aligned} \Phi_u(e^{i\omega}) &= L(e^{i\omega})\Phi_e L(e^{-i\omega}) \\ &= \sigma^2 \left((e^{i\omega} + e^{-i\omega})l_1 + (1 + l_1^2)\right). \end{aligned} \tag{6.8}$$

This leads to an additional dynamic factor

$$\begin{aligned} \Gamma(z) &= \frac{\Phi_u(z)}{\Phi_x(z)} \\ &= \frac{\frac{1}{3\sigma^4}\left((z + z^{-1})l_1 + (1 + l_1^2)\right)}{(z + z^{-1})\left(3l_1 + 8l_1^3 + 3l_1^5\right) + 5\left(1 + 3l_1^2 + 3l_1^4 + l_1^6\right)}, \end{aligned} \tag{6.9}$$

in the noncausal I-LTI-SOE. The spectral density $\Gamma(e^{i\omega})$ is shown in Figure 6.7.

It is clear that the weighting function Γ is not a constant, and that an inverted noncausal LTI-SOE, G_{fi}, will be different from a noncausal I-LTI-SOE G_i, estimated directly.

Numerical results for a simulated example (Monte Carlo simulations with 10 runs) with $N = 100\,000$ data points are presented for the Hammerstein example. The estimated models G_{fi} and G_i are shown in Table 6.1, and the resulting fit to data

$$\text{fit} = 100\left(1 - \frac{\sum_{t=1}^{N}(u(t) - \hat{u}(t))^2}{\sum_{t=1}^{N}(u(t) - \bar{u})^2}\right) \tag{6.10}$$

in Table 6.2 with

$$\hat{u}(t) = G_m y(t), \ \ m = i, fi \tag{6.11}$$

and $\bar{u}$ as the mean value of $u(t)$.

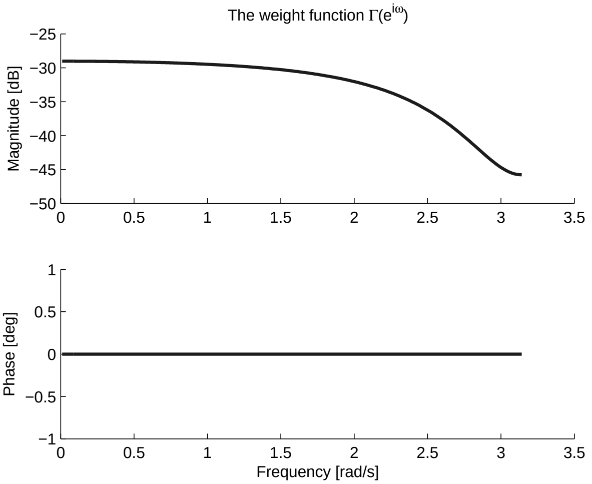


Figure 6.7: *Bode plot of the dynamic factor* $\Gamma(e^{i\omega}) = \Phi_u(e^{i\omega})/\Phi_x(e^{i\omega})$ *in* (6.9) *for a Hammerstein example.*

According to the results in Section 5.3.2, it could be expected that an inverted forward model and an inverse model are the same, up to a constant. As discussed in Section 5.3.2, these models should also be a constant times the true linear system. In this example, $b_0 = 1$ and $b_1 = 0.5$, so any model estimate where $\hat{b}_0 = 2\hat{b}_1$ fulfills these results. This is however not true for a colored input signal, illustrated here by an input filter $L(q)$. In Table 6.1, it can be seen for the white noise input that the models are the same, up to a constant, as expected (in this case, $G_{fi} = 1.66 \cdot G_i$). For the colored noise input where this is not valid for the METHOD C model, the forward model still estimates a constant times $H(q)^{-1}$.

The amplitude of the output of a precompensated versus a postcompensated system can be considerably different. To reduce the effects of the signal amplitudes, normalization can be used. Here, two different scaling approaches will be used, namely to use the signals as they are or to normalize the output such that the input and output have the same variance. The importance of normalization in METHOD C for PA predistortion is discussed in Chani-Cahuana et al. [2015].

The model fit results, which are presented in Table 6.2, are hard to draw conclusions from, but it is clear that the method of model estimation heavily affects the results. The use of normalization seems to lead to a degraded G_i. In general, the postinverse seems to perform better, but the choice of pre- or postinverse is often given by the application.

To illustrate a case where the estimated inverse can be beneficial, a purely linear system can be looked at, as in Example 6.3. That is, a Hammerstein system with $f(u) = u$.

Table 6.1: *Estimated models for a Hammerstein system. The inverse models G_{fi} and G_i have the structure $B(q) = (b_0 + b_1 q^{-1})$. $B_0(q) = (1 + 0.5q^{-1})$.*

Input filtering	Normalization	G_{fi}		G_i	
		b_0	b_1	b_0	b_1
none	none	0.333	0.166	0.200	0.100
none	variance	1.48	0.741	0.894	0.449
$L(q)$	none	0.223	0.112	0.127	0.088
$L(q)$	variance	1.228	0.611	0.691	0.482

Table 6.2: *Model fit,* (6.10)*, of reconstructed input $\hat{u}(t) = B(q)y(t)$ using models from Table 6.1.*

Input filtering	Normalization	G_{fi}		G_i	
		Pre $\mathcal{R}$	Post $\mathcal{T}$	Pre $\mathcal{R}$	Post $\mathcal{T}$
none	none	12.7	19.1	3.0	37.1
none	variance	18.9	37.0	–	–
$L(q)$	none	18.9	38.4	8.1	2.1
$L(q)$	variance	18.2	38.1	–	–

– denotes a negative fit.

Example 6.3

In this example the setup used in Example 6.1 is used (a linear resonant system). We are guessing it is a Hammerstein structured system, with a nonlinearity followed by a linear dynamic system. As a first step we want to estimate the linear part (in two ways). We can then look at the input nonlinearity non-parametrically to see what it could look like. By filtering the output with the inverse of an estimated linear model, an estimate of the intermediate signal is obtained, x, in a Hammerstein system. Plotting this signal, $\hat{x}$, as a function of the input u it should be possible to see how nonlinear the first system block really is.

A METHOD A model and a METHOD C model were identified, resulting in two FIR models with 4 terms. As was shown in Figure 6.2, the two models will catch different behaviors of the system. The METHOD A model models the $\omega = 1$ rads/s resonance peak well but misses the second peak, whereas the METHOD C estimate gives a reasonable fit at both resonance frequencies but does not capture the resonance peak behavior.

A moving average approximation of the nonlinearity has been computed from the estimated signal $\hat{x}$ and is plotted as a function of the input u in Figure 6.8, using METHODS A and C. It can be seen that the METHOD C estimate leads to a function that looks rather like a straight line (the true nonlinearity is a straight line with slope 1) whereas it would be hard to see this in the nonlinearity estimate using METHOD A. Since the METHOD C model better captures the intended use of the model, the signal estimate using the directly estimated inverse is better in this sense.

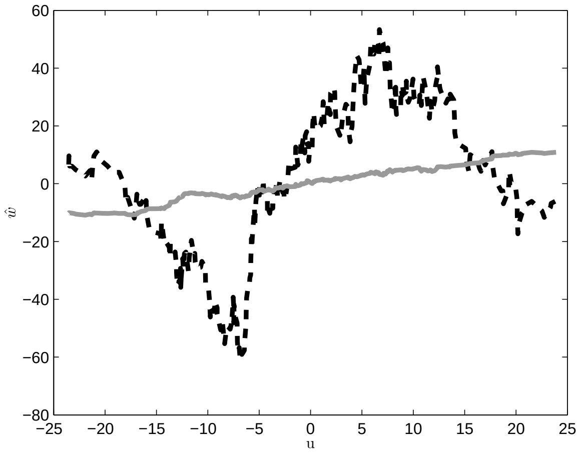


Figure 6.8: *A nonparametric moving average approximation of the nonlinearity in Example 6.3 has been obtained by taking the mean over 41 data points. The estimated signal $\hat{x}$ is presented as a function of the input amplitude, using METHOD A (black dashed line) and METHOD C (gray solid line).*

This example of a linear system interpreted as a Hammerstein system shows that estimating a model in the intended setting really does make a difference. When an inverse was estimated with METHOD C, using the inverse model to approximate the nonlinearity works well.

6.3 Linear models for Wiener systems

Another type of block-oriented system is the Wiener system, where an LTI system is followed by a static nonlinearity. To see that there is a difference in the models estimated using METHODS A and C, let us look at a similar example as in the Hammerstein case.

Example 6.4

Let a white Gaussian input signal pass through an LTI filter $L(q) = (1 + l_1 q^{-1})$ followed by a cubic nonlinearity. Then the forward model, the noncausal LTI-SOE, will be

$$\begin{aligned} G_{0,f}(z) &= \frac{\Phi_{yu}(z)}{\Phi_u(z)} = \frac{3\sigma^4(1+l_1^2)(1+l_1 z^{-1})}{\sigma^2} \\ &= 3\sigma^2(1+l_1^2)(1+l_1 z^{-1}). \end{aligned} \tag{6.12}$$

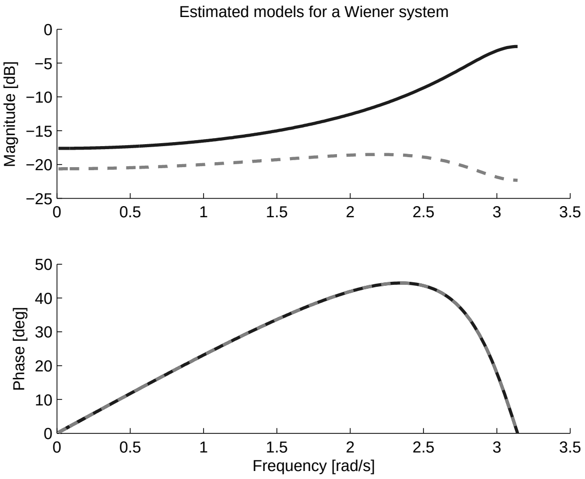


Figure 6.9: *Bode plots of the optimal inverse models for a Wiener system. White Gaussian noise has been used as input to an* LTI *filter* $L(q)$ (6.5) *and then passed through a cubic nonlinearity. The solid line is the inverted noncausal* LTI-SOE $G_{0,fi}$*, and the dashed is noncausal* I-LTI-SOE $G_{0,i}$.

The inverse, noncausal I-LTI-SOE, on the other hand is

$$G_{0,i}(z) = \frac{\Phi_{uy}(z)}{\Phi_y(z)} = \frac{3\sigma^4(1+l_1^2)(1+l_1 z)}{(z+z^{-1})\phi_1 + \phi_2} \tag{6.13}$$

with

$$\phi_1 = \sigma^6\left(9l_1 + 24l_1^3 + 9l_1^5\right)$$

and

$$\phi_2 = 15\sigma^6\left(1 + 3l_1^2 + 3l_1^4 + l_1^6\right).$$

One can see that the inverted optimal forward model (6.12) and the optimal inverse model (6.13) differ by more than a constant, so that the models estimated in the forward and inverse approaches will be different. Bode plots of the two optimal inverse models with $l_1 = 0.7$ are presented in Figure 6.9.

The results for a Hammerstein system that a forward model and its inverse differ by only a constant is not valid for a Wiener system, even when a white input is used.

6.4 Discussion

In the examples discussed in this chapter, the signals have all been noise free. This is an ideal setting used to look at what can be achieved, since all real measurements are corrupted by noise and disturbances. For an inverse model that can capture the true inverse, the METHOD A and METHOD C inverse models are the same in the examples used in this thesis, for noise-free data. However, once there are approximations and simplifications done on the model structures, different results are achieved depending on the choice of method.

In this chapter it has been shown that the method of estimation of the inverse model, even under ideal settings, will have an impact on the model performance. It has been said before for other contexts (feedback for example [Ljung, 1999]) that a model should be estimated in the setting in which it will be used. The same is valid for inverse system identification as was illustrated here. An inverted forward model will not necessarily be the same as an inverse model estimated directly, and the choice of estimation method should depend on the goal.

In this thesis, the goal is to find an inverse model and in the examples in this chapter, we have shown that it can be beneficial to estimate the inverse directly.

7 Inverse systems with noisy data

In Chapter 6, we looked at some examples where the measurements are perfect, and no noise is present. In this chapter, noisy data will be used instead. The noise can enter at the measurement (at the output) and is then called measurement noise, or it can enter the system earlier and is then called process noise.

When there is no noise present, having the true inverse will lead to a perfect inversion when used with the system, as a preinverse or a postinverse. However, as could be seen in the examples in Chapter 4, when noise is present this is no longer the case.

The different identification methods described in Chapter 5 will be illustrated and analyzed in this chapter. These include the least-squares method (LS), the instrumental variables method (IV), and the iterative method METHOD B2 where repeated measurements are used to construct a preinverse. The focus in this chapter will mainly be in preinversion. However, when applicable the inverse models will also be evaluated as a postinverse. The goal is to illustrate some of the theoretical results, and to be able to draw conclusions. The small case study will allow comparisons of the methods presented in earlier chapters. The performance of the methods can easier be seen in a small case study where the noise can be made exactly the same in the evaluation of the different methods. A next step would be to evaluate the different methods in data from, for example, power amplifiers.

We will also conclude this part of the thesis with some thoughts concerning inverse estimation.

7.1 Results using a cubic model structure

To make some comparisons, a case study with a cubic function like the one used in Example 4.2 will be used. The system is illustrated in Figure 7.1.

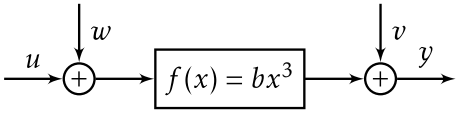


Figure 7.1: The system setup, with a cubic nonlinearity f, input signal u, process noise w, measurement noise v and output signal y.

Assume the true system is a static cubic nonlinearity,

$$\mathcal{S} = f(u) = bu^3. \tag{7.1}$$

There is process noise w affecting the input and measurement noise v at the output, such that the actual system is

$$y = \mathcal{S}(u, w, v) = b(u + w)^3 + v. \tag{7.2}$$

The signals u, w and v are white, zero-mean and mutually independent.

Method C can be used to estimate an inverse model. The model structure of the inverse model is

$$\hat{u} = \vartheta y^{1/3}. \tag{7.3}$$

The true parameters for the inverse model is

$$\vartheta_0 = \left[\frac{1}{b^{1/3}}\right] \tag{7.4}$$

showing that the true inverse is contained in the model set given by this model structure for a specific choice of parameters.

To be able to make comparisons, also forward models will be evaluated and inverted, according to Method A. Here,

$$\hat{y} = \theta u^3, \tag{7.5}$$

and the true parameters for the forward model is

$$\theta_0 = [b]. \tag{7.6}$$

In this chapter, the forward models will use θ as a parameter and the inverse models ϑ.

The true parameter is $b_0 = 4$ and the noise variances are $\sigma_w = \sigma_v = 0.2$ and $\sigma_u = 1$, that is, both process noise and measurement are present and of the same size. To evaluate the estimates, $N_{mc} = 100$ Monte Carlo simulations have been performed with $N = 10\,000$ samples in each run.

Methods A, B2 and C will be evaluated. See Section 5.2.1 and Algorithm 5.3 (page 65) for Method A, Section 5.2.3 and Algorithm 5.5, (page 68) for Method B2 and Section 5.2.4 and Algorithm 5.6 (page 68) for Method C. For Methods A and C, both ls and iv estimators will be evaluated,

The results will mainly be evaluated by Monte Carlo simulations, and illustrated by box plots. These are a way of graphically showing, for example, the

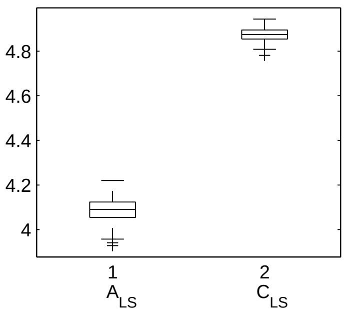


Figure 7.2: Cubic model structure, LS *method. The estimates of b using A_{LS} and C_{LS} methods. The true value is $b_0 = 4$ and it is clear that A_{LS} is closer to the true value than C_{LS}. For C_{LS}, the estimate of b has been produced by taking $\hat{b} = \frac{1}{\vartheta^3}$.*

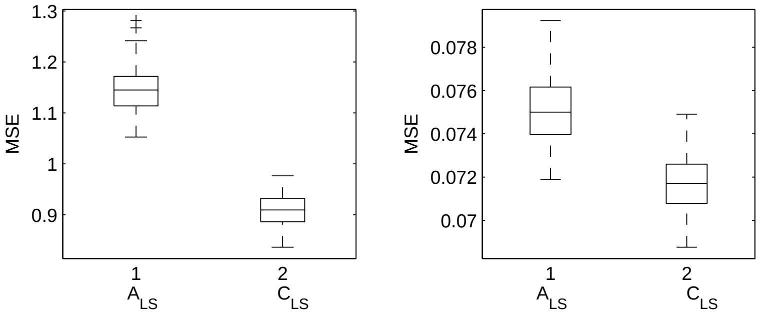
statistics of the parameter value or the MSE value. The box represents the first and third quartiles, and the band inside the box is the median. The whiskers show 1.5 times the interquartile range and outliers are represented by crosses.

In Example 4.2, calculations showed that the true inverse is neither optimal as preinverse nor postinverse, for the case with only process noise. In this section, estimation results will be shown. For fairness in the comparison, the same noise realization has been used in the evaluation.

7.1.1 Least squares method

Since the models (7.3) and (7.5) can be written as linear regressions, the least squares method can be used to estimate the parameters, cf. (2.8) and Sections 2.4 and 2.5. The fact that the structure can be written as a linear regression does not say anything about bias, etc., in the parameter estimates. Both a forward model (METHOD A) and an inverse model (METHOD C) can be estimated using least squares, and can be applied as a preinverse or a postinverse. The methods will be denoted A_{LS} and C_{LS}.

The estimates of the A_{LS} method and the C_{LS} method are presented in Figure 7.2. The C_{LS} estimate $\hat{b}$ has been produced by taking $\hat{b} = \frac{1}{\vartheta^3}$. It is clear that the A_{LS} estimate is closer to the true value of $b_0 = 4$, but this does not mean the performance as a preinverse or a postinverse is better for the inverted forward model estimate A_{LS}. The performance evaluation is done using the MSE (4.11), and the results are presented in Figure 7.3. The results using the inverted forward estimate A_{LS} and the inverse estimate C_{LS} are presented along with the true value, used as both (a) preinverse and (b) postinverse. It can be seen that the C_{LS} estimate performs better than the A_{LS} estimate in both the preinverse and postinverse cases, and that both of them perform better than the true inverse. This is because the noise present in the measurements should be taken into account when the inverse is constructed.



(a) *(b)*

Figure 7.3: *Cubic model structure,* LS *method. The performance of estimates from a least squares estimation, evaluated as (a) a preinverse and (b) a postinverse. The A_{LS} and C_{LS} model estimates are evaluated and compared to the true inverse model, and it is clear that the C_{LS} method performs best both as a preinverse and a postinverse. The A_{LS} method, which has a better estimate of b_0 performs worse than the C_{LS} method but the results are slightly better than for the true value.*

7.1.2 Instrumental variables method

The instrumental variables method described in Section 2.7 has also been evaluated for METHODS A and C. The methods will be denoted A_{IV} and C_{IV}. The instruments are

$$\zeta = \begin{bmatrix} u & u^3 \end{bmatrix}^T. \tag{7.7}$$

These models can be applied as a preinverse or a postinverse.

The parameter estimation results are presented in Figure 7.4 where it can be seen that both the forward model and the inverse model obtain a good estimate $\hat{b}$ of the parameter b. It is easily computed that the A_{IV} estimate has a small bias ($\hat{\theta} = b + 3b\frac{\sigma_w^2}{\sigma_u^2} \neq \theta_0$) but the C_{IV} estimate is unbiased ($\hat{\vartheta} = \frac{1}{b^{1/3}} = \vartheta_0$). The performance evaluation is done using the MSE (4.11), and the results are presented in Figure 7.5. Since the parameter values are close to each other and to the true value, all three sets of parameter values performs similarly, with the A_{IV} estimate slightly better both as a preinverse and as a postinverse.

7.1.3 Iterative method B2

The iterative method used in METHOD B2 is described in Section 5.2.5. There are many optimization algorithms that take the stochasticity into account, and can be used. Here, the goal is not to find the optimal solver, but more to show a proof of concept that it is possible to find a better solution than the true inverse $\mathcal{S}^{-1}$, by

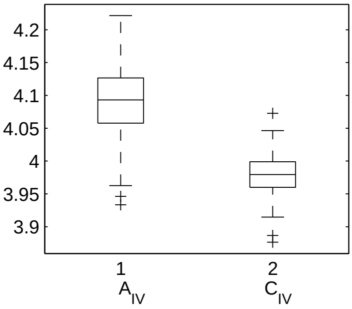
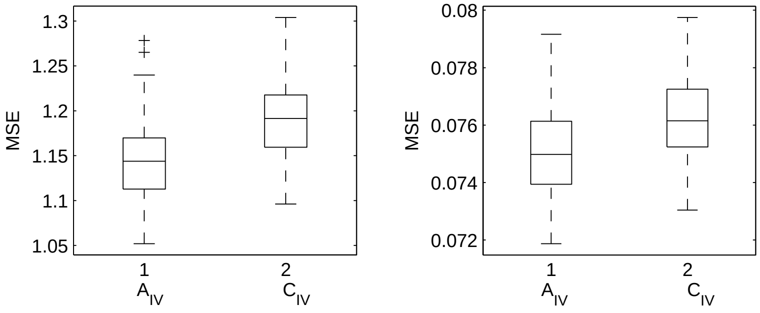
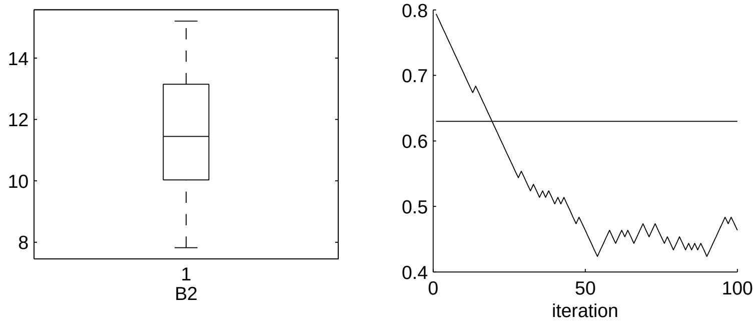


Figure 7.4: *Cubic model structure,* IV *method. The estimates of b using A_{IV} and C_{IV} methods. The true value is $b_0 = 4$ and both* METHODS *A and C estimates using* IV *are close to the true value. For the C_{IV} method, the estimate of b has been produced by taking $\hat{b} = \frac{1}{\vartheta^3}$.*



(a) *(b)*

Figure 7.5: *Cubic model structure,* IV *method. The performance of the parameter estimates evaluated as (a) a preinverse and (b) a postinverse. The model estimates using A_{IV} and C_{IV} methods are evaluated and compared to the true value, and the performance is similar for all three estimates, with A_{IV} slightly better than the others.*



(a) *(b)*

Figure 7.6: *Cubic model structure, iterative* METHOD *B2. (a) Estimates from 100 Monte Carlo simulations and (b) the values of θ during one Monte Carlo run. The initial value $\theta = 0.8$ corresponds to a $\hat{b} = 1.95$.*

using the real system in producing the preinverse. This will ensure that the noise will be accounted for in the preinverse.

The step length Δ_ϑ was fixed. No stopping criterion was used but the method runs $N_{\text{it}} = 100$ times, meaning N_{it} experiments are needed. Here, the solution has converged in around 50 iterations, but without a stopping criterion the algorithm will run through all N_{it} iterations. The number of function evaluations is $(n + 1)N_{\text{it}}$ (one in each step direction (that is, the dimension of the parameter vector n) and one final evaluation with the updated θ, for each iteration). A gridding was also performed for evaluation purposes (leading to d function evaluations, where d is the number of values evaluated for ϑ).

This naïve implementation of a stochastic solver finds a better solution (in LS sense) by minimizing the cost function $V(\theta)$ than when using the true inverse as a preinverse. The estimation results are presented in Figure 7.6. The estimate of the Monte Carlo simulations are shown in (a) and the values of θ during one Monte Carlo run in (b). Figure 7.7 shows the evaluation of METHOD B2 as a preinverse, and it is shown that it performs better than the true value.

7.1.4 Comparisons between the methods

In Figure 7.8, the estimation results of the different methods are presented, where $b_0 = 4$. Figure 7.9 shows the MSE results from the methods, evaluated as a preinverse and a postinverse.

For a preinverse, all methods are evaluated, and METHOD B2 clearly performs better than the others. The only unbiased estimator is the C_{IV}, the other methods are all biased. The estimate from METHOD B2 is much larger than the other estimates, and has a larger variance. However, this bias in the estimate takes the noise contribution into account, and in the evaluation (both as a preinverse and a postinverse) the C_{LS} estimate leads to a better MSE than the C_{IV}. The comparison

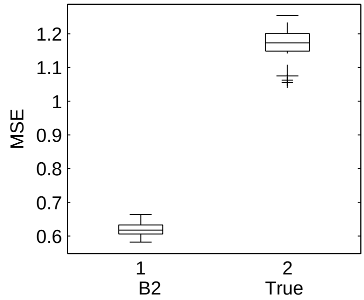


Figure 7.7: *Cubic model structure. The performance of the iterative method B2, evaluated as a preinverse and compared to the true value.*

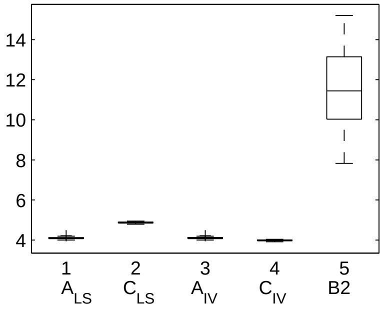


Figure 7.8: *Cubic model structure. The estimates of b using METHODS A, B2 and C. The true value is $b_0 = 4$. For B2, C_{LS} and C_{IV}, the estimate $\hat{b}$ has been produced by taking $\hat{b} = \frac{1}{\vartheta^3}$.*

is slightly unfair since we are evaluating with respect to the MSE, so that the LS method performs best is not a surprise. However, it is still an interesting comparison since many measures of goodness are based on the MSE (or the RMSE).

7.2 Results using a cubic and linear model structure

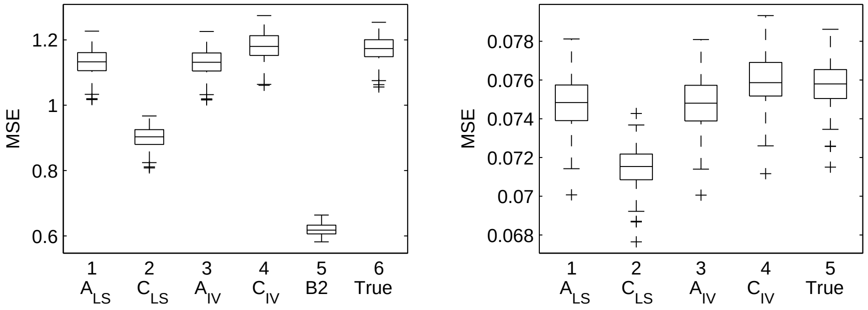
It can be beneficial to try different model structures. In this section, an expanded model structure is evaluated.

A forward model containing a linear term will be evaluated with the model structure

$$\hat{y} = \tilde{\theta}_1 u + \tilde{\theta}_3 u^3 = \begin{bmatrix} u & u^3 \end{bmatrix} \begin{bmatrix} \tilde{\theta}_1 \\ \tilde{\theta}_3 \end{bmatrix} \triangleq \phi^T \tilde{\theta}. \tag{7.8}$$

The true parameters for the forward model is

$$\tilde{\theta}_0 = [0 \quad b]^T. \tag{7.9}$$



(a) *(b)*

Figure 7.9: *Cubic model structure. The performance of estimates from an instrumental variables and a least-squares estimation, evaluated as (a) a preinverse and (b) a postinverse.* A_{LS}*,* A_{IV}*,* C_{LS} *and* C_{IV} *are evaluated and compared to the true value. For the preinverse, also the B2 estimate is evaluated. The B2 estimate is the best preinverse, followed by the* C_{LS}*. For the postinverse,* C_{LS} *is the clear winner.*

For METHOD C, the expanded structure corresponds to an inverse model structure containing a linear term,

$$\hat{u} = \tilde{\vartheta}_1 y + \tilde{\vartheta}_3 y^{1/3} = \begin{bmatrix} y & y^{1/3} \end{bmatrix} \begin{bmatrix} \tilde{\vartheta}_1 \\ \tilde{\vartheta}_3 \end{bmatrix} \triangleq \phi^T \tilde{\vartheta}. \tag{7.10}$$

The true parameters for the inverse model is

$$\vartheta_0 = \begin{bmatrix} \frac{1}{b^{1/3}} \end{bmatrix} \tag{7.11a}$$

$$\tilde{\vartheta}_0 = \begin{bmatrix} 0 & \frac{1}{b^{1/3}} \end{bmatrix}^T \tag{7.11b}$$

showing that the true inverse is contained in the model sets given by these model structures for a specific choice of parameters.

In this chapter, the forward models will use θ as a parameter and the inverse models ϑ. The *tilde* ($\tilde{\cdot}$) denotes an expanded model with an additional linear term. See also Table 7.1.

The model structure with an extra linear term is motivated by analyzing the expected value of the output

$$y = b(u + w)^3 + v = b(u^3 + 3u^2 w + 3uw^2 + w^3) + v.$$

Table 7.1: *Notation in the cubic case study.*

Structure	Model type	
	Forward	Inverse
Cubic	θ	ϑ
Cubic & linear	$\tilde{\theta}$	$\tilde{\vartheta}$

The expected value with respect to the noise w is

$$\begin{aligned} E[y] &= E\left[b(u^3 + 3u^2w + 3uw^2 + w^3) + v\right] \\ &= bu^3 + 3bu^2E\left[w\right] + 3buE\left[w^2\right] + bE\left[w^3\right] + E\left[v\right] \\ &= bu^3 + 3bu\sigma_w^2 \end{aligned}$$

since both w and v are zero-mean white noises. This means that the expected value of the parameter vector is

$$E\left[\tilde{\theta}\right] = [3b\sigma_w^2 \quad b]^T. \tag{7.12}$$

and the linear term is nonzero in the MSE, compared to the true $\tilde{\theta}_0 = [0 \quad b]^T$. Here, $\left[\tilde{\theta}\right] = [3b\sigma_w^2 \quad b]^T = [0.48 \quad 4]^T$.

7.2.1 Method A

Just like the case with the true structure as an inverse, the model (7.8) can be written as a linear regression, and both least squares and instrumental variables methods can be used to estimate the parameters. The instruments are defined in (7.7). In this section, Method A is evaluated for LS and IV identification. Since the model structure with an additional term is no longer as easily inverted, the forward model will be evaluated only based on the estimation of the parameters.

The parameter values from Monte Carlo simulations for $\tilde{\theta}_{\text{LS}}$ and $\tilde{\theta}_{\text{IV}}$ are presented in Figure 7.10. In this case (with the regressors and variables chosen), the two methods coincide, and both methods lead to a good estimate of b, $b_0 = 4$ (boxes 2 and 4), and $\hat{\tilde{\theta}}$ is close to (7.12).

7.2.2 Method C

The inverse models $\tilde{\vartheta}$ using Method C, based on LS and IV methods, are also evaluated as preinverse and postinverse. The results are shown in Figure 7.11 where the linear parameter in both methods are close to zero and the true $\tilde{\vartheta}_0 = \left[0 \quad {}^1/_{b^{1/3}}\right]^T \approx [0 \quad 0.63]^T$. The C_{LS} method underestimates $\hat{\tilde{\vartheta}}_{\text{LS}}$ compared to the true value $\tilde{\vartheta}_0 \approx [0 \quad 0.63]^T$. $\hat{\tilde{\vartheta}}_{\text{IV}}$ is closer to the true value.

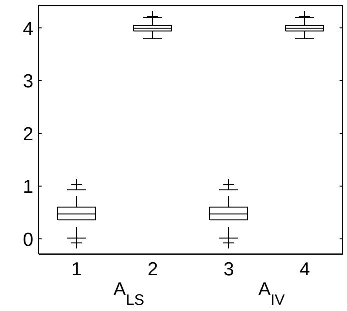


Figure 7.10: *Cubic and linear model structure,* METHOD *A. The estimates of $\tilde{\theta}$ using A_{LS} and A_{IV} methods. For the choice of regressors and instruments in this case, the* LS *and the* IV *methods coincide. Here,* $\tilde{\theta}_0 = [3b\sigma_w^2 \quad b]^T = [0.48 \quad 4]^T$.

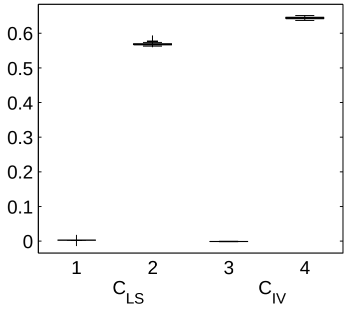


Figure 7.11: *Cubic and linear model structure,* METHOD *C. The estimates of $\tilde{\vartheta}$ using C_{LS} and C_{IV} methods. The true value is* $\tilde{\vartheta}_0 \approx [0 \quad 0.63]^T$.

7.2.3 Method B2

The iterative method is described in Section 5.2.5, where the system is used in the measurement loop, using repeated experiments. The step length Δ_ϑ was fixed. Here, the solution is found in around 50-70 iterations, but without a stopping criterion the algorithm will run through all $N_{\text{it}} = 100$ iterations. The number of function evaluations is $(n+1)N_{\text{it}}$ where n is the dimension of the parameter vector (for each iteration: one in each step direction and one final evaluation with the updated parameter vector). A gridding was also performed for evaluation purposes (leading to $d_1 \cdot d_3$ function evaluations, where d_1 is the number of values evaluated for ϑ_1 and d_3 is the number of values evaluated for ϑ_3). In this case, the gridding uses 1659 experiments, and the iterative method 300.

Figure 7.12 shows the mean squared error (MSE, defined in (4.11)) of the iterative method and the grid method along with the true value. It is clear that the iterative method does not achieve as good results as the gridding method, but it outperforms the true inverse in all Monte Carlo simulations, and the computational load is much smaller than for the gridding method. Since the grid method evaluates all values (within a predefined range) of the parameter space, it is not surprising that it achieves the best results. The estimate from method B2 performs almost as well as gridding, and always better than the true parameter value, illustrating that repeated experiments can improve the performance.

Figure 7.13 shows the value of $\tilde{\vartheta}$ for one simulation for the iterative solution and gridding as well as the true value. It can be seen that the iterative method converges rather quickly, but without a stopping criterion the method will just continue until the iteration counter reaches N_{it}. A well-formulated stopping criterion will stop the procedure earlier or continue longer if needed, but has been outside the scope of this thesis.

Figure 7.14 shows the path of the $\tilde{\vartheta}$ on top of the contours of the minimization criterion (based on the gridding method). The jaggedness is a consequence of the noise present in the Monte Carlo runs. The rhombic shape of the $\tilde{\vartheta}$ trail is a consequence of the choice of search path, where $\tilde{\vartheta}$ is always updated in pairs, with a positive or negative increment in each direction. With a variable step length in each separate direction this would not be the case, and the iterative method would probably find a better optimum. However, this is outside the scope of this thesis and will not be further explored.

7.2.4 Comparisons between the methods

The results of the estimation using methods C_{LS} and C_{IV} and B2 are presented in this section. Since the model structure with an additional term is no longer as easily inverted, the forward model will be evaluated only based on the estimation of the parameters. The inverse models $\tilde{\vartheta}$ from Method C, based on LS and IV methods, are also evaluated as postinverses.

Figure 7.15 shows the results of the model estimation and shows the estimates of b from the different methods. For methods B2, C_{LS} and C_{IV}, the estimates $\hat{b}$ have been produced by taking $\hat{b} = \frac{1}{\vartheta_3^3}$. The estimates based on the forward

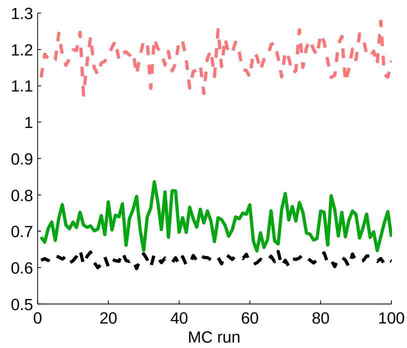


Figure 7.12: *Cubic and linear model structure, iterative* M*ETHOD* B2*. The mean square error of the different methods. The dashed pink line shows the evaluation of the true $\tilde{\vartheta}$, the green line is the iterative method and the dashed black shows the best choice from gridding the parameter space. It is clear that the gridding method is the best, followed closely by the iterative method presented here, and that they both outperform the true inverse, in all Monte Carlo runs.*

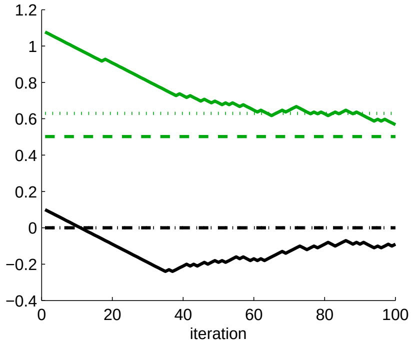


Figure 7.13: *Cubic and linear model structure, iterative* M*ETHOD* B2*. The value of $\tilde{\vartheta}$ in one Monte Carlo run. The solid line shows the iterative method, the dashed line is the grid-based method and the dotted line shows the true values. Black is the linear term $\tilde{\vartheta}_1$ and the cubic term $\tilde{\vartheta}_3$ is plotted in green.*

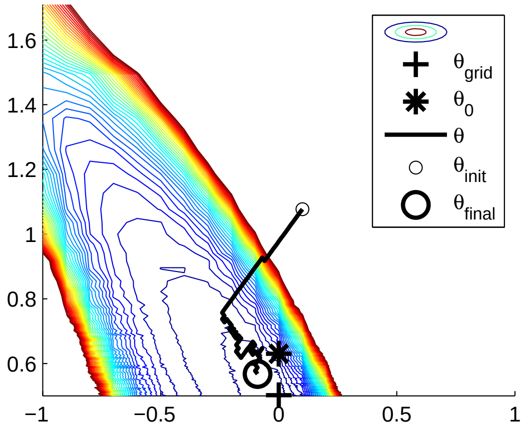


Figure 7.14: *Cubic and linear model structure, iterative* Method B2*. A path of $\tilde{\vartheta}$ in black, on top of the border lines of the cost function, based on a gridding approach. The plus sign is the minimum found through gridding and the star marks the true parameter value. The small and large circles mark the initial and final values of $\tilde{\vartheta}$ using the iterative method* Method B2*. The rhombic shape of the $\tilde{\vartheta}$ trail is the consequence of the choice of search path, where $\tilde{\vartheta}$ is always updated in pairs, with a positive or negative increment in each direction. With a variable step length in each separate direction this would not be the case, and the iterative method would probably find a better optimum.*

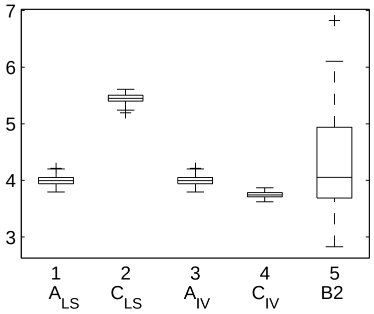


Figure 7.15: *Cubic and linear model structure. The estimate of the cubic parameter b, with $b_0 = 4$. The METHOD A estimates are closest to b_0, C_{LS} and B2 are overestimating and C_{IV} is underestimating the value. For methods B2, C_{LS} and C_{IV}, the estimate $\hat{b}$ has been produced by taking $\hat{b} = \frac{1}{\vartheta_3^3}$.*

METHOD A are close to $b_0 = 4$, whereas METHOD C either overestimates ($\mathrm{C_{LS}}$) or underestimates ($\mathrm{C_{IV}}$) $\hat{b}$. METHOD B2 leads to a much larger variance in the parameter estimate than the other methods, which could be caused by the method getting stuck in a local minimum. In Figure 7.16 the MSE results are shown for validation data from (a) preinversion and (b) postinversion. The $\mathrm{C_{LS}}$ method performs significantly better than the $\mathrm{C_{IV}}$ method or using the true value. The $\mathrm{C_{IV}}$ method performs slightly worse than the true values. The iterative METHOD B2 used as a preinverse performs better than the other methods.

7.3 Inverse identification in Hirschorn's method

In the case study above, the underlying structure of the system was known and model structures for the inverse model were based on that knowledge. For a small system, this can be done but in the general case, finding a structure for the matching inverse is nontrivial. Hirschorn's method, described in Section 3.3.2, is a way of finding an inverse to a general nonlinear system. Hirschorn's method gives a structure for the inverse, based on a model of the forward system. Exact linearization assumes full knowledge of the system, but one could imagine a situation where the structure of the nonlinear system is known, but there are unknown parameters. The identification can be done in several ways, corresponding to the methods described in Section 5.1.

METHOD A would correspond to measuring the input u and the output y, and identifying the unknown parameter values in the standard (forward) way. This estimated model could then be used to provide the inverse, since a model of the inverse system is available if a forward model is.

Since the exact linearization framework provides us with an inverse if the forward model is known, METHOD B1 does not really have an equivalence in this case – once the forward model is known, the exact inverse to match it is also

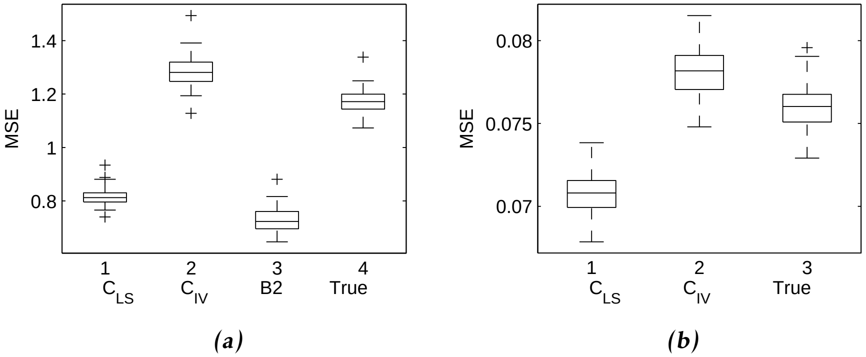


Figure 7.16: *Cubic and linear model structure. The* MSE *for the* METHODS *B2 and C, evaluated as (a) a preinverse and (b) a postinverse. For the preinverse, the iterative B2 method performs best, closely followed by the C_{LS} method.The C_{IV} method performs slightly worse than the true parameter value. For the postinverse, only* METHOD *C is evaluated and compared to the true value. The ranking of performance is the same as in the preinverse case.*

known. In the general case, this forward model could be used to estimate an approximate inverse. METHOD B2 could be used where the parameter is estimated using repeated experiments.

METHOD C would correspond to estimating the inverse $S^{\dagger}$ directly. The order of the input and the output are reversed in METHOD C, so that y is used as input and u as output. In Hirschorn's method, the inverse takes the reference r as input and the output is the control signal u. So, in order to find the inverse of $S^{\dagger}$, u is needed as output and the reference r as input. But, as the data was collected in open loop with no pre- or postdistorter, the signal r is not available, only u and y. Now, as in Section 3.3.2, assume that the system was actually preceded by a system $S^{\dagger}$, fed by a fictitious reference signal $\tilde{r}$, and that the overall behavior from $\tilde{r}$ to y is in fact linear with dynamics described by G_m. If this is true, then a signal $\bar{r} \approx \tilde{r}$ can be obtained by filtering y with $1/G_m$, and the system $S^{\dagger}$ can be identified using $\bar{r}$ as input and u as output, see Figure 7.17. So, this equals finding the inverse by using (a filtered version of) the output y as the input and u as output as in METHOD C. A benefit with Hirschorn's method is that it provides a parameterized inverse, so that the structure of this inverse system is already known.

Example 7.1: Parameter estimation for a Hirschorn inverse

Let us look at the example used in Example 3.4, where the goal is to obtain a

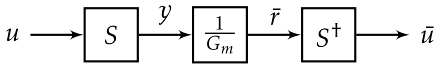


Figure 7.17: *Estimation of Hirschorn inverse model. By filtering the output y through the inverse dynamics of the desired dynamics G_m, an estimate of the reference signal $\bar{r}$ can be obtained. This means the reference signal $\bar{r}$ (or an estimate thereof) and the desired output u are available. These two signals can then be used to estimate the parameters needed in the inverse, as in METHOD C. The difference compared to Figure 3.9 is that there is no G_m block here. In Section 3.3.2, the goal was to obtain an overall behavior equal to the G_m dynamics. Here, the goal is to obtain the signals needed to perform the inverse system identification, in this case $\bar{u}$ (constructed from y) and u (available from measurements).*

linear response from reference r to output y. The nonlinear system is

$$\begin{aligned} \dot{x}_1 &= -x_1^3 + x_2 + w_1 \\ \dot{x}_2 &= -\alpha x_2 + u + w_2 \\ y &= x_1 \end{aligned} \tag{7.13}$$

with process noise $w_1 \in N(0, 0.1)$, $w_2 \in N(0, 0.05)$ and the input is a multisine. The parameter $\alpha = 0.8$ is unknown but the structure of the system is known.

To evaluate the different identification methods, a model has been estimated using measured data. The functions used to perform the grey-box modeling are `idnlgrey` and `pem` in the *system identification toolbox* in MATLAB. METHOD A uses input data $u(t)$ and output data $y(t)$ from the open-loop system. METHOD C uses the filtered output $\bar{r}$ as input data and $u(t)$ as output data. METHOD B2 uses repeated experiments, here a gridding is performed and the optimal value is determined.

The evaluation is done using a Hirschorn preinverse, where the parameter estimates are used in the preinverse, and the system uses the true value α_0. The MSE values are shown in Figure 7.18 for the true value, and the parameter estimates from METHODS A and B2. The estimate from METHOD C performs badly and is not shown here. The extra filtering and the more complicated model structure for the Hirschorn inverse seem to complicate the optimization for the inverse model using METHOD C. The MSE values are presented in Table 7.2, along with the parameter estimates $\hat{\alpha}$. The improvement shown is how much better the MSE is in percent, compared to using the true value in the inverse. It is shown that estimating the parameter in the forward model improves the performance slightly compared to using the true value, and that using METHOD B2 improves the performance even more.

Table 7.2: *MSE (4.11) of the Hirschorn identification example in Example 7.1.*

	y	True	METHOD A	METHOD B2
Estimate $\hat{\alpha}$		0.8	0.8454	1.20
MSE	0.5908	0.0281	0.0266	0.0215
Improvement [%]		0	5.6	23.5

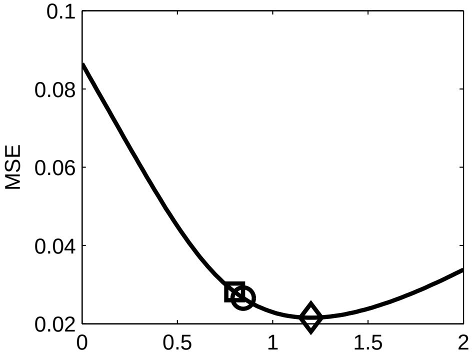


Figure 7.18: *The MSE of the Hirschorn preinverse in Example 7.1 for different identification methods. The MSE for different α in the preinverse are shown, along with the values using the true parameter value $\alpha = 0.8$ (square), the forward METHOD A estimate (circle) and the estimate from the repeated experiments using METHOD B2 (rhomb).*

7.4 Discussion on inverse system identification

In this first part of the thesis, we have looked at the estimation of inverse systems. There are three main methods, and it has been shown that the results and performance depend on the chosen method. There are cases when the various methods produce the same model, but in the general case this is not true. Therefore, it is necessary to consider the choice before estimating a model of an inverse system. System identification should be performed in the same setting as the model is intended to be used, and this is true also for inverse systems.

The discussions and analysis concern both preinversion and postinversion, where the position of the inverse is often given by the application. It has been shown that the identification of a forward system and a postinverse are more straightforward than the identification of a preinverse. For the preinverse case, the input signal to the system will change as it passes through the inverter, and this step can significantly change the properties of the signal. Therefore, one set of measurements is not enough to obtain the best performance. Furthermore, when there is noise present in the measurements, the true inverse is not optimal, and it can be beneficial to evaluate other structures.

In Part II of this thesis, focus is on preinversion, also called predistortion, of power amplifiers. The two most common methods for power amplifier predistortion are METHOD B1 (DLA) and METHOD C (ILA). METHOD C estimates a postinverse that can be used as a preinverse, and the identification process is straightforward which makes it easy to try out different model structures. However, if the goal is a preinverse, the commutation of the system and its inverse might be problematic for a nonlinear system. METHOD B1 has the advantage that it estimates a preinverse, but the identification process is more complicated than for METHOD C. We have suggested a modification of METHOD B1, denoted METHOD B2, where the preinverse is identified using the system in the loop during repeated measurements. This allows for multiple noise realizations and the altered (predistorted) input signal to correctly excite the system, but requires access to the system.

Although it has been illustrated in this first part of the thesis that using multiple experiments can be beneficial, METHOD B2 will not be used further in the second part. One of the goals of a predistorter is to be able to use it adaptively and update the parameter values to account for wear and temperature changes, and then it is crucial to have an online solution that does not require dedicated experiments. The benefits are also largest when the noise contribution is considerable, and this is not the case for the outphasing predistortion setup in Part II. The METHODS A, B1 and C will all be evaluated for the predistortion application.

Part II

Power amplifier predistortion

8 Power amplifiers

An electronic amplifier, or *power amplifier* (PA) is used to increase the power of a signal, so that the output is a magnified replica of the input. There are many different constructions of amplifiers, and they can be characterized by different measures such as gain, efficiency and linearity. Amplifiers are commonly used in many applications, such as audio applications and telecommunications, both in base stations and hand-held devices.

This chapter provides a basis to understand the amplifier related problems described in later chapters. It is by no means a complete description of PAs, but should be enough to understand this thesis. It also introduces the concepts of predistortion and linearization as well as the outphasing PA.

8.1 Power amplifier fundamentals

Today, wireless communication is used everywhere to transfer information. An important part of the technology is the possibility to transmit and receive the information, and the devices used are called *transmitter* (TX) and *receiver* (RX). The transmitter converts the information to an electrical signal suitable for the transmission in the given medium (in this case air, but in standard communication this can be a wire, fiber-optics, etc.). At the other end of the transmitting medium, a device is needed to receive the message and convert it into the original form – the receiver. This process of sending and propagating an information signal over a medium is called a *transmission*.

It is often desired that the equipment should be able to both send and receive information (a phone for example, where one can speak and listen), that is, a device that contains both a transmitter and a receiver. Such a circuit is called a *transceiver*. The physical circuit is connected to a *chip*.

By combining the receiver and transmitter into a transceiver, the circuits can

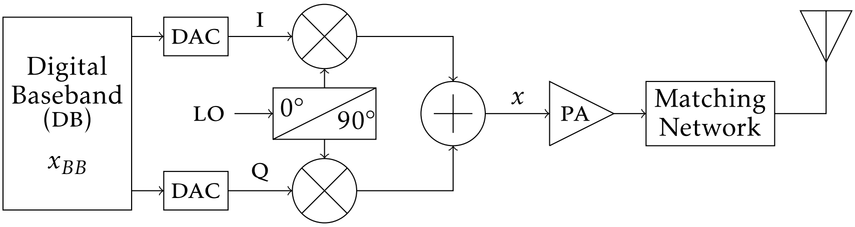


Figure 8.1: *Block diagram of a direct-conversion transmitter. The baseband signal (x_{BB}) is upconverted to radio frequencies by the modulator and passes through a* PA *before being sent to the antenna.*

be used for multiple purposes, reducing the number of components (and thus the cost) as well as the size of the chip, leading to more functionality per area. Such shareable components are antennas, oscillators, amplifiers, tuned networks and filters, frequency synthesizers and power supplies [Frenzel, 2003].

8.1.1 Basic transmitter functionality

A standard transmitter includes a *digital baseband* (DB), *digital-to-analog converters* (DACs), mixers (X) (further explained in Example 8.1), two *local oscillators* (LOs) that are 90° out of phase, a combiner, a power amplifier and a matching network before the antenna. The signal of interest, x_{BB}, is split into an in-phase channel, I, and a quadrature channel, Q,

$$x_{BB}(t) = I(t) + jQ(t) \tag{8.1}$$

by the DB, corresponding to the real (I) and imaginary (Q) parts of the signal, to generate two independent signals. Complex signals are commonly used in different modulation techniques in communications applications, see for example Frenzel [2003]. The I and Q signals are upconverted to the *radio frequency* (RF, ranging between 3 kHz and 300 GHz) carrier frequency, ω_c, and recombined, see Figure 8.1. The upconversion is done by a quadrature modulator, usually implemented by two mixers and two LO signals with a phase difference of 90°. The power of the recombined output signal,

$$x(t) = r(t)\cos(\omega_c t + \alpha(t)) \tag{8.2}$$

where

$$r(t) = \sqrt{I^2(t) + Q^2(t)} \tag{8.3}$$

and

$$\alpha(t) = \arctan(Q(t)/I(t)) \tag{8.4}$$

is often too low for transmission, and it has to pass through a power amplifier before being sent to the antenna.

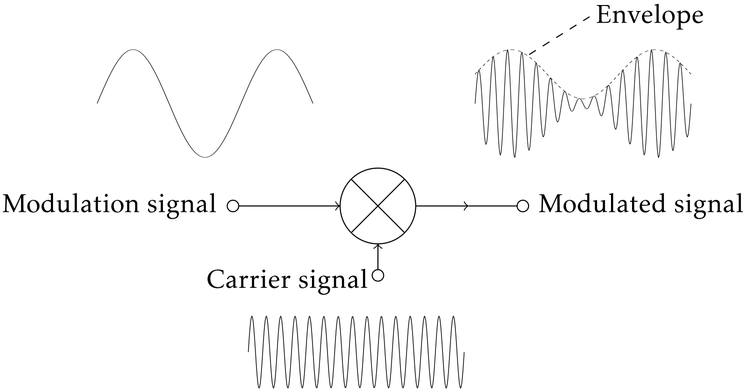


__Figure 8.2:__ Amplitude modulation. The information in the modulation signal is upconverted in the mixer to the carrier frequency (frequency of the carrier signal) and the shape (envelope) of the modulated signal contains the original information in the modulation signal.

Example 8.1: Amplitude modulation

Modulation is the process of varying the properties of a high-frequency signal, the *carrier signal* (usually a sine wave) with a *modulation signal* that contains the information to be transmitted. The modulation can be performed using a mixer, a component that multiplies the two (possibly shifted) inputs. When *amplitude modulation* (AM) is used, the information can be found in the amplitude of the modulation signal. The imaginary line that connects the peaks of the modulated signal is the information signal, and is called the *envelope*. Other common analog modulation techniques include *phase modulation* (PM) and *frequency modulation* (FM). Here, the envelope of the signal is kept constant but the phase shift or the frequency, respectively, of the carrier frequency is varied. These modulation techniques can also be combined into more complex modulation techniques.

For the example in Figure 8.2, the modulation (information) signal is a sine wave. The carrier is a sine wave of much higher frequency, and the modulated output is a high frequency signal where the shape of the envelope contains the information in the modulation signal.

The amplitude modulation in Example 8.1 is an analog modulation scheme that can be used for continuous signals. If the baseband signal is digital, a digital modulation is needed which will be introduced in Example 8.2.

Example 8.2: Digital modulation

One digital modulation scheme is *phase-shift keying* (PSK) that changes, modulates, the phase of the carrier signal. A digital modulation uses a finite number of distinct signals to represent digital data. In PSK, the phase is unique for each signal section, or *symbol*, that is transmitted. The demodulator, at the receiver end, should interpret the signal and map it back to the original symbol. This

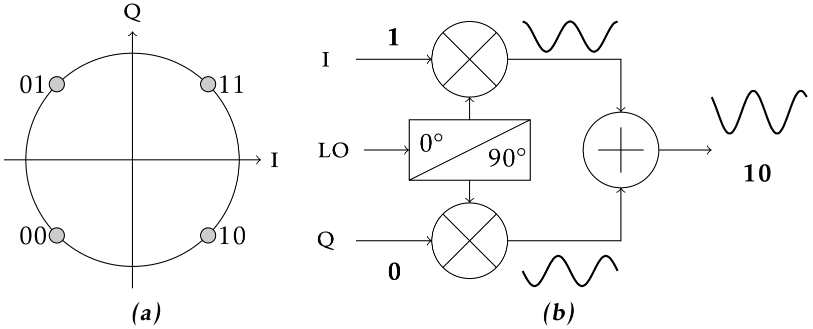


Figure 8.3: *(a) Constellation diagram for quadrature phase-shift keying, a digital modulation scheme. The four symbols represent the bits 00, 01, 11 and 10. (b) An example where the symbol 10 is to be transmitted. The I part is 1 and the Q part is 0. The bits are modulated by a carrier signal, a sinusoidal with a 90° phase shift between the I and Q parts, and the signals are added. Typically, the zero is coded as −1. The phase of the output is unique and can be mapped back to the I and Q parts, as seen in Figure 8.4.*

requires the receiver to be able to compare the phase of the received signal to a reference signal. Such a system is termed coherent.

One type of digital PSK modulation is *quadrature phase-shift keying* (QPSK) which uses four phases, and can encode two data bits per symbol. In a constellation diagram, the QPSK scheme has four points spread out around a circle, as seen in Figure 8.3a.

We will here look at an example where the symbol to be transmitted is 10. The IQ decomposition is done such that the odd-numbered bit (1) is the I component and the even-numbered bit (0) is the Q component, as seen in Figure 8.3b. The bits are modulated by the carrier signal, a sinusoidal with a 90° phase shift between the I and Q branches, and the signals are added. The resulting signal is unique, as seen in the bottom row of Figure 8.4, and can be mapped back to the I and Q components.

8.2 Power amplifier characterization

The choice of PA is a trade-off between different properties such as output power, efficiency and linearity, and will depend on the application. If power efficiency is an important property, such as in handheld devices where it will reflect directly on the battery time, a lower linearity might be accepted, whereas an audio amplifier, always connected to the power net, might focus more on the linearity and gain than on the efficiency. Any number of PAs can be cascaded in order to combine the benefits of each step.

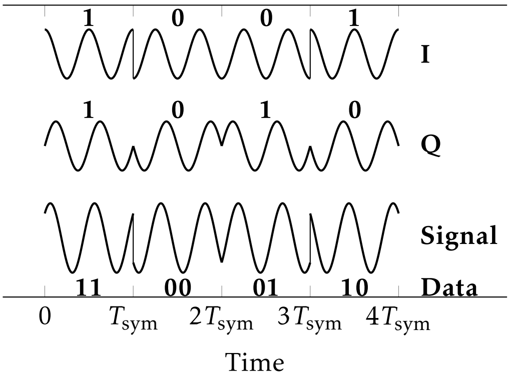


Figure 8.4: *The modulated signals in the IQ modulation, where the two carrier waves are sinusoidal with a 90° phase shift. The odd-numbered bits encode the in-phase (I) component and the even-numbered bits encode the quadrature (Q) component. The total signal is shown at the bottom, together with the mapping. The digital data transmitted by this signal is 1 1 0 0 0 1 1 0. T_{sym} is the symbol duration.*

8.2.1 Gain

An amplifier is of course supposed to amplify the input signal, and this property is described by the gain. The gain of an amplifier expresses the relationship between the input and the output [Frenzel, 2003], and is usually described by the voltage gain, A_V,

$$A_V = \frac{V_{out}}{V_{in}}, \tag{8.5}$$

where V_{in} and V_{out} are the input and output voltages, respectively. It can also be expressed by the power gain, A_P,

$$A_P = \frac{P_{out}}{P_{in}},$$

where P_{in} and P_{out} are the input and output powers, respectively, see Figure 8.5. The gain is usually expressed in decibels (dB), so that the power gain is

$$A_P = 10\log_{10}\left(\frac{P_{out}}{P_{in}}\right). \tag{8.6}$$

8.2.2 Efficiency

Another important property of a PA is the efficiency, which describes the amount of power needed to perform the amplification. A part of the input power will be dissipated in the circuit and can be counted as losses. The efficiency of a PA will

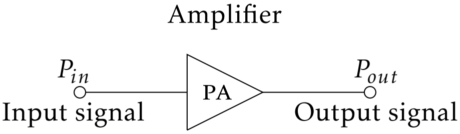


Figure 8.5: *Amplifier with input and output. The power gain is* $A_P = \frac{P_{out}}{P_{in}}$.

directly affect the battery time for a cell phone for example, and a high efficiency is desired.

The output efficiency, η, of a PA is defined as the ratio between the output power at the fundamental frequency, P_{out}, and the DC supply power of the last amplifier stage, P_{DC}, [Cripps, 2006]

$$\eta = \frac{P_{out}}{P_{DC}}, \tag{8.7}$$

and is often denoted *drain efficiency* (DE). Another efficiency measure is the *power added efficiency* (PAE),

$$\text{PAE} = \frac{P_{out} - P_{in}}{P_{DC}}, \tag{8.8}$$

where P_{DC} now represents the total power consumption of all amplifier stages constituting the whole PA [Razavi, 1998].

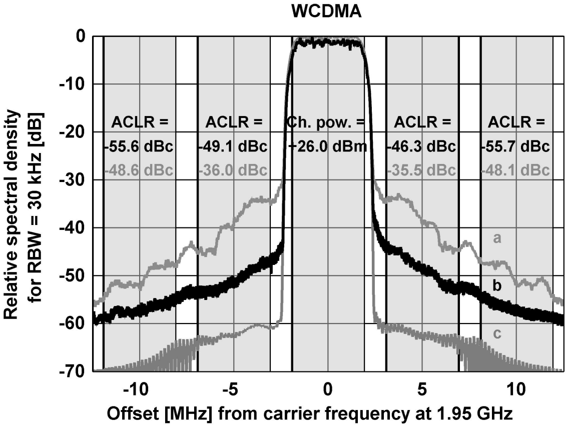
8.2.3 Linearity

By assigning transmissions different frequency bands, many transmissions can be done at the same time. For this setup to work, each of these transmissions must send only in the allotted slot, or *channel*. A radio transmission is allocated a frequency band with a certain bandwidth, ω_b, around a center frequency, f_c, where power may be transmitted. Any power falling outside the boundaries will cause disturbances in the neighboring channels. Broadening of the spectrum can be caused by, for example, nonlinearities in the PA. So to be practically useful in radio communications, PAs need to be linear. This means that the signal should be amplified in such a way that the output is an exact replica of the input but with a larger amplitude, and not be transferred to other frequencies. This is not possible in practice, and the level of linearity, or rather nonlinearity, is quantified by measures such as spectral mask, *adjacent channel power ratio* (ACPR) and *error vector magnitude* (EVM).

Spectral mask A spectral mask is a nonlinearity measure describing the amount of power that is allowed to be spread to adjacent frequencies. It is usually specified in decibel to carrier (dBc, the power ratio of a signal to a carrier signal, expressed in decibels) or in power levels given in dBm (power expressed in dB with one milliwatt as reference) in a specified bandwidth at defined frequency

***Table 8.1:** Spectral mask limitations for an EDGE signal*

Offset [kHz]	100	200	250	400	600	1000
Limit [dBc]	0	-30	-33	-54	-60	-60



© 2013 IEEE

***Figure 8.6:** Spectrum at 1.95 GHz for (a) measured output without DPD, (b) measured output with predistortion (linearization) and (c) the input signal for a WCDMA signal. The measured ACLR are printed in gray for the original output signal (without predistortion) and in black for the predistorted output. The gray shadows represent the passband in which the integration takes place.*

offsets [Fritzin, 2011]. Table 8.1 shows an example of the spectral mask limits for an EDGE signal.

Adjacent channel power ratio The ACPR is a measure that, like the spectral mask, describes the amount of power spread to neighboring channels. It is defined as the power in a passband away from the main signal divided by the power in a passband within the main signal [Anritsu, 2013]. The power at frequencies that are not in the main signal is the power transmitted in neighboring channels, i.e., the distortion caused by nonlinearities. Another measure is the *alternate channel power ratio*, which is defined as the ratio between the power in a passband two channels away from the main signal, over the power within the main signal.

The bandwidths and limits are connected to the standard used (for example WCDMA and LTE). For a WCDMA signal, the ACPR can be calculated by integrating the spectrum over a bandwidth of ω_b = 3.84 MHz at ±5 MHz distance from the

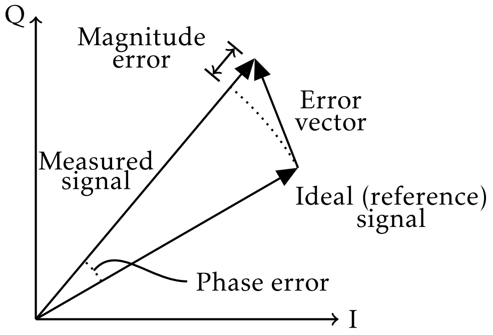


Figure 8.7: *Error vector magnitude (EVM) and related quantities.*

center frequency, as

$$\mathrm{ACPR} = \frac{\int\limits_{f_c+l\cdot 5-1.92}^{f_c+l\cdot 5+1.92} \mathrm{WCDMA}_{\mathrm{spectrum}}\,\mathrm{d}f}{\int\limits_{f_c-1.92}^{f_c+1.92} \mathrm{WCDMA}_{\mathrm{spectrum}}\,\mathrm{d}f}. \tag{8.9}$$

Here, f_c is the center frequency in the main signal and $l = \pm 1$ for the adjacent and $l = \pm 2$ for the alternate channel power ratio. ACPR is also named *adjacent power leakage ratio* (ACLR). An example of the ACLR can be seen in Figure 8.6.

Error vector magnitude The *error vector magnitude* (EVM) is a description of the quality of a signal with both magnitude and phase, such as the IQ signals as described in Section 8.1. The error vector is defined as the difference between the ideal signal and the measured signal [Agilent, 2013], see Figure 8.7.

Gain compression, AM-AM and AM-PM In traditional transistor-based power amplifier architecture, there is a point where a change in input amplitude does not result in a corresponding change in output amplitude, as illustrated in Figure 8.8. This phenomenon is called *gain compression* and leads to nonlinearities in the output, since different amplitudes of the input will be amplified in different ways.

Other nonlinearity measures describing the amplitude and phase distortion are the *amplitude modulation to amplitude modulation* (AM-AM) and the *amplitude modulation to phase modulation* (AM-PM). The AM-AM maps the input amplitude to the output amplitude (similar to the gain compression graph in Figure 8.8) and deviations from the straight line will result in output distortion. The AM-PM maps the input amplitude to the output phase, where an increasing input amplitude results in an additional output phase shift [Cripps, 2006]. Here, the optimal phase change is of course zero for all input amplitudes.

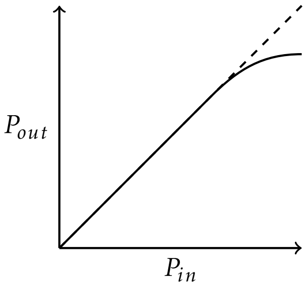


Figure 8.8: *Gain compression due to saturation in an amplifier transistor. The dashed line represents the ideal operation of the amplifier, while the solid line is the true output of the* PA *and a consequence of gain compression.*

8.3 Classification of power amplifiers

There are many different types of amplifiers, but they can be divided into two basic types; linear and switched amplifiers. See for example Frenzel [2003] and Jaeger and Blalock [2008] for a more thorough description of the different PA classes and the circuitry to implement them. Classical PAs usually assume both the input and the output to be sinusoidal, which limits the efficiency. If this assumption is disregarded, higher efficiency can be achieved [Razavi, 1998]. Here, the different classes are described.

8.3.1 Transistors

An important part of power amplifier implementation are the transistors, and we will start with a short overview of transistor functionality. A transistor is a device that uses a small signal to control a much larger signal. The two basic types of transistors are *bipolar junction transistors* (BJT s) and *field-effect transistors* (FETs). The structure of the commonly used FETs using semiconducting material has led to the name *metal-oxide-semiconductor field-effect transistor* (MOSFET). Depending on how the silicon is doped, the FETs can be either of p-type (PMOS) or n-type (NMOS), and thus have different conduction capabilities with respect to the applied voltages at the transistor terminals. Doping is the process of intentionally introducing impurities into an extremely pure semiconductor for the purpose of modulating its electrical properties. *Complementary metal-oxide-semiconductor* (CMOS) is a technology that typically uses complementary and symmetrical pairs of p-type and n-type MOSFETs for logic functions.

The FETs have three terminals, labeled gate (G), source (S) and drain (D), and a voltage at the gate controls the current between source and drain, see Figure 8.9. See for example Jaeger and Blalock [2008] for more insights into the workings and construction of transistors. For an NMOS transistor, a high voltage at the gate leads to a large current between source and drain, and for a small gate voltage, there is no current. For a PMOS transistor the relations are reversed, and a small gate voltage leads to a large current between source and drain, and a large gate

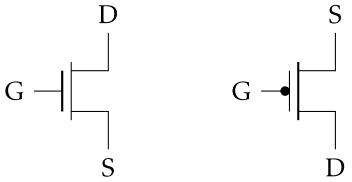


Figure 8.9: *The symbols of* NMOS *(left) and* PMOS *(right) and the associated ports. The ports are labeled gate (G), source (S) and drain (D).*

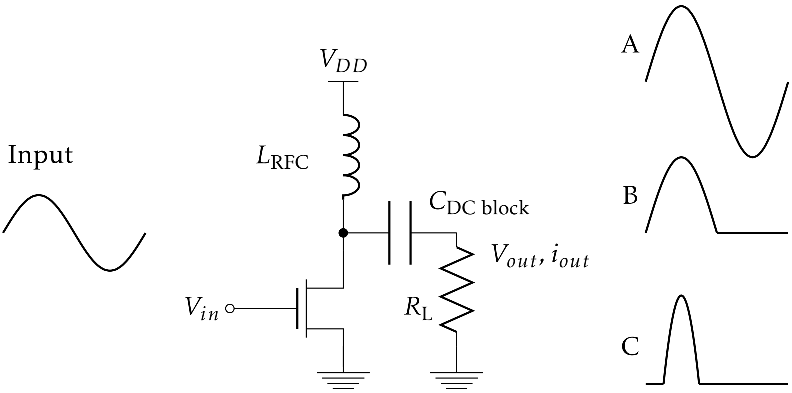


Figure 8.10: *Generic Class A/B/C power amplifier. The biasing of the transistor determines the conduction angle of the* PA*, as illustrated in the amplification of a sinewave input (left). The conduction angles are (from top to bottom)* 360° *for the Class A,* 180° *for the Class B and* 90° *for the Class C here.*

voltage opens the circuit and no current flows. Common uses for transistors are as amplifiers and switches, depending on the circuitry surrounding them.

8.3.2 Linear amplifiers

Linear amplifiers provide an amplified replica of the input. The drawback is that linear amplifiers often require a high power level and provide a rather low efficiency, as they operate far from their maximum output power where the linearity is limited.

Class A amplifiers

A Class A amplifier operates linearly over the whole input and output range. It is said to conduct for 360° of an input sine wave, that is, it will amplify for the whole of the input cycle, see Figure 8.10. Since the device is always conducting, a lot of power will be dissipated and the maximum achievable output efficiency is low, only 50%.

Class B amplifiers

In a Class B amplifier, the device is biased so that it only conducts for half of the input cycle, i.e., it has a conduction angle of 180°, see Figure 8.10. In this region the amplifier is linear, and at the rest of the input it is turned off, and the efficiency reaches $\eta = \pi/4 \approx 78.5\%$, with η defined in (8.7).

Class B amplifiers are often connected in a push-pull circuit, so that two amplifiers are connected, each of them conducting for half of the cycle, and together they conduct for the whole 360°. The efficiency is still the same, and in theory this will be a completely linear amplifier. In practice, however, if the biasing of the two amplifiers is not perfect, this will cause cross-over distortion at the time of switching between the two amplifiers [Jaeger and Blalock, 2008].

Class AB amplifiers

The Class AB amplifier uses the same idea as the Class B configuration with two amplifiers, but the amplifiers are slightly overlapping such that the cross-over distortion is minimized. Each amplifier thus has a larger conduction angle than the 180° of a Class B amplifier, but less than the full 360° of a Class A amplifier. This reduction of cross-over distortion is at the expense of efficiency.

Class C amplifiers

Class C amplifiers have a conduction angle smaller than 180°, typically between 90° and 150°, see Figure 8.10. This causes a very distorted output consisting of short pulses, and the amplifier usually has some form of resonant circuit connected to recover the original sine wave.

8.3.3 Switched amplifiers

The low efficiency of linear amplifiers is caused by the high power dissipation due to constant conduction. Switched amplifiers consist of transistors that are either *on* (conducting) or *off* (nonconducting). In the off state (cutoff state), no current flows so there is (almost) no dissipation. When the transistor is conducting, the resistance across it is very low, and so is the power dissipation.

The output of a switched amplifier is a square wave, which is passed through a filter to obtain a sinusoidal signal.

Class D amplifiers

A Class D amplifier consists of two transistors that alternately are on and off. The output is a *pulse-width modulated* (PWM) signal, which can be filtered to obtain the fundamental sine wave, see Figure 8.11. With ideal switches and ideal series resonant network (C_1 and L_1) stopping all frequencies but the fundamental tone, the theoretical maximum efficiency is 100%.

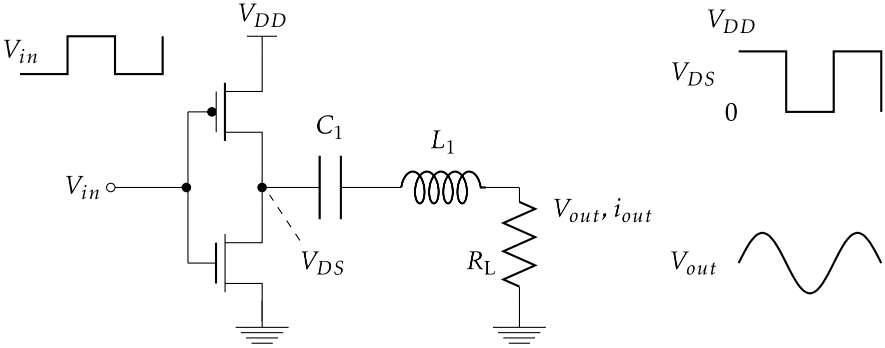


Figure 8.11: *Class D power amplifier.*

Class E amplifiers

In a Class E amplifier, only one transistor is used (compared to the two for Class D). By choosing a suitable load matching network, the drain current and voltage can be shaped to not overlap each other, making the theoretical efficiency 100%.

8.3.4 Other classes

There exist many other classes including Class F (a variation of the Class E amplifier) and Class S (a variation of switching amplifier using pulse-width modulation), see for example Frenzel [2003].

8.4 Outphasing concept

An outphasing amplifier is based on the idea that a nonconstant envelope signal, with amplitude and phase information, can be decomposed into two constant envelope signals with phase information only. The two signals can then be amplified separately by two nonlinear and highly efficient amplifiers and recombined, as presented in Cox [1974] and Chireix [1935]. The output signal will be amplitude and phase modulated, just like the input signal. Another name for the outphasing concept is *linear amplification with nonlinear components* (LINC).

The outphasing concept is illustrated in Figure 8.12. Here, a nonconstant envelope-modulated signal

$$s(t) = r(t)e^{j\alpha(t)} = r_{\max}\cos(\varphi(t))e^{j\alpha(t)}, \quad 0 \le r(t) \le r_{\max} \tag{8.10}$$

where $r_{\max}$ is a real-valued constant, and α and φ are angles, is used to create two constant-envelope signals, $s_1(t)$ and $s_2(t)$. This is done in the *signal component*

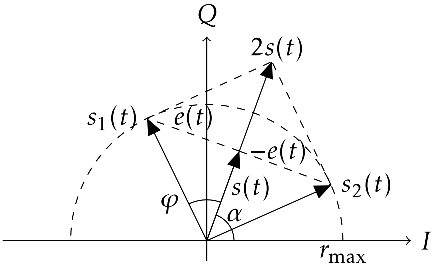


Figure 8.12: *Outphasing concept and signal decomposition.*

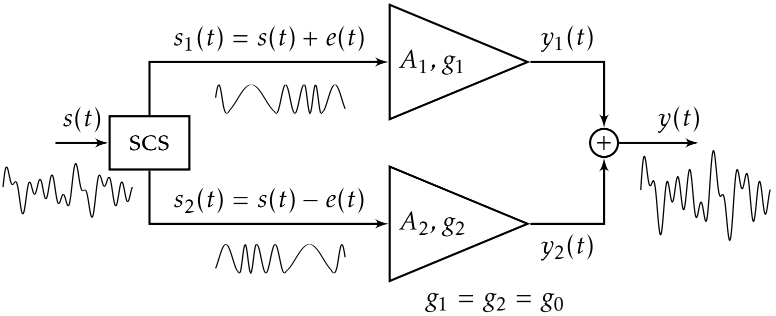


Figure 8.13: *Illustration of ideal power combining (the plus sign) of the two constant-envelope signals. The signals are amplified separately by two non-linear amplifiers, A_1 and A_2, and recombined to an amplified replica of the input $s(t)$.*

separator (SCS) in Figure 8.13 as

$$\begin{aligned} s_1(t) &= s(t) + e(t) = r_{\max} e^{j\alpha(t)} e^{j\varphi(t)} \\ s_2(t) &= s(t) - e(t) = r_{\max} e^{j\alpha(t)} e^{-j\varphi(t)} \\ e(t) &= js(t)\sqrt{\frac{r_{\max}^2}{r^2(t)} - 1}. \end{aligned} \tag{8.11}$$

The outphasing signals $s_1(t)$ and $s_2(t)$ contain the original signal, $s(t)$, and a quadrature signal, $e(t)$, and are suitable for amplification by switched amplifiers like Class D/E. By separately amplifying the two constant-envelope signals and combining the outputs of the two individual amplifiers as in Figure 8.13, the output signal is an amplified replica of the input signal.

In theory, the two quadrature signals will cancel each other perfectly in the combiner, but in practice, implementation imperfections and asymmetries will cause distortion. Letting g_1 and g_2 denote two positive real-valued gain factors, in each branch $s_1(t)$ and $s_2(t)$, and δ denote a phase mismatch in the path for $s_1(t)$,

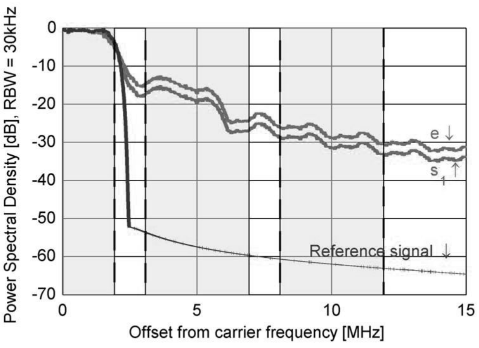


Figure 8.14: The bandwidth of the quadrature signal $e(t)$, and thus the outphasing signals $s_1(t) = s(t) + e(t)$ and $s_2(t) = s(t) - e(t)$, is much larger than that of the original signal $s(t)$. Any remainders of the quadrature signal caused by PA imperfections will thus lead to degraded ACLR and reduced margins to the spectral mask. From Fritzin [2011] with permission.

it is clear from

$$\begin{aligned} y(t) &= g_1 e^{j\delta} s_1(t) + g_2 s_2(t) \\ &= [g_1 e^{j\delta} + g_2] s(t) + [g_1 e^{j\delta} - g_2] e(t), \end{aligned} \tag{8.12}$$

that besides the amplified signal, a part of the quadrature signal remains. As the bandwidth of the quadrature signal, $e(t)$, is larger than the original signal, $s(t)$, see Figure 8.14, this would lead to a degraded ACLR and reduced margins to the spectral mask [Birafane and Kouki, 2005, Birafane et al., 2010, Romanò et al., 2006].

The phase and gain mismatches between $s_1(t)$ and $s_2(t)$ must be minimized in order not to allow a residual quadrature component to distort the spectrum or limit the *dynamic range* (DR),

$$c_{DR} = 20 \log_{10}\left(\frac{\max(|y(t)|)}{\min(|y(t)|)}\right) = 20 \log_{10}\left(\frac{|g_1 + g_2|}{|g_1 - g_2|}\right), \tag{8.13}$$

of the PA [Birafane and Kouki, 2005]. The DR defines the ratio of the maximum and minimum output amplitudes the PA can achieve. However, all phases and amplitudes within the DR can be reached by changing the phases of the outphasing signals $s_1(t)$ and $s_2(t)$.

Since an outphasing amplifier only uses two states (on or off), it will not experience problems like the conventional PAs such as gain compression (see Section 8.2.3), where the peak amplitudes are clipped. Instead, the smallest amplitudes will not be properly amplified in outphasing PAs, since any mismatch of the amplifier gains will make it impossible for $s_1(t)$ and $s_2(t)$ to cancel each other, compare Figures 8.12 and 8.15. Thus, the DR in an outphasing PA limits the spectral performance when amplifying modulated signals.

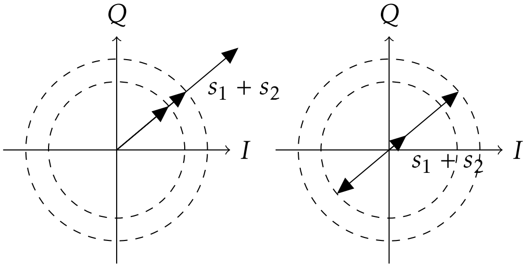


Figure 8.15: *The outphasing concept when the gain factors g_1 and g_2 are not identical. In the left figure, the outphasing signals are parallel and the resulting output is the maximal one. In the right figure, the nonidentical gain factors cannot cancel each other, and some remains are left. The dynamic range (the ratio between the maximal and minimal amplitudes, see* (8.13)*) of the power amplifier will determine the limit of small amplitude clipping.*

As the output of a Class D stage can be considered as an ideal voltage source whose output voltage is independent of the load [Yao and Long, 2006], i.e., the output is connected to either V_{DD} or *GND*, the constant gain approximations g_1 and g_2 are appropriate and make Class D amplifiers suitable for nonisolating combiners like transformers [Xu et al., 2010]. The implementation of the combiner (the plus sign in Figure 8.13) can be done in a multitude of ways, see for example Fritzin [2011] and the references therein.

8.5 Linearization of power amplifiers

The increased use of nonlinear amplifiers in an attempt to improve efficiency also requires new linearization methods. As described in Chapter 3, there are different approaches to do linearization. Since it is desirable to work with the original signal, and not with the amplified output of the PA, a prefilter is desired, also called a *predistorter* [Kenington, 2000]. Originally, these predistorters consisted of small analog circuits, but now they are often implemented in a *look-up table* (LUT) or a *digital signal processor* (DSP). Such an implementation is called a *digital predistorter* (DPD). The idea behind predistortion is presented in Figure 8.16. The predistortion can be divided into two parts, the construction of the predistorter functions and the implementation of the obtained DPD.

The implementation of predistortion methods entails further considerations, and as concluded in Guan and Zhu [2010], "different methodologies or implementation structures will lead to very different results in terms of complexity and cost from the viewpoint of hardware implementation". An implementation using a look-up table will grow quickly with the resolution of the DPD, and thus needs a large chip area, but avoids the necessity of calculations needed in a polynomial implementation (leading to a larger power consumption). The implementation issues have not been considered in this thesis.

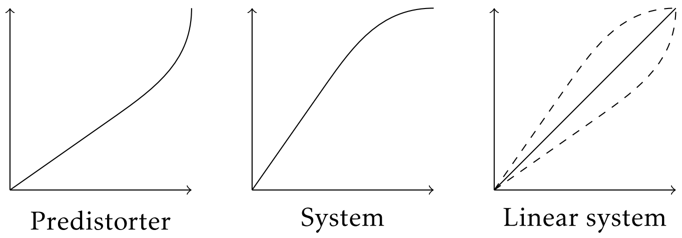


Figure 8.16: The main idea behind predistortion is to compensate for future nonlinearities and dynamics so that the overall system is linear.

An overview of different model structures for behavioral modeling of PAs, and implicitly predistorters, are presented in Ghannouchi and Hammi [2009].

8.5.1 Volterra series

The theory of p-th order Volterra inverses, introduced in Section 3.2.3, allows for the simpler postinverse (see METHOD C in Section 5.1) to be calculated and then used as the desired preinverse. This is used in the predistortion, or linearization, of for example RF power amplifiers. See also Section 3.2.2 and Section 4.2 for discussions on preinverse versus postinverse.

Since Volterra series consist of an infinite sum of integrals, the use of general Volterra theory is rather limited. To reduce the complexity a pruned, or truncated, version of the Volterra series is often used, where the memory length and/or the order of nonlinearity is limited. This heavily reduces the complexity of the sum, but the computational growth is still exponential/polynomial in memory length/order of nonlinearity, limiting the practical use of Volterra series. The memory of an inverse Volterra kernel is usually higher than the kernel of the original system [Tsimbinos and Lever, 1996].

Using pruned Volterra series as a means for modeling and predistortion of high-power amplifiers is presented in Tummla et al. [1997] and is shown to work for simulated data with memory length of 1 and nonlinearity order of 7. In Zhu et al. [2008], pruning techniques have been applied to drastically reduce the number of terms in the (discrete time) Volterra series and the method was applied to experimental data. A memory length of 2 and an order of nonlinearity of 11 was used. Volterra based predistorters have also been implemented in *field programmable gate array* (FPGA), shown in Guan and Zhu [2010]. An FPGA is a circuit that can be configured by the user and are used to implement complex digital computations.

8.5.2 Memory polynomials

A popular structure to use in PA predistortion is the memory polynomial or the generalized memory polynomial [Ding et al., 2004, Hussein et al., 2012, Landin et al., 2014]. Memory polynomials are less complex than Volterra series, and linear in the parameters. Parallel Wiener and parallel Hammerstein structures (see next section) are special cases of memory polynomials [Ding et al., 2004].

8.5.3 Block-oriented models

Since general nonlinear systems are very difficult to model, a common assumption is that the dynamics are linear, and that the nonlinearity is static, which gives a block-oriented model. This will be the case when there is, for example, a nonlinear actuator (due to saturation) in a linear control application.

A Hammerstein system consists of a static nonlinear system followed by a linear dynamic system and in a Wiener system, the static nonlinearity is at the output of the linear dynamics, see also Example 3.3. One way to broaden the use of the Hammerstein system is to use a more general *parallel Hammerstein* system, where multiple Hammerstein systems are branched. This structure is often used in modeling of power amplifiers, where a basic assumption is that the main part of the signal is amplified in a nonlinear way through the PA, and distortions are added to the output. The number of branches in the parallel Hammerstein structure determines the complexity of the model.

In Gilabert et al. [2006], a Wiener model of the PA has been used in combination with a Hammerstein structure predistorter, with memoryless nonlinearities followed by linear blocks using *finite impulse response* (FIR) and *infinite impulse response* (IIR) filters. The block-oriented structures have also been used in Gilabert et al. [2005] where both a Wiener and a Hammerstein structure have been evaluated for a PA model, combined with a Hammerstein DPD. The Hammerstein-Hammerstein setup presented a better performance. In Nader et al. [2011], parallel Hammerstein structures have been used for modeling both PA and DPD, and compared to peak-to-power power ration reduction. The implementation of a Hammerstein predistorter in FPGA technique is discussed in Xu et al. [2009] using a WCDMA input signal. Gan and Abd-Elrady [2008] use an IIR Hammerstein model structure for the PA and an IIR Wiener model for the DPD, such that the model structures of the power amplifier and the predistorter match.

8.5.4 Model structure considerations in B1 methods

When using METHOD B1 (DLA), there are two model structure choices: one for the power amplifier and one for the predistorter. These can be chosen to be the same or different. For general model structures, it can be natural to use the same structure for both models. This is done in Ding et al. [2004] where memory polynomials are used for both the PA and the DPD models, and in Isaksson and Rönnow [2007] reduced Volterra series are used.

For block-oriented structures, the choice has to be made regarding whether the same structure will be used or not. There are combinations of PA-DPD using Wiener-Hammerstein [Gilabert et al., 2006], Hammerstein-Wiener [Gan and Abd-Elrady, 2008] but also Hammerstein-Hammerstein [Gilabert et al., 2005] and parallel Hammerstein-parallel Hammerstein [Nader et al., 2011].

As illustrated in Example 3.3, in the noise free case, a Hammerstein system and the corresponding Wiener system (consisting of the inverses of the static nonlinearity and the dynamic system, respectively) commute and the exact inverse can be obtained. However, when there is noise present in the setup, this is no longer a guarantee. As shown in Examples 4.1-4.4, the true inverse is not necessarily the optimal inverse structure, so using an easier model structure (in parameter estimation, variance or complexity sense) can be beneficial. So, using a Hammerstein structure as both a model and its inverse, though counterintuitive at first, could be a good idea.

8.5.5 Outphasing power amplifiers

In outphasing PAs, there is no linearity between the individual outphasing signals, and any gain or phase mismatch between the two signal paths will cause spectral distortion, see for example Birafane and Kouki [2005] and Romanò et al. [2006]. Typical requirements are approximately 0.1 – 0.5 dB in gain matching and 0.2 ° – 0.4 ° in phase matching, which is very hard to achieve [Zhang et al., 2001].

The gain mismatch could be eliminated by adjusting the voltage supplies in the output stage [Moloudi et al., 2008], but this would require an extra, adjustable voltage source on the chip, which is undesirable. For the outphasing amplifier, all amplitudes within the dynamic range and all phases can be achieved by tuning the outphasing signals $s_1(t)$ and $s_2(t)$, see Figures 8.12 and 8.13. This can be used in the predistortion, so that the two signals are adjusted in a way to compensate for gain errors and possibly other unwanted effects in the PA.

Earlier predistortion methods for outphasing PAs compensate for the gain and phase mismatches in the signal branches. In Myoung et al. [2008], a mismatch detection algorithm has been evaluated using four test signals. These two-tone signals are used to calculate the amplitude and phase mismatches of the amplifier using a closed-form expression, later used for predistortion. Chen et al. [2011] present a *signal component separator* (SCS) implementation with a built-in compensation for branch mismatches in phase and amplitude. The SCS performs the decomposition of the original signal $s(t)$ into the outphasing signals $s_1(t)$ and $s_2(t)$, (8.11). By taking gain and phase mismatches into account, the SCS has a built-in predistorter.

Helaoui et al. [2008] discuss the impact of the combiner on the outphasing PA performance. The choice of combiner is a trade-off between linearity and power consumption. Nonlinearities can be introduced by a nonisolated combiner such that the output distortion depends on the input power. These nonlinearities were successfully reduced by the use of a predistorter.

The solutions in Myoung et al. [2008] and Chen et al. [2011] consider the gain mismatch between the two branches and compute the ideal phase compensation when the outputs are approximated as two signals with constant amplitudes. This is possible when there is no interaction between the amplifier stages. In this thesis, the outputs are still considered as two constant amplitude signals generating amplitude and phase distortion. Furthermore, an amplitude dependent phase distortion, occurring due to the interaction and signal combining of the amplifiers' outputs, is also considered.

Parts of the results in Chapters 9-11 can also be found in Fritzin et al. [2011a] and Jung et al. [2013]. The nonconvex algorithm, presented in Fritzin et al. [2011a], has been developed in Landin et al. [2012] to include a method for finding good initial values to the nonlinear optimization. However, the basic problem of nonconvexity has not been solved there and local minima still risk posing problems in the optimization. In Jung et al. [2013], the nonconvex formulation has been reformulated into a convex method. In this method, the PA model is estimated in a least-squares setting and an analytical calculation of the predistorter is used. Furthermore, a theoretical characterization of an outphasing PA is presented and form a basis for an ideal DPD. This characterization has also been used to obtain an estimate thereof.

9 Modeling outphasing power amplifiers

In this chapter, one way of modeling of the outphasing power amplifier using knowledge of the physical structure of outphasing amplifiers is presented. It consists of a new decomposition of the outphasing signals making use of the knowledge of the uneven amplification in the two branches, as well as a way to incorporate the possible nonlinearities in the branches.

Despite the fact that the PA is analog and the baseband model is in discrete time, the notation t is used to indicate the dependency on time. Based on the context, t may thus be a continuous or discrete quantity and denote the time or the time indexation. For notational convenience, the explicit dependency on time will be omitted in parts of this chapter and the following one.

9.1 An alternative outphasing decomposition

As mentioned in Chapter 8, the PA output signal $y(t)$ is a distorted version of the input signal. The nonlinearities are due to *(i)* the nonidentical gain factors g_1 and g_2, and *(ii)* nonlinear distortion in the amplifier branches. First, a novel decomposition will be described, accounting for the nonidentical gain factors g_1 and g_2, followed by a description of how these can be used in the modeling of the outphasing power amplifier. Since it is desired that the predistorter should invert all effects of the PA except for the gain, the signals can be assumed to be normalized such that

$$\max_t |s(t)| = \max_t |y(t)| = 1. \tag{9.1}$$

As described in Figure 8.12, the amplitude information of the original input signal $s(t)$ can be found in the angle between $s_1(t)$ and $s_2(t)$. Let

$$\Delta_\psi(s_1, s_2) = \arg(s_1) - \arg(s_2) \tag{9.2}$$

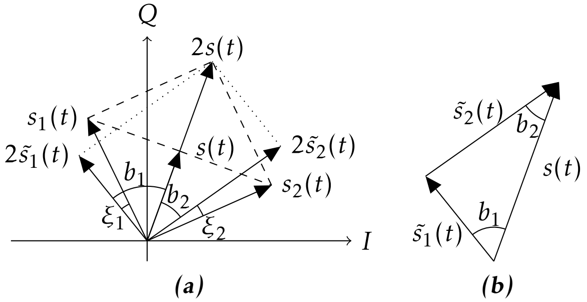


Figure 9.1: *(a) Decomposition of the input signal $s(t)$ into $s_1(t)$ and $s_2(t)$ when $g_1 = g_2 = g_0 = 0.5$ and into $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$ when decomposed as in (9.3) with nonidentical gain factors g_1 and g_2. (b) Trigonometric view of the decomposition of $s(t)$ using nonidentical gain factors. Note that $|\tilde{s}_k| = g_k, k = 1, 2$.*

denote the phase difference of the outphasing signals $s_1(t)$ and $s_2(t)$. Since the amplitude of the nondecomposed signal in the outphasing system is determined by $\Delta_\psi(s_1, s_2)$, this difference can be used instead of the actual amplitude in many cases. For notational convenience, Δ_ψ will be used instead of $\Delta_\psi(s_1, s_2)$, unless specified otherwise. Here, all phases are assumed unwrapped.

To describe the distortions caused by the imperfect gain factors, consider again the decomposition of $s(t)$ into $s_1(t)$ and $s_2(t)$ in (8.11). This is only valid when $g_1 = g_2$ but we can use an alternative decomposition of $s(t)$ into $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$ such that

$$\tilde{s}_1(t) + \tilde{s}_2(t) = s(t), \tag{9.3a}$$

$$|\tilde{s}_k| = g_k, \quad k = 1, 2, \quad \text{and} \tag{9.3b}$$

$$\arg(\tilde{s}_1) \geq \arg(\tilde{s}_2). \tag{9.3c}$$

Assuming knowledge of g_1 and $g_2 = 1 - g_1$ and given $s(t)$, the signals $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$ can be computed from (9.3). Let

$$b_1 = \arg(\tilde{s}_1) - \arg(s)$$

and

$$b_2 = \arg(s) - \arg(\tilde{s}_2)$$

denote the angles between the decomposed signals and $s(t)$ as shown in Figure 9.1a.

Figure 9.1b shows that the decomposition can be viewed as a trigonometric problem and application of the law of cosines gives

$$g_2^2 = g_1^2 + |s|^2 - 2g_1|s|\cos(b_1) \tag{9.4}$$

and

$$g_1^2 = g_2^2 + |s|^2 - 2g_2|s|\cos(b_2). \tag{9.5}$$

The angles b_1 and b_2 that define $\tilde{s_1}(t)$ and $\tilde{s_2}(t)$ can be computed from these expressions and can be viewed as functions of Δ_ψ since $|s| = r_{\max}\cos(\Delta_\psi/2)$. This means that the angles

$$\xi_1(\Delta_\psi) \triangleq \arg(\tilde{s_1}) - \arg(s_1) = b_1 - \frac{1}{2}\Delta_\psi \tag{9.6}$$

and

$$\xi_2(\Delta_\psi) \triangleq \arg(\tilde{s_2}) - \arg(s_2) = \frac{1}{2}\Delta_\psi - b_2 \tag{9.7}$$

can also be viewed as functions of Δ_ψ.

When the goal is to model the phase distortions in the two branches, this alternative way of defining the decomposition reflects the physical behavior better than the standard outphasing decomposition in (8.11). The output $y(t)$ can be decomposed in the same way to $y_1(t)$ and $y_2(t)$, taking the gain factors g_1 and g_2 into account.

9.2 Nonconvex PA model estimator

A first step on the way to model the outphasing PA is to observe that although the two branches are identical in theory, once implemented in hardware this will not be the case. Since the signals $s_1(t)$ and $s_2(t)$ are amplified by two different amplifiers, there might be a small amplification difference resulting in a gain offset between these signals, as well as a time delay stemming from the fact that $s_1(t)$ and $s_2(t)$ take different paths to the power combiner. With this insight, a first model structure with a gain mismatch between g_1 and g_2 and a phase shift δ in one branch is proposed. This leads to a model structure described by

$$y(t) = g_1 e^{j\delta} s_1(t) + g_2 s_2(t), \tag{9.8}$$

where g_1, g_2 and δ are real-valued constants.

When adding more complex behavior to the model structure, the structure of the physical PA must still be kept in mind. The separation of the two branches is still valid, but each branch can be affected by other factors than the gain difference and possible phase shift. As the amplitudes of the outphasing signals are fixed, a distortion based on changing the phase only in each branch is proposed.

To model an amplitude dependent phase shift while keeping in mind the constant amplitude of the signals $s_1(t)$ and $s_2(t)$, a model structure with an exponential function can be used. An amplitude-dependent phase distortion in $y_k(t), k = 1, 2$ (the two amplifier branches) can be written as

$$y_k(t) = g_k e^{j f_k(\Delta_\psi)} s_k(t), \quad k = 1, 2, \tag{9.9a}$$

$$y(t) = y_1(t) + y_2(t), \tag{9.9b}$$

as in Figure 9.2. Here, f_1 and f_2 are two real-valued functions describing the phase distortion

$$\arg(y_k) - \arg(s_k) = f_k(\Delta_\psi), \quad k = 1, 2, \tag{9.10}$$

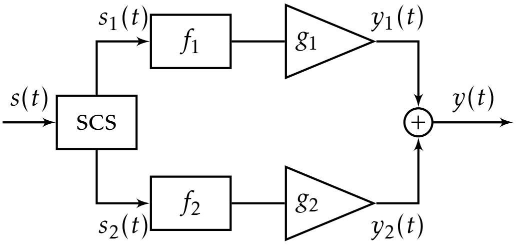


Figure 9.2: *A schematic picture of the amplifier branches setup. Note that the functions f_k, $k = 1, 2$, are not functions of the input to the block only but are used to show the general functionality of the PA with the separation of the two branches.*

in each signal path. Furthermore, g_1 and g_2 are the gain factors in each amplifier branch. Hence, an ideal PA would have $f_1 = f_2 = 0$ and $g_1 = g_2 = g_0$ and any deviations from these values will cause nonlinearities in the output signal and spectral distortion as previously concluded.

The functions f_1 and f_2 describing the phase distortion in the separate branches can be described by arbitrary basis functions. Here, polynomials

$$\hat{f}_k = p(\eta_k, \Delta_\psi) = \sum_{i=0}^{n} \eta_{k,i} \Delta_\psi^i, \quad k = 1, 2, \tag{9.11}$$

where

$$\eta_k = \begin{pmatrix} \eta_{k,0} & \eta_{k,1} \cdots & \eta_{k,n} \end{pmatrix}^T,$$

have been used as parameterized versions of the functions f_k, motivated by the Stone-Weierstrass theorem, see Rudin [1976, Theorem 7.26].

The model parameters in the given model structure are estimated by minimizing a quadratic cost function [Ljung, 1999] as in

$$\hat{\theta} = \arg\min_{\theta} V(\theta), \tag{9.12}$$

$$V(\theta) = \sum_{t=1}^{N} \left| y(t) - \hat{y}(t, \theta) \right|^2 \tag{9.13}$$

with

$$\hat{y}(t, \theta) = g_1 e^{j\, p(\eta_1, \Delta_\psi(s_1, s_2))} s_1(t) + g_2 e^{j\, p(\eta_2, \Delta_\psi(s_1, s_2))} s_2(t) \tag{9.14}$$

where $\theta = [g_1 \quad g_2 \quad \eta_1^T \quad \eta_2^T]^T \in \mathbb{R}^{2n+4}$, $y(t)$ is the measured output data and $\hat{y}(t, \theta)$ is the modeled output. The model (9.14) can be compared to the structure (9.9), where $y(t) = g_1 e^{j\, f_1(\Delta_\psi)} s_1(t) + g_2 e^{j\, f_2(\Delta_\psi)} s_2(t)$. This structure leads to a nonlinear and nonconvex optimization problem, so the minimization algorithm might find a local optimum instead of a global. In order to obtain a good minimum in a nonconvex optimization problem, it is essential to have good initial values, and one way to obtain these is presented in Landin et al. [2012]. Convexity and nonconvexity will be further discussed in Section 9.5.

Here, a model of the PA was estimated by minimizing a quadratic cost function measuring the difference between the measured and modeled output signal. This estimation problem involves solving a nonconvex optimization problem. However, using the knowledge of the structure of the outphasing amplifier, there is an alternative way which essentially only involves solving standard least-squares problems, presented in the next section.

9.3 Least-squares PA model estimator

The output distortions originate both from imperfect gain factors and nonlinearities in the amplifiers. Once the gain factor impact has been accounted for, the amplifier nonlinearities can be modeled. This means that the modeling optimization problem can also be rewritten as a *separable least squares* (SLS) problem, also presented in Jung et al. [2013]. A separable least squares problem is when one set of parameter enters the model linearly and one set nonlinearly. Given the nonlinear parameters, the linear part can be solved for efficiently, leaving a nonlinear problem of a lower dimension [Ljung, 1999]. See also Section 2.6.1 for a short introduction to SLS problems.

Often, the minimization is done first for the linear part and then the nonlinear parameters are solved for and this nonlinear minimization problem now has a reduced dimension. Here, the idea is to use knowledge of the gain factors to make a nonlinear transformation of the data using the decomposition (9.3). Once this decomposition is done, the minimization can be rewritten as a *least-squares* (LS) problem in the phase distortion in the two branches. This is not the usual SLS method since it involves a nonlinear transformation of the data, but the basic idea of separating out the nonlinear parameters to obtain a LS problem still applies. We will here explore two ways of estimating the gain factors g_1 and g_2. One is based on the dynamic range of the PA and the other is based on a parameter gridding of possible values of g_1 and g_2.

Assuming the gain factors to be known, we know what the phases of the outputs from the two outphasing branches must be in order for the two signals to sum up to the measured output $y(t)$. It is now possible to decompose the output $y(t)$ into $y_1(t)$ and $y_2(t)$, using the decomposition in Section 9.1. What is left to determine is the phase distortion in the branches, described by the functions f_k. Since the gain factor influence is handled by the alternative decomposition of $y(t)$, the phase distortion is now described by the difference between the phase of the input $s_k(t)$ and the output $y_k(t), k = 1, 2$ and this can be formulated as a least-squares problem.

Consider first the two gain factors g_1 and $g_2 = 1 - g_1$, where the relation between them comes from the normalization (9.1). Let

$$\begin{aligned} g_1 &= g_0 \pm \Delta_g, \\ g_2 &= g_0 \mp \Delta_g, \end{aligned} \tag{9.15}$$

where $\Delta_g \geq 0$ represents the gain imbalance between the amplifier stages and $g_0 = 0.5$. Inserting (9.15) into (8.13) gives

$$c_{DR} = 20 \log_{10}\left(\frac{g_0}{\Delta_g}\right). \tag{9.16}$$

Hence, the imbalance term Δ_g can be computed as

$$\Delta_g = g_0 \cdot 10^{-c_{DR}/20}, \tag{9.17}$$

making it possible to find approximations of g_1 and g_2 from the dynamic range of the output signal. The value of c_{DR} can be estimated from measurements as the ratio between the maximum and minimum output amplitudes. The estimate is noise sensitive, but this can be handled by averaging multiple realizations. These approximations are valid for input signals with large peak to minimum power ratios, like WCDMA and LTE, where the PA generates an output signal including its peak and minimum output amplitudes, i.e., its full dynamic range. If this is not fulfilled or the noise influence is too large, an alternative approach is to evaluate a range of values of g_1 and $g_2 = 1 - g_1$ and then solve the PA modeling problem for each pair of gain factors, as in the usual SLS approach.

Once the gain factors have been determined, $s(t)$ can be decomposed into $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$, and $y(t)$ into $y_1(t)$ and $y_2(t)$ using (9.3) to (9.5). Furthermore, the standard outphasing decomposition of $s(t)$ into $s_1(t)$ and $s_2(t)$ as in (8.11) will be used in the sequel.

Since the gain factor mismatch has been accounted for, it is now possible to determine the impact of the nonlinearities on the two branches. The phase distortion in each signal path caused by the amplifiers can thus be modeled from measurements of $s(t)$ and $y(t)$. Here, polynomials

$$p(\eta_k, \Delta_\psi) = \sum_{i=0}^{n} \eta_{k,i} \Delta_\psi^i, \quad k = 1, 2,$$

have been used as parameterized versions of the functions f_k, as in (9.11). Estimates $\hat{\eta}_{k,i}$ of the model parameters $\eta_{k,i}$ have been computed by minimizing a quadratic cost function, i.e.,

$$\hat{\eta}_k = \arg\min_{\eta_k} V_k(\eta_k), \quad k = 1, 2, \tag{9.18}$$

where

$$V_k(\eta_k) = \sum_{t=1}^{N} \Big(\arg\left(y_k(t)\right) - \arg\left(\tilde{s}_k(t)\right) - p\left(\eta_k, \Delta_\psi(s_1(t), s_2(t))\right)\Big)^2, \tag{9.19}$$

and

$$\eta_k = \begin{pmatrix} \eta_{k,0} & \eta_{k,1} \cdots & \eta_{k,n} \end{pmatrix}^T.$$

The cost function (9.19) can be motivated by the fact that the true functions f_k satisfy (9.10) when the amplifier is described by (9.9). Minimization of V_1 and V_2

are standard least-squares problems, which guarantees that the global minimum will be found [Ljung, 1999].

Once the LS problem is solved for each setup of g_1 and g_2, the problem of finding the best setup is now reduced to a one dimensional (possibly nonconvex) optimization problem over $g_1 (g_2 = 1 - g_1)$, which is much easier to solve than the original, multidimensional problem. A problem this small can be solved at a small computational cost.

The parameter estimates $\hat{\eta}_k$ define function estimates

$$\hat{f}_k(z) = p(\hat{\eta}_k, z), \quad k = 1, 2, \tag{9.20}$$

that, together with the gain factor estimates $\hat{g}_1$ and $\hat{g}_2$ describe the power amplifier behavior. The different steps are also described in Part A – Estimation of PA model in Algorithm 10.1, page 155.

The alternative decomposition described in Section 9.1 depends on the gain factors g_1 and g_2 via a nonlinear relation, but with these given, the problem is reduced to a LS-problem in the phase as in (9.19). If the gain factor estimation is done using the DR as in (9.15) and (9.17), this will result in two LS-problems to solve, and gridding of g_1 will result in $\left(\frac{g_{\max} - g_{\min}}{p_M} + 1\right)$ LS problems. The values $g_{\min}$ and $g_{\max}$ bound the values of g_1 and g_2 that one wants to evaluate and p_M is the precision, so that $g_1 \in [g_{\min}, g_{\min} + p_M, \ldots, g_{\max}]$ and $g_2 = 1 - g_1$. Compare to Algorithm 10.1, page 155, for notation. This is not the standard SLS method, since a nonlinear transformation of the data is done before solving the LS problem, but the separation of the linear and nonlinear parameters applies. This separation reduces the optimization to a number of LS problems and a nonlinear optimization in only one dimension, g_1 ($g_2 = 1 - g_1$ due to the normalization (9.1)). This is clearly a reduction from the nonlinear optimization in $2n + 4$ dimensions of the original problem.

9.4 PA model validation

As an evaluation of the different approaches presented above, the models have been compared. Figures 9.3-9.6 present the amplitude and phase of the measured output and the model output. The amplitude error $|y - \hat{y}|$ and the phase error $\arg(y) - \arg(\hat{y})$ are also included. The first simple model in (9.8), using only the gain factors g_1 and g_2 and a phase shift δ, is presented in Figure 9.3. The more complex model structure (9.14) is presented in Figures 9.4, 9.5 and 9.6, using the different modeling methods. The model obtained by the nonconvex approach as in (9.12)-(9.14) is presented in Figure 9.4. The LS method using (9.18)-(9.19) and the dynamic range to obtain the gain factors is presented in Figure 9.5. In Figure 9.6, the LS method using gridding of g_1 over a range of values and then determining the best fit is presented.

The more complex models perform very well, and fairly similarly. This is easier to see in Figure 9.7, where the errors for the different modeling methods are plotted together. Though the models all perform well, there are still errors. These errors are largest where the input amplitude is small, such as around time

Table 9.1: *PA model validation*

Method	g_1	g_2	$\|y - \hat{y}\|_2^2$, (9.13)
Delay only, model structure (9.8)	0.4911	0.5089	62.99
Nonconvex	0.4986	0.5014	0.9985
LS, grid	0.50	0.50	1.119
LS, DR	0.4994	0.5006	0.9781

152 μs and 156.5 μs. The result of the DR LS model is also presented in an IQ plot in Figure 9.8 where the signals are plotted in the complex plane. Also in this plot, the model shows a very good behavior, with a slightly worse performance for small amplitudes.

The gain factor estimates are presented in Table 9.1 together with the cost function (9.13) for the different methods. As seen in the rightmost column where $|y - \hat{y}|_2^2$, (9.13), is presented, the added model complexity with nonlinearities makes a large improvement in the model fit. The LS method using DR and the nonconvex method achieve rather similar results with the gridding LS method slightly behind. The results of the nonconvex method depends on the number of iterations used in the optimization.

Except for the first simple model, the other methods perform very similarly with a very good fit to validation data. This clearly shows that the nonlinear extension to the model has a significant impact on the model properties. This also means that the choice of method comes down to other considerations than the fit. The lack of guarantees of convergence to a global minimum of nonconvex optimization methods is a reason to avoid the method described in Section 9.2. If the LS method is chosen, this also entails the choice of gridding or using the dynamic range. Gridding is more robust against noise, since the DR estimation is done using only two measurements (the one with minimal and the one with maximal amplitude), so noise at either of these data points will have a large impact. A drawback with gridding is the risk of missing the best value, if the precision p_M (difference in g_1 and g_2) is chosen too large, or the performance is sensitive to changes in the the parameter. A smaller p_M, on the other hand, will increase the number of LS problems that need to be solved. Benefits and drawbacks for the dynamic range method are the opposite.

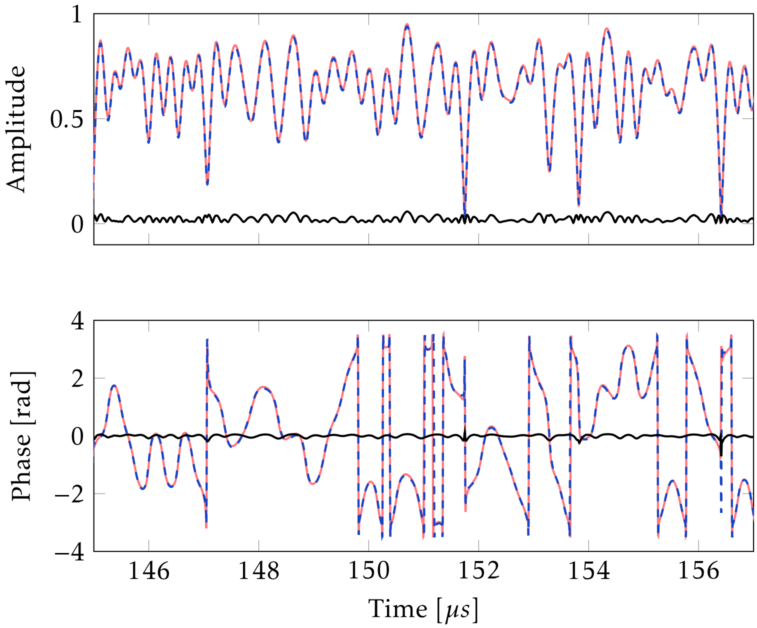


Figure 9.3: *Model validation of the model produced using the first structure* (9.8)*, with gain factors g_1 and g_2 and a phase shift δ only. The upper plot shows the amplitude of the measured signal (solid pink), the model output (dashed blue) and the error (black). The lower plot shows the phase.*

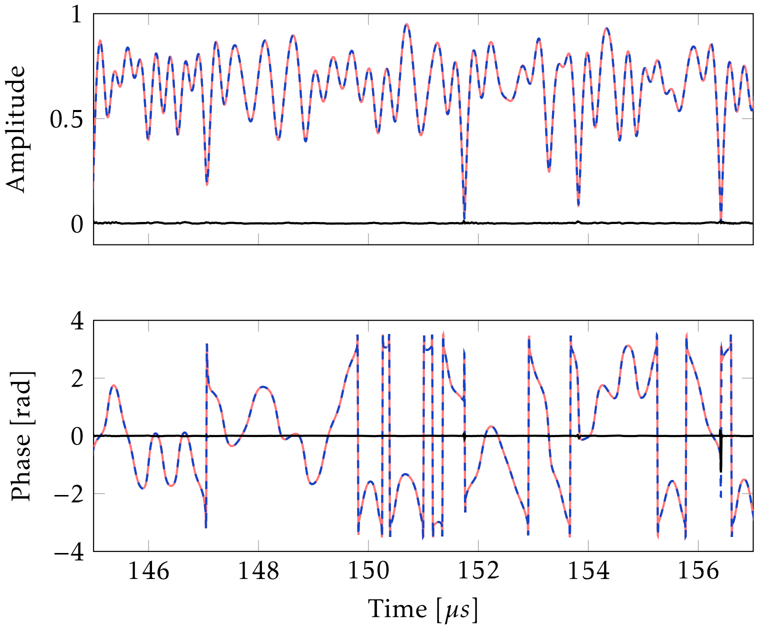


Figure 9.4: *Model validation of the model produced using the original, non-convex, optimization in* (9.12)-(9.14)*. The upper plot shows the amplitude of the measured signal (solid pink), the model output (dashed blue) and the error (black). The lower plot shows the phase.*

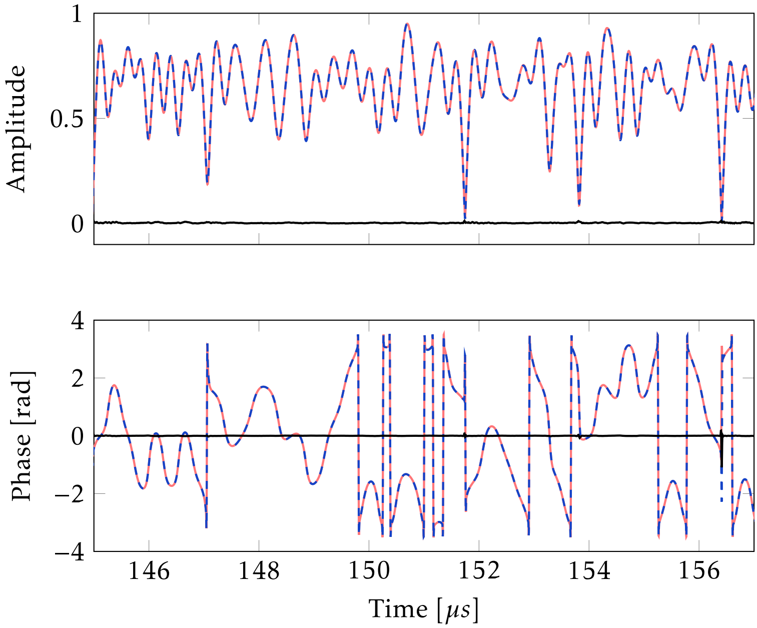


Figure 9.5: *Model validation of the model produced using the convex method in* (9.18)-(9.19) *and the dynamic range has been used to determine the gain factors as in* (9.17) *and* (9.15). *The upper plot shows the amplitude of the measured signal (solid pink), the model output (dashed blue) and the error (black line). The lower plot shows the phase.*

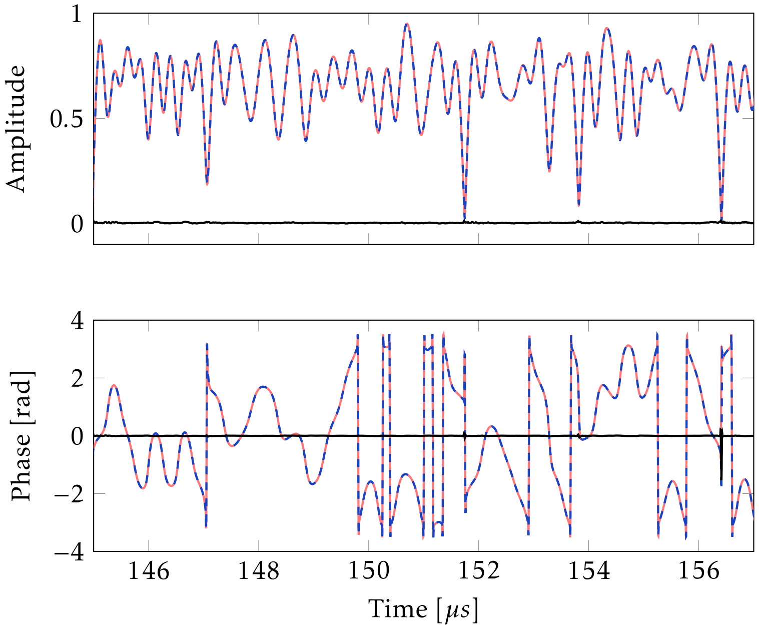


Figure 9.6: Model validation of the model produced using the convex method in (9.18)-(9.19) *and* g_1 *has been gridded in* $[g_{min}, g_{max}] = [0.4, 0.6]$ *with precision* $p_M = 0.005$. *The upper plot shows the amplitude of the measured signal (solid pink), the model output (dashed blue) and the error (black line). The lower plot shows the phase.*

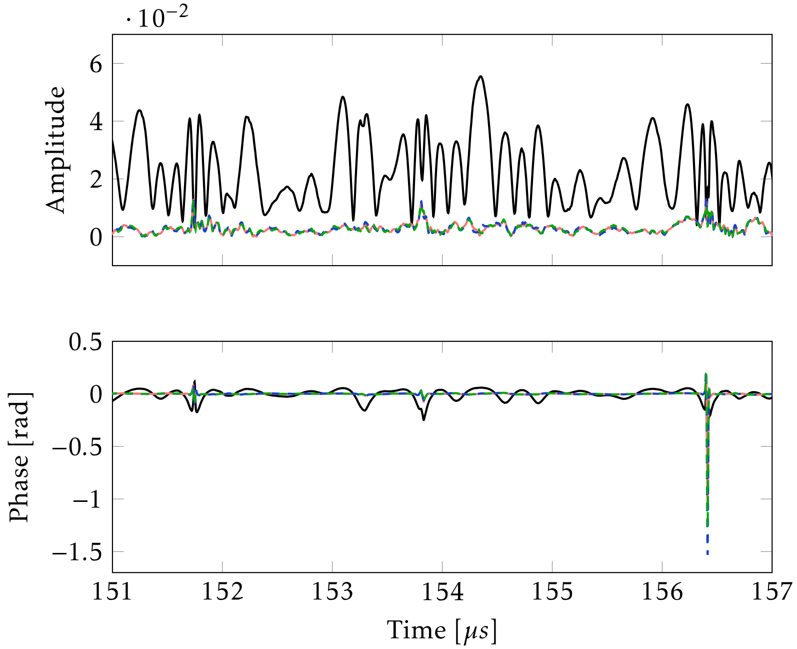


Figure 9.7: *A summary of the model errors of the different models. The upper plot shows the amplitude error* $|y - \hat{y}|$ *and the lower plot shows the phase errors* $\arg(y) - \arg(\hat{y})$. *The simple model* (9.8) *is plotted in black, the* LS *methods using* DR *in solid pink and gridding in dashed blue. The model obtained by the nonconvex method is plotted in a green dashed line. The three models describing a nonlinear behavior perform very well and in a very similar way, as seen in the figure where the lines are almost on top of each other.*

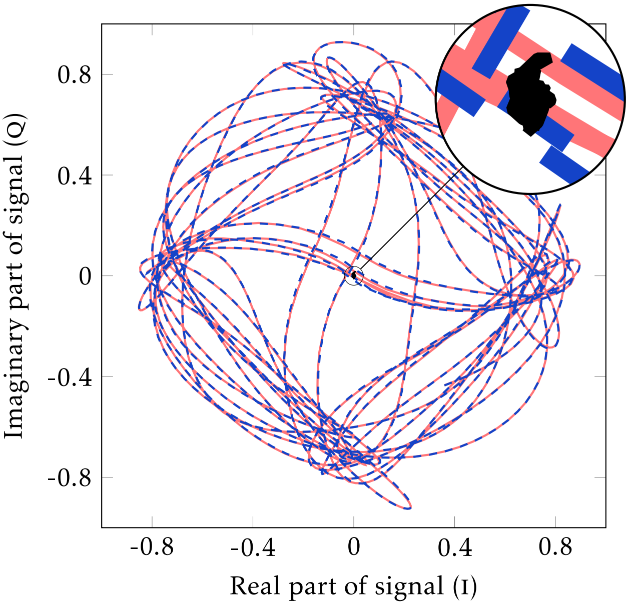


Figure 9.8: *IQ plot (imaginary part, Q, vs real part, I) of the measured signal (solid pink) and the model output (dashed blue) and the error $y - \hat{y}$ (black). The model was estimated by the LS method using DR to estimate g_1 and g_2. The zoom-in in the upper right corner is a ten times amplification of the error signal.*

The estimated phase distortion functions, $\hat{f}_1$ and $\hat{f}_2$, from the models can be plotted as functions of Δ_ψ and the results for a WCDMA signal for the different methods are rather similar. The function $\hat{\hat{f}}$ describes the phase change between the two outphasing signals at the output, and thus the amplitude change of the output. The phase distortion functions $\hat{\hat{f}}$ are presented in Figure 9.9 as deviations from the ideal phase distortion, which should be as close to zero as possible. The ideal phase distortion includes the compensation for nonequal gain factors. By this, it is clear that at amplitudes close to zero (Δ_ψ close to π), a zero distortion will not be possible for nonequal gain factors. In Figure 9.10, the functions f_k, $k = 1, 2$, are shown for the different methods. The methods achieve similar results, but at the expense of the number of computations in the nonconvex approach, where 25 000 function evaluations have been performed to achieve the optimum.

Even though the methods result in similar validation results, the largest differences are found close to the edges of the interval. In the WCDMA signal, 99.1% of the measured data points have $0.8 \leq \Delta_\psi \leq 3.0$, so the focus of the fit is where the most data points are. Compared to Figure 8.12 and (9.2), it is clear that the data points with a very large Δ_ψ (close to π) have a very small amplitude, and errors in the phase distortion modeling might not affect as much as the data points with a small Δ_ψ (large amplitude). It can thus be concluded that it could be more important to obtain a good model for small values of Δ_ψ than for large values (something that could be achieved by weighting functions). It can also be noted that, if the amplitude of the input had been used instead of the angle $\Delta_\psi = \arg(s_1) - \arg(s_2)$, more weight would have been put at the largest amplitudes. This is not done now since a large input amplitude equals a small Δ_ψ and vice versa.

In polynomial fitting, the agreement with the function f is often bad at the outer parts of the interval to be approximated. If one can choose the points at which the polynomial is to be fitted, Chebyshev points should be chosen, with more points at the outskirts of the interval [Dahlquist and Björck, 2008, p. 377-379]. Here, we are fitting a polynomial using the method of least squares, but the same reasoning holds. To obtain a smaller error at the peak power, more data points could have been collected there. Instead, the least-squares fitting focuses on fitting the overall performance, and hence more effort is made to obtain a small error in the parts where there is a larger point density. For the signals used in this thesis, this area of larger point density is in the center of the interval, where an improvement will be clearly seen in for example Figure 11.9. We will return to this subject in Chapter 11 when evaluating the predistortion results.

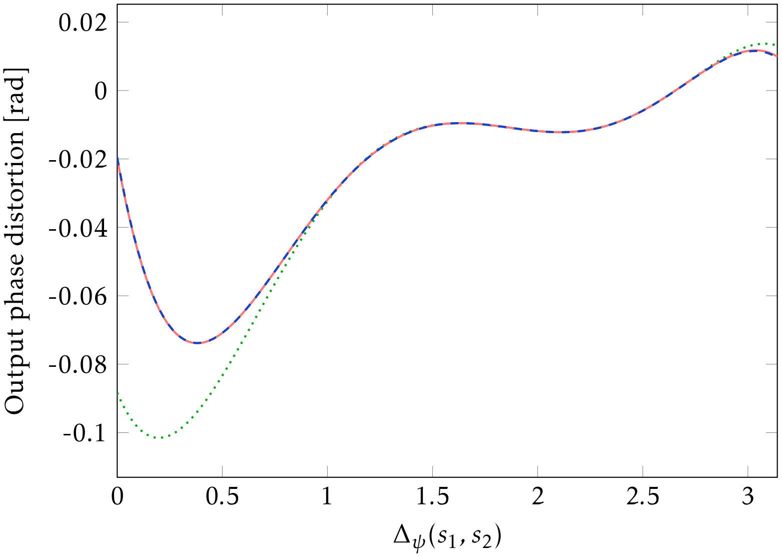


Figure 9.9: *Simulated output phase distortion of the models from the nonconvex method (dotted green) and the* LS *methods using* DR *(dashed blue) and gridding (pink) (the two model outputs are almost completely on top of each other). The lines describe the modeled phase difference as a function of the input signal amplitudes, that is, taking the different gain factors into account. The three methods evaluated estimate the phase shift almost equally for the middle range where most of the data points are (99.1% have* $0.8 \leq \Delta_\psi \leq 3.0$*), but the differences are visible at the edges.*

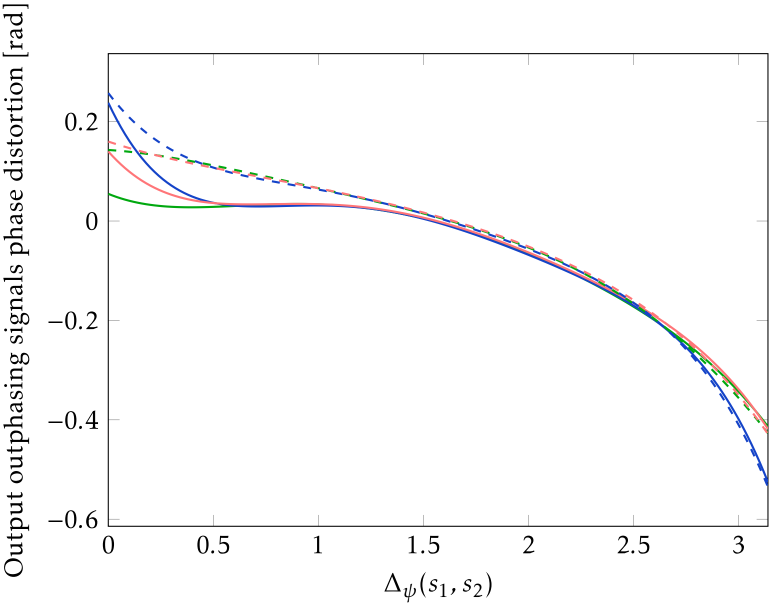


Figure 9.10: *Simulated outphasing output phase distortion of the models from the nonconvex method (green) and the* LS *methods using* DR *(blue) and gridding (pink). The lines describe the modeled phase in each branch as a function of the input signal amplitudes. Branch one is plotted in solid lines and branch two in dashed lines.*

9.5 Convex vs nonconvex formulations

The minimization of the cost function (9.12)-(9.14) is a nonconvex optimization problem in $2n+4$ dimensions with possible presence of local minima. Nonconvex optimization problems can either be solved by a local optimization method or a global one, see also Section 2.6. A local optimization method minimizes the cost function over points close to the current point, and guarantees convergence to a local minimum only. Global methods find the global minimum, at the expense of efficiency [Boyd and Vandenberghe, 2004]. Hence, even under ideal conditions (noise-free data, true PA described exactly by one model with the proposed structure), there is no guarantee that the nonconvex approach will produce an optimal model of the PA in finite time. The least-squares approach in (9.18)-(9.19) does exactly this and results in a closed-form expression for the parameter estimate. This is a major advantage since it removes the need for error-prone sub-optimality tests and possible time-consuming restarts of the search algorithm. Additionally, the computation time for the iterative, nonconvex, and potentially sub-optimal solution is significantly longer compared to the least-squares method.

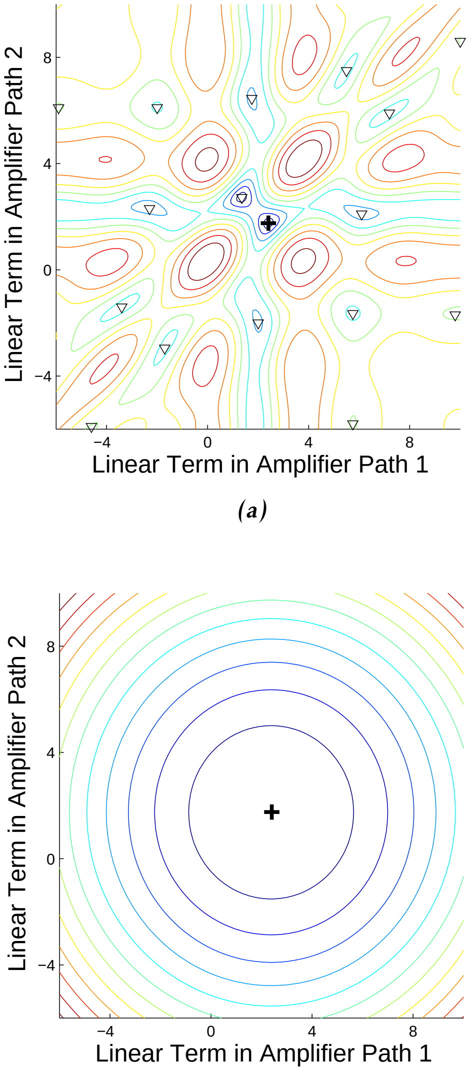
A two dimensional projection of the cost functions to be minimized, (9.13) in the nonlinear formulation and (9.19) in the LS reformulation, can be seen in Figure 9.11. All parameters but two have been fixed at the optimum, and the linear term in each amplifier branch ($\eta_{k,1}$ in (9.11)) has been varied. Clearly, there is a risk of finding a local minimum in the nonconvex formulation illustrated in (a) whereas there is only one (global) optimum in the least-squares formulation in (b).

The local minima in themselves might not be a problem if they are good enough to produce a well performing DPD, but there are no guarantees that this is the case. Typically, a number of different initial points need to be tested in order to get a reasonable performance.

9.6 Noise influence

Noise is always present in measurements, and the noise will effect the models. The algorithms presented in this chapter are sensitive to noise especially in two steps; the normalization $g_1 + g_2 = 1$ in (9.1) and the calculation of c_{DR} in (8.13). Both these calculations are based on very few measurements, one for the normalization (the largest amplitude) and two for the DR calculation (the smallest and the largest amplitudes), so noise at these instances might have a large influence on the estimation, and thus the performance of the predistorter.

The measurements used for the modeling and model validation in this chapter were recorded using the same measurement setup and power amplifier that will be used in Section 11.4. To avoid the influence of measurement noise, the same input was applied a number of times, K, and the output was measured, whereupon the average over the different realizations was calculated. In measurements used for the PA model estimation described here, $K = 10$. No automatic synchronization between input and measured output is done, so a manual syn-



(a)

(b)

Figure 9.11: *Two dimensional projections of the cost functions of (a) the original nonconvex optimization problem* (9.12)-(9.14) *and (b) the least-squares reformulation,* (9.18)-(9.19) *using the dynamic range for the estimation of g_1 and g_2. All but two parameters in each amplifier branch have been fixed at the optimal value, and the linear terms ($\eta_{k,1}$ in* (9.11)*) are varied. In (a), the visible local minima are marked with* ▽ *and the minimum obtained clearly depends on the initial point of the local optimization. In the least-squares formulation illustrated in (b), there is only one minimum (the global one) and convergence is guaranteed. The + marks the global minimum.*

chronization has to be performed. This also means that the sample times of the output differ between different measurement sets and that the synchronization between input and output is not the same for different data sets. When looking at the different data sets, the most dominant noise effect seems to stem from this time mismatch, which is evenly distributed around the mean value. The noise levels in general are very low.

9.7 Memory effects and dynamics

A more complex model structure has also been investigated by adding memory, that is to say that the output depends not only on the current input but also on the previous inputs, as in the model structure

$$p_{mem}(\alpha, \bar{\beta}_{n_m}(s)) = \sum_{m=0}^{n_m} \sum_{j=0}^{n} \alpha_{mj} \beta(s(t-m))^j, \tag{9.21}$$

with a memory depth n_m, where

$$\bar{\beta}_{n_m}(s) = \Big(\beta\big(s(t-m)\big)\Big)_{m=0}^{n_m}. \tag{9.22}$$

This approach did not lead to a better fit in the model validation, nor did it give any significant improvement in predistortion.

If dynamics are present in the PA, it is not unreasonable to assume that they would appear in the combiner, since the amplifier components in each branch can be assumed to contribute with little dynamics. This would mean that we have a parallel Hammerstein system with two parallel nonlinear, static branches (the amplifiers) followed by a dynamic system (the combiner). To investigate how such dynamics would effect the method described above, a dynamical system has been simulated at the output of a static model. The model was estimated using the LS method with DR. The dynamical system was a first order system with different values of the time constant in the range $[0.2T_s \;\; 5T_s]$, where T_s is the sample time. The same identification method was then applied to this data. In this case, the decomposition of the output using an estimate of g_1 and g_2 (obtained by dynamic range or gridding), is no longer a good approximation of the system, and the method will not perform in a satisfactory way. Thus, further investigation of how to include dynamics is needed.

10

Predistortion

Power amplifiers in communication devices are often nonlinear and/or dynamic, which causes interference in adjacent transmitting channels. To reduce this interference, linearization is needed. This is preferably done at the input, so that a prefilter inverts the nonlinearities/dynamics. This prefilter is called a *predistorter* (PD). Originally, these predistorters consisted of small analog circuits, but now they are often implemented in a *look-up table* (LUT) or a *digital signal processor* (DSP). Such an implementation is called a *digital predistorter* (DPD).

For the outphasing amplifiers evaluated in this thesis, the gain mismatch could be eliminated by adjusting the voltage supplies in the output stage, but this would require an extra adjustable voltage source on the chip, which is undesirable. Instead, the goal is to find a predistorter that uses only the phases of the two outphasing signals. By adjusting the outphasing signals, it is possible to achieve all amplitudes (within the dynamic range) and phases, and this idea will be explored in the construction of a predistorter.

In this chapter, a description of an ideal DPD will be presented and different methods to obtain it will be described. As a first step, the evaluation of the predistorters will be based on a model of the PA (described in Chapter 9), on simulated data only. In Chapter 11, the predistorters will be evaluated on real measurement data.

10.1 A DPD description

With the description of the power amplifier in (9.9)-(9.10), it is clear that an ideal PA would have $f_1 = f_2 = 0$ and $g_1 = g_2 = g_0 = 0.5$ and any deviations from these values will cause nonlinearities in the output signal and spectral distortion. In order to compensate for these effects, a DPD can be used to modify the input outphasing signals to the two amplifier branches, i.e., $s_1(t)$ and $s_2(t)$.

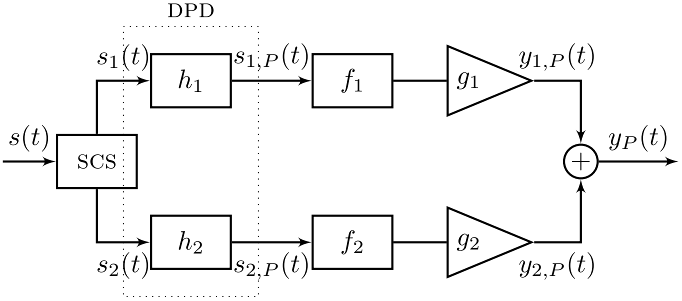


Figure 10.1: *A schematic picture of the amplifiers with predistorters. Note that the functions f_k and h_k, $k = 1, 2$, are not functions of the input to the block only, but are used to show the general functionality of the PA and the DPD with the separation of the two branches.*

Since the outputs of the Class D stages (the amplifiers in each branch) have constant envelopes, the DPD may only change the phase characteristics of the two input outphasing signals. With this in mind, a DPD that produces the predistorted signals

$$s_{k,P}(t) = e^{j\,h_k(\Delta_\psi)} s_k(t), \quad k = 1, 2, \tag{10.1}$$

to the two amplifier branches is proposed. Here, h_1 and h_2 are two real-valued functions that depend on the phase difference between the two signal paths. By modifying the signals in each branch using the DPD in (10.1), shown in Figure 10.1, the predistorted PA output $y_P(t)$ can be written

$$y_P = \underbrace{g_1 e^{j\,f_1(\Delta_\psi(s_{1,P},s_{2,P}))} s_{1,P}}_{\triangleq y_{1,P}} + \underbrace{g_2 e^{j\,f_2(\Delta_\psi(s_{1,P},s_{2,P}))} s_{2,P}}_{\triangleq y_{2,P}}. \tag{10.2}$$

The output is thus a sum of the two predistorted branches. In each branch $k = 1, 2$, the phase of the input is changed to counteract the effects of the nonequal gain factors and the PA nonlinearities. Each branch is predistorted separately and sent to the outphasing PA.

We will start by describing the effects of the predistorter on the output. The phase difference between the two paths after the predistorters is described by

$$\begin{aligned}\Delta_\psi(s_{1,P}, s_{2,P}) &= \arg(s_{1,P}) - \arg(s_{2,P})\\ &= [\arg(s_1) + h_1(\Delta_\psi)] - [\arg(s_2) + h_2(\Delta_\psi)]\\ &= \Delta_\psi + h_1(\Delta_\psi) - h_2(\Delta_\psi) \triangleq \tilde{h}(\Delta_\psi),\end{aligned} \tag{10.3}$$

and the phase difference between the two paths at the (predistorted) outputs by

$$\begin{aligned}
\Delta_\psi(y_{1,P}, y_{2,P}) &= \arg(y_{1,P}) - \arg(y_{2,P}) \\
&= \Big[\arg(s_{1,P}) + f_1(\Delta_\psi(s_{1,P}, s_{2,P}))\Big] - \Big[\arg(s_{2,P}) + f_2(\Delta_\psi(s_{1,P}, s_{2,P}))\Big] \\
&= \Big[\arg(s_1) + h_1(\Delta_\psi) + f_1(\tilde{h}(\Delta_\psi))\Big] - \Big[\arg(s_2) + h_2(\Delta_\psi) + f_2(\tilde{h}(\Delta_\psi))\Big] \\
&= \Delta_\psi + h_1(\Delta_\psi) - h_2(\Delta_\psi) + f_1(\tilde{h}(\Delta_\psi)) - f_2(\tilde{h}(\Delta_\psi)) \\
&= \tilde{h}(\Delta_\psi) + f_1(\tilde{h}(\Delta_\psi)) - f_2(\tilde{h}(\Delta_\psi)) \\
&\triangleq \tilde{f}(\tilde{h}(\Delta_\psi)).
\end{aligned} \tag{10.4}$$

These phase differences correspond to the amplitude of the signal, since it is known that $|s| = \cos(\Delta_\psi/2)$, cf. Figure 8.12. The absolute phase change in each branch is given by

$$\arg(y_{k,P}) = \arg(s_k) + h_k(\Delta_\psi) + f_k(\Delta_\psi(s_{1,P}, s_{2,P})) \tag{10.5}$$

for $k = 1, 2$. We now have a model structure describing how the phases of each outphasing signal, and thus the amplitude and phase of the output, depend on the characteristics g_1, g_2, f_1 and f_2 of the PA and the predistorter functions h_1 and h_2.

10.2 The ideal DPD

As mentioned above, the PA output signal $y(t)$ is a distorted version of the input signal. An ideal DPD should compensate for this distortion and result in a normalized output signal $y_P(t) = y_{1,P}(t) + y_{2,P}(t)$ that is equal to the input signal $s(t) = 0.5s_1(t) + 0.5s_2(t)$. In the ideal case when $g_1 = g_2 = g_0 = 0.5$, this is obtained when $y_1(t) = 0.5s_1(t)$ and $y_2(t) = 0.5s_2(t)$. However, this is not possible to achieve when $g_k \neq 0.5, k = 1, 2$. In this case, the ideal values for $y_{1,P}(t)$ and $y_{2,P}(t)$ are instead $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$, as described in (9.3). These signals define an alternative decomposition of $s(t)$ such that the gain mismatch is accounted for.

Assume now that an ideal DPD (10.1) is used together with the PA (9.9). In this case, the equalities

$$y_{1,P}(t) = \tilde{s}_1(t) \tag{10.6}$$

and

$$y_{2,P}(t) = \tilde{s}_2(t) \tag{10.7}$$

hold, which results in

$$y_P(t) = y_{1,P}(t) + y_{2,P}(t) = \tilde{s}_1(t) + \tilde{s}_2(t) = s(t).$$

That is, when the ideal DPD is applied to the PA, the original input will be retrieved. This assumes that the model perfectly describes the PA. Some more conclusions can be drawn about the ideal DPD by looking at the amplitudes and the phases of the input and the output. In order not to distort the amplitude at the output, the phase difference between $y_{1,P}(t)$ and $y_{2,P}(t)$ must be equal to the one between $\tilde{s_1}(t)$ and $\tilde{s_2}(t)$, i.e.,

$$\begin{aligned}\Delta_\psi(y_{1,P}, y_{2,P}) &= \Delta_\psi(\tilde{s_1}, \tilde{s_2}) = \arg(\tilde{s_1}) - \arg(\tilde{s_2}) = \\ &= \Big[\arg(s_1) + \xi_1(\Delta_\psi)\Big] - \Big[\arg(s_2) + \xi_2(\Delta_\psi)\Big] \\ &= \Delta_\psi + \xi_1(\Delta_\psi) - \xi_2(\Delta_\psi) \overset{\Delta}{=} \tilde{\xi}(\Delta_\psi). \end{aligned} \tag{10.8}$$

Hence, inserting (10.8) into (10.4) gives

$$\tilde{f}(\tilde{h}(\Delta_\psi)) = \tilde{\xi}(\Delta_\psi) \quad \Leftrightarrow \quad \tilde{h}(\Delta_\psi) = \tilde{f}^{-1}(\tilde{\xi}(\Delta_\psi)), \tag{10.9}$$

assuming that $\tilde{f}$ is invertible. Furthermore, for (10.6) and (10.7) to hold, that is, $y_{1,P} = \tilde{s_1}$ and $y_{2,P} = \tilde{s_2}$, we require that the phases of the two signals are equal,

$$\arg(y_{k,P}) = \arg(\tilde{s_k}), \quad k = 1, 2. \tag{10.10}$$

Now, we have a description of how the predistorter will affect the output as well as how the gain factors g_1 and g_2 change the desired outphasing output signals. The phase condition (10.10) combined with (10.3), (10.5) as well as (9.6) or (9.7), respectively (for each branch), gives

$$\arg(s_k) + h_k(\Delta_\psi) + f_k(\tilde{h}(\Delta_\psi)) = \arg(s_k) + \xi_k(\Delta_\psi),\ k = 1, 2.$$

That is, if we know the power amplifier functions f_k, the predistorter functions h_k is the only unknown in each branch and can be solved for. This results in

$$\begin{aligned} h_k(\Delta_\psi) &= -f_k(\tilde{h}(\Delta_\psi)) + \xi_k(\Delta_\psi) \\ &= -f_k(\tilde{f}^{-1}(\tilde{\xi}(\Delta_\psi))) + \xi_k(\Delta_\psi) \end{aligned} \tag{10.11}$$

for $k = 1, 2$. Here, (10.9) has been used in the last equality.

Hence, using the predistorters (10.11) in (10.1), the output $y(t)$ will be an amplified replica of the input signal $s(t)$, despite the gain mismatch and nonlinear behavior of the amplifiers. This is valid within the DR, which is where we have the opportunity to improve the behavior.

10.3 Nonconvex DPD estimator

A first approach to identify the predistorter is to notice that the goal is to minimize the difference between the normalized input and the normalized predistorted output. This can be written down in a straightforward way as solving the minimization criterion

$$\hat{\theta}_{\text{DPD}} = \underset{\theta_{\text{DPD}}}{\operatorname{argmin}} \sum_{t=1}^{N} \left| s(t) - \hat{y}_{\text{P}}(t, \theta_{\text{DPD}}) \right|^2, \tag{10.12}$$

$$\hat{y}_{\text{P}}(t, \theta_{\text{DPD}}) = \hat{g}_1 e^{j\,p(\hat{\eta}_1, \Delta_\psi(s_{1,P}, s_{2,P}))} s_{1,P}(t) + \hat{g}_2 e^{j\,p(\hat{\eta}_2, \Delta_\psi(s_{1,P}, s_{2,P}))} s_{2,P}(t), \tag{10.13}$$

where

$$s_{k,P}(t) = e^{j\,p(\eta_{k,\text{DPD}},\Delta_\psi(s_1,s_2))} s_k(t), \quad k = 1, 2, \tag{10.14}$$

and $\theta_{\text{DPD}} = [\eta_{1,\text{DPD}}^T \quad \eta_{2,\text{DPD}}^T]^T \in \mathbb{R}^{2n+2}$. The signal $\hat{y}_P(t)$ is the output from a PA model, using a predistorted input, as in Figure 10.1, where the amplifiers are replaced by the obtained models thereof. The DPD is thus identified based on a model of the forward system, according to Method B1, Procedure 5.4. The forward model was approximated by polynomials, $\hat{f}_k(\Delta_\psi) = p(\eta_k, \Delta_\psi)$, according to (9.11), and this is used in (10.12)-(10.13) to explicitly point out the dependence on the model parameters. When identifying the DPD model, the model structure was assumed to be the same as for the PA model, see (9.11), motivated by the Stone-Weierstrass theorem (Theorem 7.26 in Rudin [1976]), so that

$$\hat{h}_k(\Delta_\psi) = p(\eta_{k,\text{DPD}}, \Delta_\psi) = \sum_{i=0}^{n_h} \eta_{k,i,\text{DPD}} \Delta_\psi^i, \quad k = 1, 2, \tag{10.15}$$

where

$$\eta_{k,\text{DPD}} = \begin{pmatrix} \eta_{k,0,\text{DPD}} & \eta_{k,1,\text{DPD}} \cdots & \eta_{k,n,\text{DPD}} \end{pmatrix}^T.$$

The resulting estimated parameter vector $\hat{\theta}_{\text{DPD}}$ contains the DPD model parameters.

This formulation leads to a nonconvex optimization problem and is thus at a risk of obtaining a suboptimal solution if the optimization algorithm finds a local minimum. To restart the algorithm at different initial points is a possible way to reduce the risk of getting stuck in a local minimum instead of the global minimum, but this solution would not be useful in an online implementation, see also Section 9.5 for a discussion on convex and nonconvex optimization.

10.4 Analytical DPD estimator

The ideal DPD outlined in Section 10.2 requires knowledge of the PA model, and once the PA characteristics g_1, g_2, f_1 and f_2 are known (or estimated), the predistorter functions can be determined. The first step to construct a DPD is thus to obtain a model of the PA, as described in Chapter 9. This method is similar to Method A, Procedure 5.3, where a model of the system itself is used to analytically produce an inverse. Here, the goal is not to reconstruct the original inputs exactly, but the overall idea is still the same.

The parameter estimates $\hat{\eta}_k$ define function estimates (9.20)

$$\hat{f}_k(x) = p(\hat{\eta}_k, x), \quad k = 1, 2,$$

from which an estimate

$$\hat{\tilde{f}}(x) = x + \hat{f}_1(x) - \hat{f}_2(x) \tag{10.16}$$

of the function $\tilde{f}$ from (10.4) can be computed. Provided that this function can be inverted numerically, estimates $\hat{h}_k$ of the ideal phase correction functions can be computed as in (10.11), i.e.,

$$\hat{h}_k(\Delta_\psi) = -\hat{f}_k(\hat{\tilde{f}}^{-1}(\tilde{\xi}(\Delta_\psi))) + \xi_k(\Delta_\psi) \tag{10.17}$$

for $k = 1, 2$, where Δ_ψ is given by (9.2) and $\tilde{\xi}$, ξ_1 and ξ_2 by (10.8), (9.6) and (9.7), respectively.

Hence, the complete DPD estimator consists of the selection of gain factors g_1 and g_2, see Sections 9.1 and 9.3. Also, the two least-squares estimators given by (9.18), a numerical function inversion in order to obtain $\hat{\tilde{f}}^{-1}$ and the expressions for the phase correction functions in (10.17) make part of the complete DPD estimator. The DPD estimation can either be done at each point in time, or (as has been done here) by evaluating the function for the range of possible Δ_ψ and saving this nonparametric, piecewise constant function.

The DPD estimator will result in two functions $\hat{h}_1$ and $\hat{h}_2$ which take Δ_ψ as argument, and by using these as in (10.1), the predistorted input signals $s_{1,P}(t)$ and $s_{2,P}(t)$ can be calculated for arbitrary data. Measurement results for a validation data set, not used during the modeling, will be presented in Chapter 11.

The algorithm thus consists of two main parts, A – Estimation of PA model and B – Calculation of DPD functions. Part A consists of three subparts where the first, A.I, produces candidates for the gain factors g_1 and g_2 by either using the DR by gridding possible values. A.II produces LS estimates of the nonlinear functions $\hat{f}_1$ and $\hat{f}_2$ for each pair of g_1 and g_2 and in A.III, the best performing model is chosen among all the candidates. In Part B, the DPD functions $\hat{h}_1$ and $\hat{h}_2$ are calculated. The different steps are described in more detail in Algorithm 10.1.

10.5 Inverse least-squares DPD estimator

In the deduction of the predistorter described above, the ideal DPD was produced using analytical relationships between the input and the desired output, following the basic METHOD A. By instead choosing METHOD C, we want to estimate the inverse directly. This means that the system input $s(t)$ (or rather $s_1(t)$ and $s_2(t)$) will be considered as output to the identification, and $y(t)$ (or $y_1(t)$ and $y_2(t)$) as input, see also Procedure 5.6.

Since g_1 and g_2 can be found rather easily (through the dynamic range or gridding), these can still be assumed to be known, so the decomposition of $y(t)$ into $y_1(t)$ and $y_2(t)$ can be performed using (9.3). In each branch $k = 1, 2$ we thus have

$$\arg(s_k) = \arg(y_k) - h_k\left(\Delta_\psi(y_1, y_2)\right). \tag{10.18}$$

The left hand side is the input, which is known. The first term on the right hand side represents what we have measured, using the decomposition (9.3). The second term represents how the outphasing outputs should be modified to match the input, a *postdistorter*. The only unknown is thus the predistorter functions h_1 and h_2 in the two branches. By approximating these as polynomials,

$$\hat{h}_k \approx p(\zeta_k, \Delta_\psi(y_1, y_2)) = \sum_{i=0}^{n_h} \zeta_{k,i} \Delta_\psi^i(y_1, y_2), \quad k = 1, 2, \tag{10.19}$$

where

$$\zeta_k = \begin{pmatrix} \zeta_{k,0} & \zeta_{k,1} \dots & \zeta_{k,n} \end{pmatrix}^T.$$

Algorithm 10.1 LS modeling and analytical DPD method

```
Require: model order $n$, method for choice of $g_1$ and $g_2$, precision of PA model
    ($p_M$) and inverse ($p_I$), estimation data.

    {A – Estimation of PA model}
 1: Normalize the output $y(t) = \frac{y(t)}{max(|y(t)|)}$
 2: Calculate $\Delta_\psi$ $\forall t$ according to (9.2).

    {A.I – Estimation of gain factor candidates $g_1$ and $g_2$}
 3: if Use Dynamic Range to determine $g_1$ and $g_2$ then
 4:   Calculate $c_{DR}$ using (8.13), and $\Delta_g$ using (9.17).
 5:   Calculate possible choices of $g_1, g_2$ according to (9.15).
 6: else {$g_1$ and $g_2$ over a range of values}
 7:   Grid $g_1 \in [g_{\min}, g_{\max}]$ with precision $p_M$ and let $g_2 = 1 - g_1$.
 8: end if

    {A.II – Estimation of nonlinearity function candidates $\hat{f}_1$ and $\hat{f}_2$}
 9: for all pairs of $g_1, g_2$ do
10:   Create $\tilde{s}_k = g_k e^{j\ \arg(\tilde{s}_k)}$ and $y_k = g_k e^{j\ \arg(y_k)}, k = 1, 2$ using (9.4) to (9.7).
11:   Find $\eta_k$ using (9.18) and calculate $\hat{f}_k, k = 1, 2$ using (9.20).
12:   Simulate the output $\hat{y}_{g_1,g_2}(t) = g_1 e^{j\,\hat{f}_1(\Delta_\psi)} s_1(t) + g_2 e^{j\,\hat{f}_2(\Delta_\psi)} s_2(t)$.
13:   Calculate error $V_g(g_1, g_2) = \sum_t |y(t) - \hat{y}_{g_1,g_2}(t)|^2$.
14: end for

    {A.III – Choose best forward model, $\hat{g}_1, \hat{g}_2, \hat{f}_1$ and $\hat{f}_2$}
15: Select $\hat{g}_1 = \arg\min_{g_1} V_g(g_1, 1 - g_1)$, $\hat{g}_2 = 1 - \hat{g}_1$ and the corresponding $\hat{f}_1, \hat{f}_2$.

    {B – Calculation of DPD functions $\hat{h}_1$ and $\hat{h}_2$}
    {Create a look-up table (LUT) for different values of $\Delta_\psi$ by creating an
    intermediate signal $s$}
16: Grid $\Delta_\psi \in [0, \pi]$ with precision $p_I$.
17: for each value of $\Delta_\psi$ do
18:   Create $s = \cos(\Delta_\psi/2)$ according to (8.10) assuming $\alpha = 0$ and $r_{\max} = 1$
      ($\varphi = \Delta_\psi/2$).
19:   Create $s_1$ and $s_2$ according to (8.11) and $\tilde{s}_1$ and $\tilde{s}_2$ using (9.3) to (9.5).
20:   Find $\tilde{\xi}$ using (10.8), (9.6) and (9.7).
21:   Calculate $\hat{\tilde{f}}(\tilde{\xi})$ using (10.16).
22: end for
23: Invert $\hat{\tilde{f}}(\tilde{\xi})$ numerically to get $\hat{\tilde{f}}^{-1}$. This can e.g. be done by calculating $\tilde{f}(\tilde{\xi})$
    for a number of values of $\tilde{\xi} \in [0, \pi]$, grid $\hat{\tilde{f}}(\tilde{\xi})$ and match with the $\tilde{\xi}$ that gives
    the closest value.
24: for each value of $\Delta_\psi$ in line 16 do
25:   Find estimate $\hat{h}_k(\Delta_\psi)$ according to (10.17).
26: end for
```

as was done for the PA model, the parameters corresponding to the h_k-functions can be found.

The estimates $\hat{\zeta}_{k,i}$ of the model parameters have been computed by minimizing a quadratic cost function, i.e.,

$$\hat{\zeta}_k = \arg\min_{\zeta_k} V_k^h(\zeta_k), \quad k = 1, 2, \tag{10.20}$$

where

$$V_k^h(\zeta_k) = \sum_{t=1}^{N} \Big(\arg\left(y_k(t)\right) - \arg\left(s_k(t)\right) - p\left(\zeta_k, \Delta_\psi(y_1(t), y_2(t))\right)\Big)^2. \tag{10.21}$$

The parameter estimates $\hat{\zeta}_k$ define inverse function estimates

$$\hat{h}_k(x) = p(\hat{\zeta}_k, x), \quad k = 1, 2, \tag{10.22}$$

that can be used as a DPD. As discussed in Chapter 3, this method assumes commutativity of the two systems (system and inverse), so that the inverse which was estimated at the output of the power amplifier, a *postdistorter*, can also be used at the input as a *predistorter*. The method is summarized in Algorithm 10.2.

Algorithm 10.2 Inverse LS DPD method

```
Require: model order $n_h$, method for choice of $g_1$ and $g_2$, precision of gain factors ($p_M$), estimation data.

Normalize the output $y(t) = \frac{y(t)}{max(|y(t)|)}$

{I – Estimation of gain factor candidates $g_1$ and $g_2$}
if Use Dynamic Range to determine $g_1$ and $g_2$ then
    Calculate $c_{DR}$ using (8.13), and $\Delta_g$ using (9.17).
    Calculate possible choices of $g_1, g_2$ according to (9.15).
else {$g_1$ and $g_2$ over a range of values}
    Grid $g_1 \in [g_{\min}, g_{\max}]$ with precision $p_M$ and let $g_2 = 1 - g_1$.
end if

{II – Estimation of nonlinearity function candidates $\hat{h}_1$ and $\hat{h}_2$}
for all pairs of $g_1, g_2$ do
    Create $y_k = g_k e^{j \arg(y_k)}$ using (9.4) to (9.7) and $s_k$ using (8.11).
    Calculate $\Delta_\psi(y_1, y_2) \ \forall t$ according to (9.2).
    Find $\hat{\zeta}_k$ using (10.20) and calculate $\hat{h}_k, k = 1, 2$ using (10.22).
    Simulate the input $\hat{s}_{g_1,g_2}(t) = e^{-j \hat{h}_1(\Delta_\psi(y_1,y_2))} y_1(t) + e^{-j \hat{h}_2(\Delta_\psi(y_1,y_2))} y_2(t)$
    Calculate error $V_g(g_1, g_2) = \sum_t |s(t) - \hat{s}_{g_1,g_2}(t)|^2$
end for

{III – Choose best inverse model, $\hat{h}_1$ and $\hat{h}_2$}
Select $\hat{g}_1 = \arg\min_{g_1} V_g(g_1, 1 - g_1)$, $\hat{g}_2 = 1 - \hat{g}_1$ and the corresponding $\hat{h}_1$ and $\hat{h}_2$.
```

Table 10.1: *DPD model validation*

Method	$\|s - \hat{y}_p\|_2^2$
Analytical	0.0532
LS	1.008
Gain factors only	65.5

10.6 Simulated evaluation of analytical and LS predistorter

The goal here is to evaluate the performance of the predistorter methods in simulations, and determine how well the different methods achieve an inversion. One way is to look at the AM-AM modulation to assess how much the amplitude of the predistorted output is distorted. For an outphasing PA, this is connected to the phase difference $\Delta_\psi(y_{1,P}, y_{2,P})$ of the outphasing outputs $y_{1,P}$ and $y_{2,P}$.

The predistorter methods in Sections 10.4 and 10.5 are evaluated using a model of the amplifier as "the truth". The model is presented in Chapter 9, where the gain factors were estimated using the DR and the nonlinearities using the LS approach, see Section 9.3 and the model validation in Section 9.4 and Figure 9.5. The same validation data have been used in order to evaluate the different predistorter methods. Evaluation on a real PA will be presented in Chapter 11.

Test 1 – Inversion Evaluation

We will start by looking at the AM-AM modulation to determine how much the amplitude of the predistorted output is changed. The deviation from the ideal phase difference at the output (i.e., the output amplitude) with and without predistortion is presented in Figure 10.2. Both the analytical method and the LS method clearly reduce the phase shift introduced by the PA. Figure 10.3 shows the estimated deviation from the ideal phase for each signal branch with and without predistortion, with rather similar performance for the two DPD methods.

The values of the cost function (10.12) are presented in Table 10.1 for the two methods. The result using only the estimation of the gain factors and the alternative decomposition (using the knowledge of the nonequal gain factors) is also presented. It is clear that incorporating the nonlinearities improves the performance. For cases when the gain factors differ more from the ideal $g_1 = g_2 = 0.5$ than in this case ($g_1 = 0.4986$ and $g_2 = 0.5014$), the alternative decomposition (9.3) will have a larger improvement on the modeling than in this case, when the difference is small.

For the LS method, the fit is almost perfect in the middle range, which is to be expected since a polynomial is used (see discussion in Section 9.4. Also the number of measurements is unevenly spread out over Δ_ψ with most data in the middle, only 0.9% of the estimation data have $\Delta_\psi < 0.8$ or $\Delta_\psi > 3$. For the

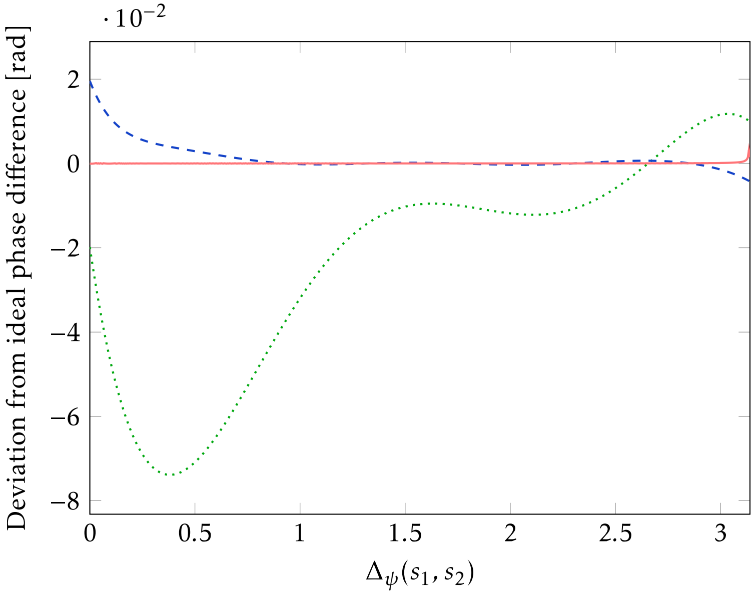


Figure 10.2: *Simulated predistorter evaluation for a model with polynomial degree $n = 5$ using the* WCDMA *input signal (see Chapter 11). The signals are generated using the* DPD *functions and the* PA *model. For an ideal* PA*, there is no amplitude distortion, that is, the phase difference of the outphasing signals is the same at the output and the input. The deviation from this ideal phase difference for the output signal (modeled, not predistorted) $\hat{y}$ is shown in dotted green and the predistorted output signals $\hat{y}_P$ in pink and blue. The pink line shows the result using the analytical inversion as described in Sections 10.2 and 10.4 and the dashed blue line shows the result of the* LS *approach in Section 10.5, with predistorter degree $n_h = 5$ in* (10.19)*. The two methods both perform very well in a large interval.*

analytical solution, one can see an inversion error close to $\Delta_\psi = \pi$. This is a consequence of the nonequal gain factors, $\Delta_\psi = \pi$ should represent a complete opposition of the two outphasing signals such that the output amplitude is zero. If $g_1 \neq g_2$ however, this is not possible and no phase combination of the two outphasing signals will lead to a zero-amplitude output. A power amplifier with a large dynamic range (DR, difference between the gain factors g_1 and g_2) will have a very small distortion close to $\Delta_\psi = \pi$, whereas a PA with a small DR will show this distortion in a larger region. The errors of the two methods when compared to validation data are shown in Figure 10.4. Also in this plot, it can be seen that both methods reduce the power amplifier distortion, and that the analytical inversion performs slightly better.

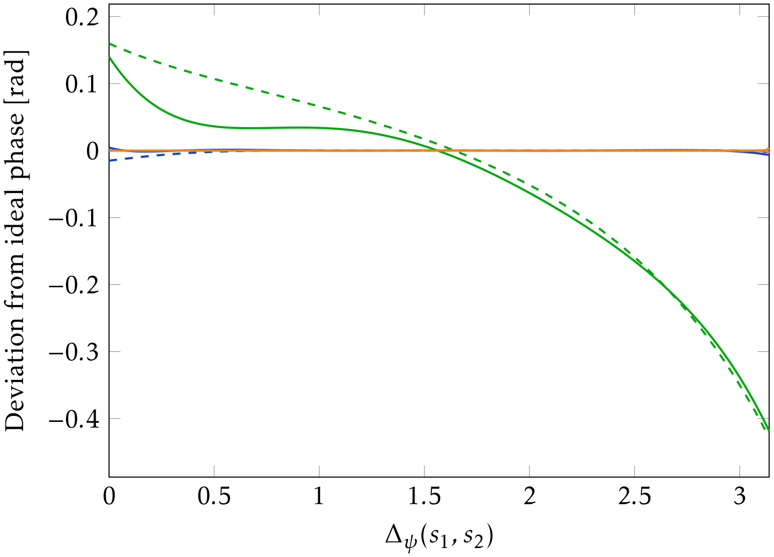


Figure 10.3: *Simulated predistorter evaluation for a model with polynomial degree $n = 5$ using the WCDMA input signal. The signals are generated using the DPD functions and the PA model. The deviation from the ideal phase for the output outphasing signals (modeled, not predistorted) $\hat{y}_1$ and $\hat{y}_2$ are shown in green and the predistorted output signals $\hat{y}_{1,P}$ and $\hat{y}_{2,P}$ in pink and blue. The pink lines show the results using the analytical inversion as described in Sections 10.2 and 10.4 and the blue lines show the result of the LS approach in Section 10.5. Branch one is plotted in solid lines and branch two in dashed lines. It should be noted that the analytical method uses a look-up table for with $2 * 3142$ elements and the LS method uses polynomials with $2 * 6$ coefficients ($n_h = 5$ in (10.19)). So for a fair comparison between the two methods we would need to look at more even number of parameters, but the goal here is to show that both methods perform well.*

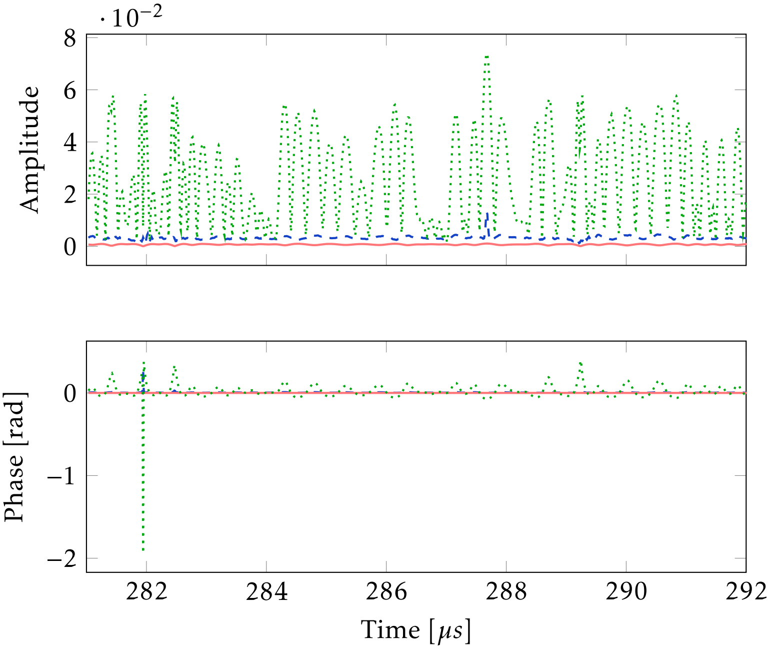
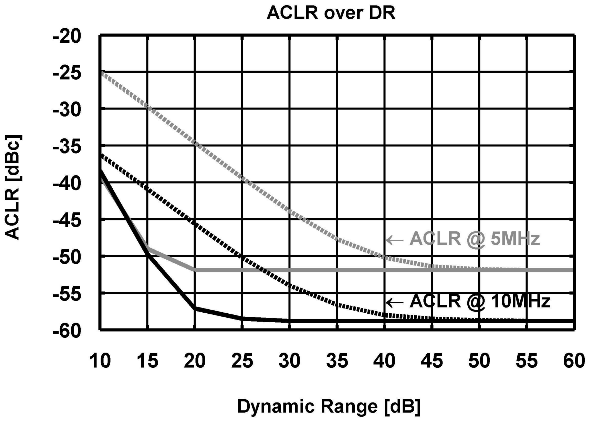


Figure 10.4: *The upper plot shows the amplitude error,* $|s - \hat{y}_p|$*, and the lower plot shows the phase error,* $\arg(s) - \arg(\hat{y}_p)$*, for the two DPD methods. The analytical method is in pink and the LS in blue. As a comparison, the errors for the original, unpredistorted signal* $y(t)$ *are also plotted in green.*



© 2013 IEEE

Figure 10.5: Simulated ACLR at 5 MHz and 10 MHz offset with DPD (solid line) and without (dashed line) for the WCDMA signal.

Test 2 – Impact of ACLR on predistorter performance

As previously explained, the result of a limited dynamic range is that all amplitude and phase errors occurring outside the DR cannot be corrected. The signal clipping in an outphasing PA occurs at small amplitudes, while in a conventional linear PA, the peak amplitudes are clipped. Thus, the DR in an outphasing PA limits the spectral performance when amplifying modulated signals. To investigate the performance limits of the predistorter, simulations have been done using two amplifiers with a given DR (no phase distortion), with and without DPD. In Figure 10.5, the ACLR over DR at 5 MHz and 10 MHz for the WCDMA signal are plotted with and without DPD. Here, the phase error between the outphasing signals is assumed to be zero. For a PA with a DR of 25 dB the differences in ACLR between the nonpredistorted and predistorted outputs are 8-13 dB. When the DR is 25 dB the optimal theoretical ACLR is achieved after DPD. For a PA with 45 dB of DR, the difference between when a DPD is used or not is negligible.

Summary

In this simulated evaluation, both DPD methods achieve an improvement, compared to the original power amplifier output. The analytical inversion leads to slightly better results at the cost of a higher computational complexity. The look-up table for the analytical DPD has $2 * 3142$ elements (with precision $p_I = 0.001$ in Algorithm 10.1), and the polynomials contain $2 * 6$ coefficients ($n_h = 5$ in (10.19)). Using a higher polynomial degree could lead to improved results for the LS method, and a smaller LUT could lead to a degradation of the analytical method. As implementation issues are out of scope for this thesis, the methods are not optimized for implementation and therefore these considerations have not been further pursued.

10.7 Recursive least-squares and least mean squares

Here, a few aspects of a possible future implementation of the DPD methods are presented. In addition to the guaranteed convergence, least-squares formulations also have the advantage that there are many efficient numerical methods for solving this type of problems. They can be solved recursively by, for example, the *recursive least-squares* (RLS) method [Björck, 1996] making them suitable for an online implementation. An even less complex parameter estimation algorithm is the *least mean square* (LMS) method, which can make use of the linear regression structure of the optimization problem, developed here in (9.11) and (9.19). LMS has been used for RF PA linearization in Montoro et al. [2007] and implemented in *field programmable gate array* (FPGA) technology, as shown in Gilabert et al. [2009].

With a recursive implementation of the algorithm, it is even more important that the algorithm can be proved to converge to good values, as no monitoring of the performance should be necessary in order for the method to be useful in practice. This also means that a nonconvex solution as in (9.12)-(9.13) is not suitable for online implementation since it cannot guarantee convergence to good enough minima. In an offline application, the possibility to restart the optimization could be added but, together with the lack of a bound on the number of iterations, this does not seem like a good solution for an online version. Using well explored methods like RLS or LMS would result in a low-complexity implementation, and though it is hard to judge the exact complexity of the iterative implementation that would be needed for the online version of nonconvex solution, it is clear that it would be very hard to find a simpler one than for the low-complexity LMS version of the convex method.

Since circuitry will behave differently depending on the settings under which it operates, it is important to be robust to such conditions. This is covered in the concept of *process, voltage and temperature variations* (PVT variations). One way to handle the PVT variations and changes in the setting, such as aging, would be to use a method with a forgetting factor, reducing the influence of older measurements [Ljung, 1999]. The RLS and LMS solutions assume the changes in the operating conditions to be slow.

11

Predistortion measurement results

The models presented in Chapter 9 and the predistorters in Chapter 10 are based on measured data from a power amplifier. In Chapter 10, the methods' ability to invert the nonlinearities was investigated, using a forward model as a "true" system. In this chapter, the methods will be evaluated in real measurements. The predistorters are applied to a new data set, *validation data*, that is not the same as the signal used for estimation. To start off, a short introduction to the signal types used and the measurement setup will be presented.

11.1 Signals used for evaluation

The predistortion methods have been evaluated for the different signal types EDGE, WCDMA and LTE. Mobile communication technologies are often divided into generations, and the new devices of today are the fourth or fifth generation, 4G or 5G. The first generation, 1G, was the first analog mobile radio systems of the 1980s. 2G was the first digital mobile systems and 3G the first mobile systems handling broadband data.

Enhanced data rates for GSM evolution (EDGE) is a mobile phone technology with higher bit rates than *general packet radio service* (GPRS) [Ahlin et al., 2006], and has been called 2.75G since it did not quite reach the 3G standards. The carrier frequency used is 2 GHz, and the bandwidth is 200 kHz. *Wideband code division multiple access* (WCDMA) is a third generation (3G) mobile phone technology, and is one of the 3G mobile communications standards [Frenzel, 2003]. The carrier frequency used is 2 GHz, and the bandwidth is 5 MHz. The bandwidth of the *long term evolution* (LTE) signal is variable, and can be adjusted between 1 and 20 MHz. It is sometimes called 4G or 3.9G since it does not completely satisfy the 4G requirements [Dahlman et al., 2011].

The WCDMA and LTE have large peak-to-minimum power ratio, i.e., the PA

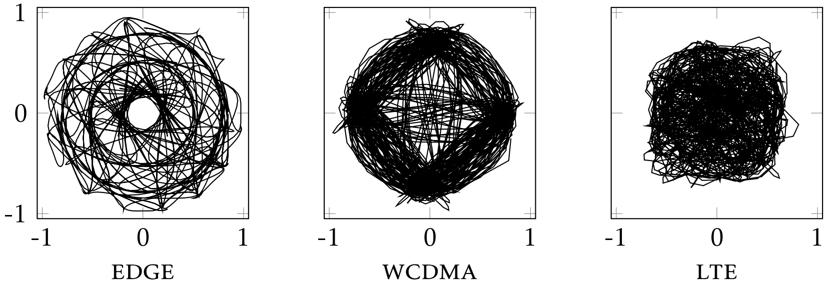


Figure 11.1: *IQ plots (imaginary part, Q, vs real part, I) of signal realizations of the EDGE, WCDMA and LTE standards in the complex plane. The sampling frequency in the modeling data sets is four times higher than that shown here.*

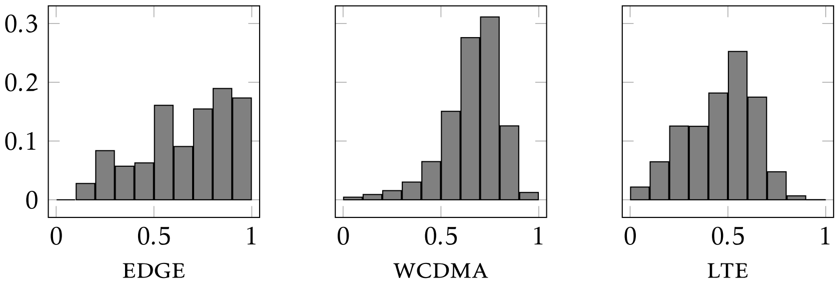


Figure 11.2: *Histograms of the distribution of the input amplitude of signal realizations of the EDGE, WCDMA and LTE standards. This difference in input distribution affects the peak-to-average power ratio, and it also implicitly determines the weighting of the fit of the polynomials, see also the discussion on polynomial fitting on page 143.*

output signals include the minimum and maximum amplitudes (the full dynamic range). For these signals, the DR of the PA will effect the output signal, by clipping the smallest amplitudes. For EDGE, the signal amplitude is never close enough to zero to be effected by the PA DR, and no clipping will occur. Realizations of each signal type (EDGE, WCDMA and LTE) are shown in Figure 11.1 as IQ-plots. Histograms of the distribution of the input amplitude are shown in Figure 11.2. The distribution also implicitly determines the weighting of the fit of the polynomials, see also the discussion on polynomial fitting on page 143. One characteristic of a signal is the *peak-to-average power ratio* (PAPR). A signal with a high PAPR sets high standards on the linearity of the PA, since a large range of input signal amplitudes has to be amplified.

The signals used are created as random signals with predefined characteristics.

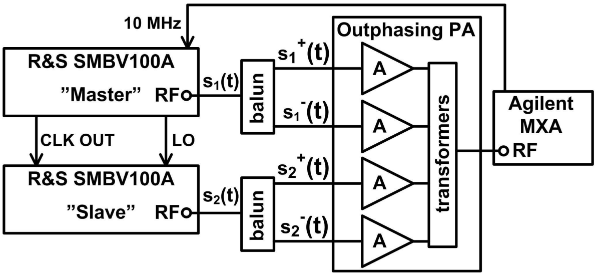


Figure 11.3: *Measurement setup for IQ-data with two Master-Slave-configured SMBV signal generators [Rohde & Schwarz].*

11.2 Measurement setup

The measurements that will be discussed in Section 11.3 have been performed using an SMU200A signal generator with two phase-coherent RF outputs and an arbitrary waveform generator where the input signals ($s_1(t)$ and $s_2(t)$) and the predistorted input signals ($\tilde{s}_1(t)$ and $\tilde{s}_2(t)$) were stored. For the measurements that will be discussed in Section 11.4, two R&S SMBV100A signal generators with phase-coherent RF outputs and arbitrary waveform generators with maximum IQ sample rate of 150 MHz have been used. Figure 11.3 shows the measurement setup.

The outphasing power amplifiers used in the measurements have been developed by Jonas Fritzin et al. and are briefly described in Appendix A and in more detail in Fritzin [2011].

Sampling

The sampling rate in Section 11.4 was 92.16 MHz in the measurements, six times the original sampling frequency of the signal. The impact of baseband filtering and limited bandwidth is investigated in Gerhard and Knöchel [2005a,b], where it was concluded that to obtain an optimal signal/distortion ratio over the entire bandwidth, a compromise between the sampling frequency and the filter characteristics has to be made. Here, we have evaluated the required bandwidth/sampling rate based on measurements with two signal generators and one combiner, no PA was used. Increasing the sampling frequency from the original 15.36 MHz to 30.72 MHz and 61.44 MHz, the ACLR is improved, see Table 11.1.

Thus, for the specific tests performed here, the ACLR at 5 and 10 MHz can be improved by 6-9 dB and 4-8 dB, respectively, when increasing the sampling rate up to four times the original sampling rate of 15.36 MHz. Further increasing the sampling frequency, up to 92.16 MHz, shows no significant change.

Table 11.1: *Measured spectral performance at 1.95 GHz for WCDMA and LTE uplink signals for different sampling frequencies.*

	Measured parameter	15.36 MHz	30.72 MHz	61.44 MHz
WCDMA	ACLR @ 5 MHz [dBc]	-44	-50	-52
	ACLR @ 10 MHz [dBc]	-48	-52	-56
LTE	ACLR @ 5 MHz [dBc]	-34	-43	-46

11.3 Evaluation of nonconvex method

In this section, the nonconvex approach presented in Sections 9.2 and 10.3 has been evaluated. The PA model has been obtained by minimizing the nonconvex cost function in (9.13) and the corresponding DPD by minimizing (10.12). The method involves solving two nonconvex optimization problems, and corresponds to METHOD B1 in Section 5.1. This method has been evaluated on the PA described in Appendix A.1 and Fritzin [2011].

The predistortion methods were evaluated on a physical chip. The measurement setup was optimized and the branch amplifiers were tuned to achieve the best performance possible. The phase offset between $s_1(t)$ and $s_2(t)$ in the baseband was adjusted to minimize phase mismatch (ideally 180 ° between the two RF inputs for nonmodulated $s_1(t)$ and $-s_2(t)$ in Figure A.2, i.e. maximum output power for a continuous signal). Since this is not a reasonable assumption in a real-life application, an additional phase error of 3 ° was added in one of the branches.

Measurements of input $s(t)$ and output $y(t)$ of length N_{id} were collected K times, and an average was taken to avoid the influence of measurement noise. This data was used to model the power amplifier. Based on this PA model, a predistorter model was produced. Polynomials with order n have been used as parameterized versions of the PA nonlinearities and of order n_h for the predistorter functions. The predistorted input signals, $s_{1,P}$ and $s_{2,P}$, were then computed (in MATLAB) for a validation input signal of length N_{val}. The predistorted outphasing input signals were sent to the PA, resulting in a predistorted output. The additional phase error was still applied during the predistorter validation.

For the computation of the model parameters, a large number of algorithms are available for solving a nonlinear optimization problem. Here, the MATLAB routine `fminsearch`, based on the Nelder-Mead simplex method, was used. The estimation and validation data sets contain N_{id} and N_{val} samples, respectively. The input and output sampling frequencies are denoted f_{s} and $f_{\text{s,out}}$, respectively. To minimize the influence of measurement noise, the signals were measured K times, and a mean was calculated. The data collection parameters are shown in Table 11.2. Since the WCDMA is a more wide-band signal than the EDGE signal, the number of samples N_{id} and N_{val} were chosen to be larger.

Table 11.2: *Data collection, nonconvex method*

	N_{id}	N_{val}	f_{s}	$f_{\text{s,out}}$	K
EDGE	40 001	80 001	8.67 MHz	34.68 MHz	150
WCDMA	153 600	153 600	61.44 MHz	61.44 MHz	200

Table 11.3: *Measured spectral performance of the EDGE signal*
(a) With no phase error and no DPD.
(b) For a 3 ° phase error and no DPD.
(c) When DPD is applied to (b).

Freq.	Freq. offset	Spec.	Meas. (a)	Meas. (b)	Meas. (c)
2 GHz	400 kHz	-54 dB	-54.4 dB	-53.5 dB	-65.9 dB
	600 kHz	-60 dB	-60.3 dB	-59.9 dB	-68.2 dB

11.3.1 Measured performance of EDGE signal

EDGE is a rather narrow-band signal with a *peak-to-average power ratio* (PAPR) of 3.0 dB. The spectrum of the estimation input data set is shown in Figure 11.4(d). The output of a perfectly matched PA in Figure 11.4(a) fulfills the requirements, but without any margins to the spectral mask. The spectral mask is a nonlinearity measurement that describes the amount of power that is allowed to be spread to the neighboring channels. The requirements for an EDGE signal are summarized in Table 8.1 and illustrated in Figure 11.4. As the phase error cannot be assumed to be 0 ° in a transceiver, a phase error of 3 ° was added and led to a violated spectral mask as in Figure 11.4(b).

When predistortion was applied to a validation data set, not used for estimation, the linearity improves, as seen in Figure 11.4(c). The PA model was of order $n = 5$ and the predistorter of order $n_h = 5$. The measured power at 400 and 600 kHz offsets were -65.9 and -68.2 dB, with margins of 11.9 and 8.2 dB, respectively. The average power at 2 GHz was +7 dBm with 22 % PAE and *root mean square* (RMS) EVM of 2 %. The measured performance of the amplifier for an EDGE signal is summarized in Table 11.3.

11.3.2 Measured performance of WCDMA signal

The PAPR of the WCDMA signal was 3.2 dB and the spectrum of the estimation data set is shown in Figure 11.5(d). Figure 11.5(a) shows the measured WCDMA spectrum at 2 GHz, with minimized phase mismatch and no predistortion. When the same phase error of 3 ° as for the EDGE signal was added to simulate reasonable phase settings, a distorted spectrum as in Figure 11.5(b) was measured. The ACLR is an integrated measure that describes the power spread to adjacent chan-

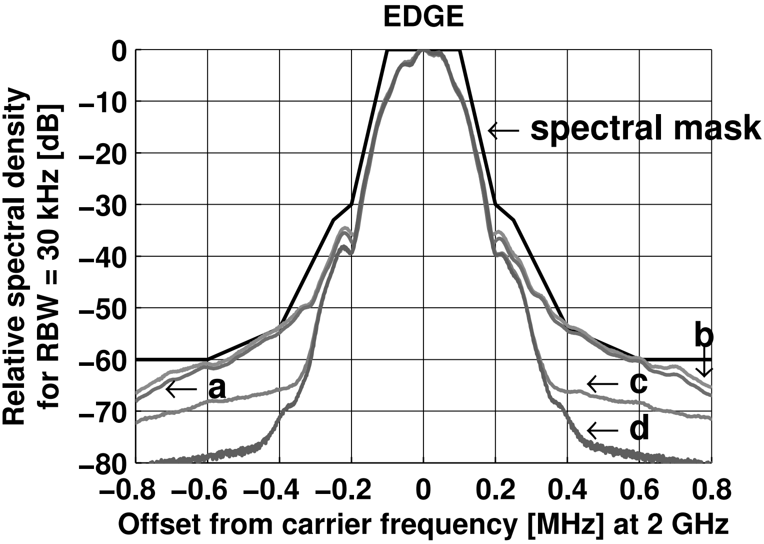


Figure 11.4: *Measured EDGE spectrum at 2 GHz.*
(a) Output spectrum without phase error between $s_1(t)$ and $s_2(t)$.
(b) Output spectrum with 3 ° phase error between $s_1(t)$ and $s_2(t)$.
(c) Output spectrum when DPD is applied to (b).
(d) Spectrum of the estimation signal. The spectrum of the validation signal was similar.

Table 11.4: *Measured spectral performance of the WCDMA signal*
(a) With no phase error and no DPD.
(b) For a 3 ° phase error and no DPD.
(c) When DPD is applied to (b).

Freq.	ACLR	Spec.	Meas. (a)	Meas. (b)	Meas. (c)
1 GHz	5 MHz	-33 dBc	-40.6 dBc	-39.4 dBc	-53.6 dBc
	10 MHz	-43 dBc	-59.8 dBc	-56.2 dBc	-60.3 dBc
2 GHz	5 MHz	-33 dBc	-43.4 dBc	-38.0 dBc	-50.2 dBc
	10 MHz	-43 dBc	-53.9 dBc	-50.9 dBc	-52.2 dBc

nels. At 1 GHz and 2 GHz, the power amplifier fulfills the requirements, also with the additional phase error, as seen in Table 11.4.

The phase predistortion method, with $n = 5$ and $n_h = 4$, for a validation signal, improves the measured ACLR. A spectrum is shown in Figure 11.5(c). The channel power at 2 GHz was +6.3 dBm with PAE of 22 % and RMS composite EVM of 1.4 % (0.6 % after DPD). The measured performance of the amplifier for a WCDMA signal is summarized in Table 11.4.

11.3.3 Summary

The nonconvex predistortion method clearly improves the PA performance for both EDGE and WCDMA signals, even when an extra phase error is added. The measured spectral performance at 400 kHz offset and the ACLR at 5 MHz is comparable to state-of-the-art EDGE [Mehta et al., 2010] and WCDMA [Huang et al., 2010] transmitters.

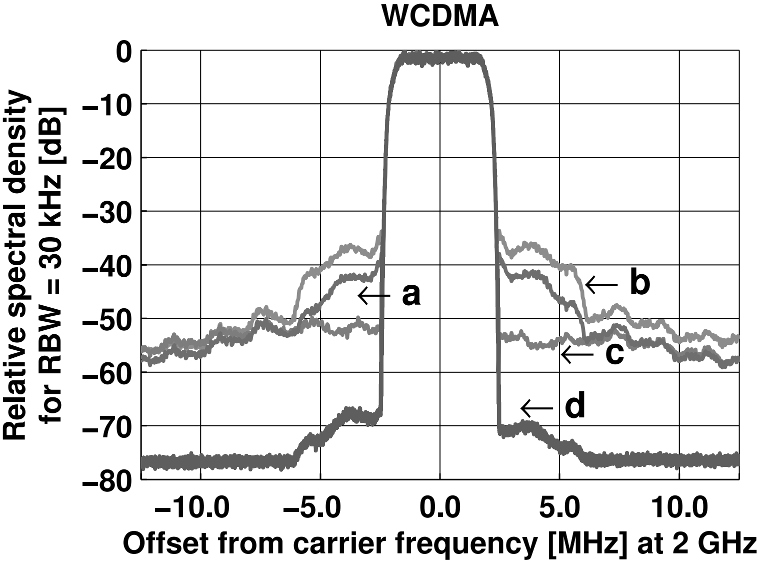


Figure 11.5: *Measured* WCDMA *spectrum at 2 GHz.*
(a) Output spectrum without phase error between $s_1(t)$ and $s_2(t)$.
(b) Output spectrum with 3 ° phase error between $s_1(t)$ and $s_2(t)$.
(c) Output spectrum when DPD *is applied to (b).*
(d) Spectrum of the estimation signal. The spectrum of the validation signal was similar.

Table 11.5: *Data collection, least-squares and analytical method*

	N_{id}	N_{val}	f_{s}	$f_{\text{s,out}}$	K
WCDMA	100 000	100 000	92.1 MHz	92.1 MHz	10
LTE	100 000	100 000	92.1 MHz	92.1 MHz	10

11.4 Evaluation of least-squares PA and analytical inversion method

In this section, the least-squares modeling of the PA, using the DR to estimate g_1 and g_2, has been applied. An analytical inversion has been used to construct the predistorter functions, as in METHOD A in Section 5.1. The PA modeling is described in Section 9.3, the DPD in Section 10.4 and the method is summarized in Algorithm 10.1, page 155. This method has been evaluated on the PA described in Appendix A.2 and Fritzin et al. [2011c].

The measurement setup was optimized and the branch amplifiers were tuned to achieve the best performance possible. For the measurements without predistortion, the phase offset between $s_1(t)$ and $s_2(t)$ in the baseband was adjusted to minimize phase mismatch (ideally $0\,^\circ$ between nonmodulated $s_1(t)$ and $s_2(t)$, that is, maximum output power for a continuous signal). Moreover, the IQ-delay between the signal generators was adjusted for optimal performance [Rohde & Schwarz].

Measurements of input $s(t)$ and output $y(t)$ were collected K times, and an average was taken to avoid the influence of measurement noise. This averaged data set was used to model the PA, and based on the PA model, a predistorter model was produced. Polynomials with order n have been used as parameterized versions of the PA nonlinearities and based on this model, an approximation of the ideal predistorter has been constructed. The predistorted input signals, $s_{1,P}$ and $s_{2,P}$, were then computed (in MATLAB) for a validation input signal. The predistorted outphasing input signals were sent to the PA, resulting in a predistorted output.

The estimation and validation data sets contain N_{id} and N_{val} samples, respectively. The input and output sampling frequencies are denoted f_{s} and $f_{\text{s,out}}$, respectively. The data collection parameters are shown in Table 11.5. In all following experiments, the DPD estimates $\hat{h}_k, k = 1, 2$, have been calculated for 3142 uniformly distributed points ($p_I = 0.001$ in Algorithm 10.1). This LUT has been used in the construction of the predistorted outphasing input signals. For each input phase difference Δ_ψ, the outphasing input signals $s_1(t)$ and $s_2(t)$ were adjusted according to the nearest neighbor principle.

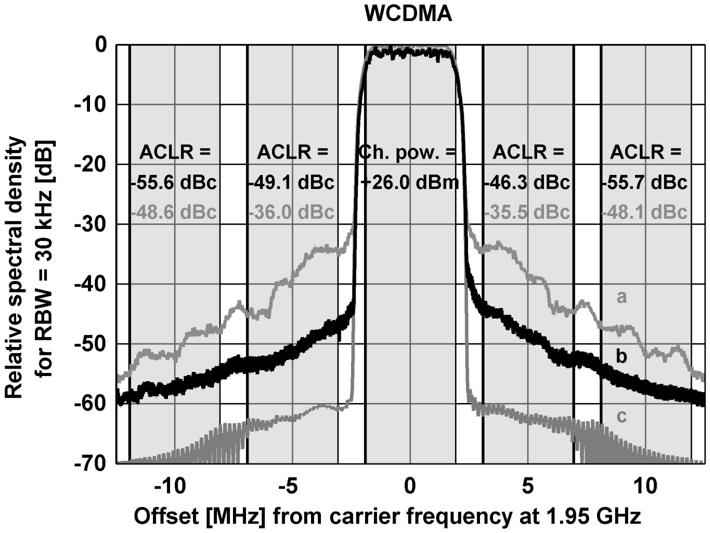


Figure 11.6: *Measured WCDMA spectrum at 1.95 GHz.*
(a) Measured WCDMA spectrum without DPD. The measured ACLR is printed in gray.
(b) When DPD is applied to (a). The measured ACLR is printed in black.
(c) Spectrum of estimation signal. Spectrum of validation signal was similar.

11.4.1 Measured performance of WCDMA signal

The PAPR of the WCDMA uplink signal was 3.5 dB. The spectrum of the estimation data is shown in Figure 11.6(c). For the WCDMA signal at 1.95 GHz without predistortion, the measured ACLR at 5 MHz and 10 MHz offsets were -35.5 dBc and -48.1 dBc, respectively. The spectrum is shown in Figure 11.6(a). The estimation output data $y(t)$ were used in the predistortion method to extract the model parameters, with $n = 5$. The ACLR is a measure describing the amount of leakage into adjacent channels that can be tolerated, and the standards for WCDMA are -33 dBc and -43 dBc at 5 MHz and 10 MHz offsets, respectively.

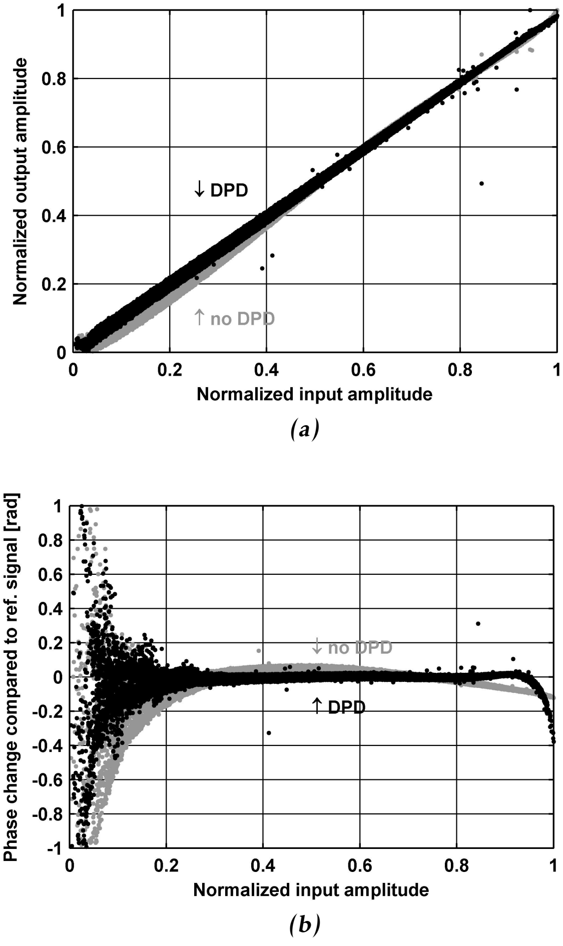
The predistorted input signals, $s_{1,P}(t)$ and $s_{2,P}(t)$, were computed for the validation input signal, resulting in an output spectrum as shown in Figure 11.6(b). The power spectral densities of the predistorted input is similar to that of the nonpredistorted input signal, and therefore not included (similarly for the LTE signal). With predistortion, the measured ACLR at 5 MHz and 10 MHz offsets were -46.3 dBc and -55.6 dBc, respectively. Thus, the measured ACLR at 5 MHz and at 10 MHz offsets were improved by 10.8 dB and 7.5 dB, respectively. The average power at 1.95 GHz was +26.0 dBm with 16.5 % PAE. It is clear that the predistortion reduces the spectral leakage.

Figure 11.7 shows the measured AM-AM (output amplitude vs. input amplitude) and AM-PM (phase change vs. input amplitude) characteristics with and without DPD for the WCDMA signal. The upper figure shows the amplitude modulation, and should ideally be a straight line from lower left corner (0,0) to the upper right (1,1), such that the output amplitude equals the input amplitude for the whole range of the signal. If this is not the case, there will be amplitude distortions. Here, the improvement can be seen in normalized amplitudes smaller than 0.4. The lower plot shows the phase distortion, and the ideal is zero. It can be seen that the DPD reduces the phase distortion for normalized amplitudes in the range $0.05 \lesssim |s| \lesssim 0.95$. For amplitudes close to one, the distortion is slightly worse with a predistorter than without. This is due to the polynomial fit of the PA model, which has the best fit in the middle region where the density of data points is largest.

11.4.2 Measured performance of LTE signal

The PAPR of the LTE uplink signal was 6.2 dB and the spectrum of the estimation data sets is shown in Figure 11.8(c). For the LTE signal at 1.95 GHz without predistortion, the measured ACLR at 5 MHz offset was -34.1 dBc. The spectrum is shown in Figure 11.8(a). The estimation output data $y(t)$ were used in the predistortion method to extract the model parameters with $n = 5$. The predistorted input signals, $s_{1,P}(t)$ and $s_{2,P}(t)$, were computed for the validation input signal, resulting in an output spectrum as shown in Figure 11.8(b). With the predistorted spectrum in Figure 11.8(b), a small asymmetry can be observed, which was expected due to the asymmetrical frequency spectrum of the reference signal. With predistortion, the measured ACLR at 5 MHz offset was -43.5 dBc. Thus, the measured ACLR at 5 MHz offset was improved by 9.4 dB. The average power at 1.95 GHz was +23.3 dBm with 8.0 % PAE.

Figure 11.9 shows the measured AM-AM and AM-PM characteristics with and without DPD for the LTE signal. The amplitude mapping in the upper figure should ideally be a straight line from the lower left corner to the upper right one, and the bottom figure should be zero for all input amplitudes. The figure shows that the amplitude and phase errors are significantly reduced for small amplitudes, with a normalized amplitude $|s| \lesssim 0.4$.



(a)

(b)

Figure 11.7: *(a) Measured AM-AM characteristics (output amplitude vs. input amplitude) with DPD (black) and without DPD (gray) for WCDMA signal. (b) Measured AM-PM characteristics (phase change vs. input amplitude) with DPD (black) and without DPD (gray) for WCDMA signal.*

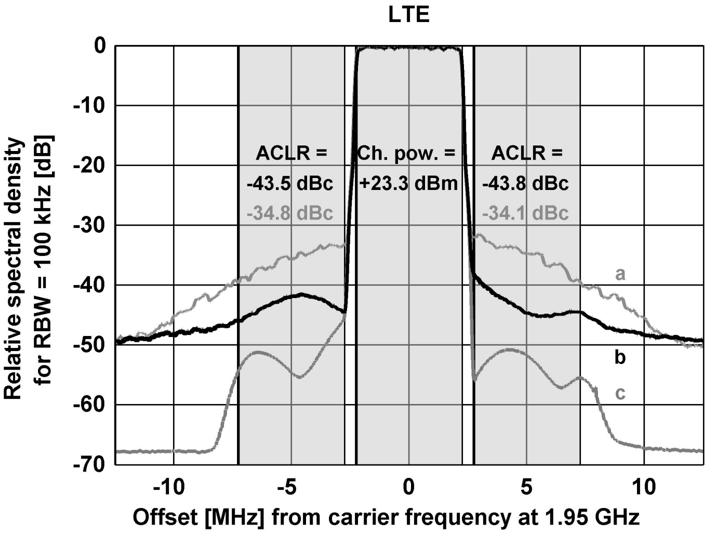
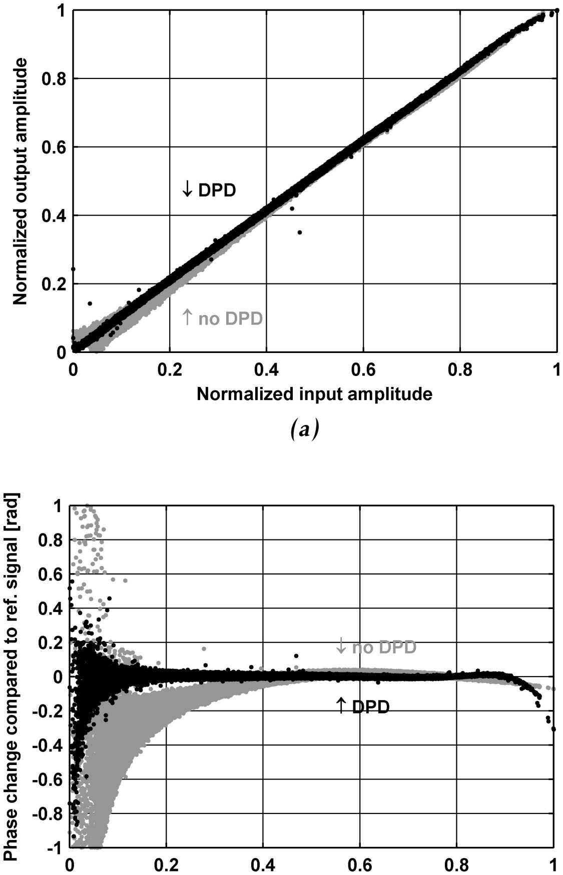


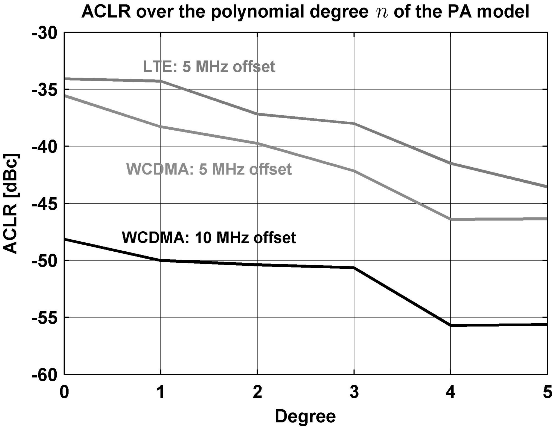
Figure 11.8: *Measured LTE spectrum at 1.95 GHz.*
(a) Measured LTE spectrum without DPD. The measured ACLR is printed in gray.
(b) When DPD is applied to (a). The measured ACLR is printed in black.
(c) Spectrum of estimation signal. Spectrum of validation signal was similar.



(a)

(b)

Figure 11.9: *(a) Measured AM-AM characteristics (output amplitude vs. input amplitude) with DPD (black) and without DPD (gray) for LTE signal. (b) Measured AM-PM characteristics (phase change vs. input amplitude) with DPD (black) and without DPD (gray) for LTE signal.*



© 2013 IEEE

Figure 11.10: *Measured ACLR depending on the polynomial degree n of the PA model. Degree $n = 0$ represents the performance without predistortion. The nonlinear modeling and distortion clearly improves the performance by reducing the ACLR.*

11.4.3 Evaluation of polynomial degree

A small evaluation of the impact of polynomial degree in the PA model has been performed, and the result is presented in Figure 11.10. It is clear that the addition of nonlinear terms improves the ACLR and reduces the spectral leakage. Polynomials with orders above $n = 5$ did not further improve the results significantly. A discussion on the impact of the choice of data points used in the LS problem can be found in Section 9.4 on page 143.

11.4.4 Summary

The measured performance of the PA for modulated signals is summarized in Table 11.6. The table shows measured ACLR with DPD, without DPD, and the required (Req) ACLR for the WCDMA [3GP] and the LTE [3GPP] standards. In measurements at 1.95 GHz, the DPD proved to be successful and improved the WCDMA ACLR at 5 MHz and 10 MHz offsets by 10.8 dB and 7.5 dB, respectively. The LTE ACLR at 5 MHz offset was improved by 9.4 dB. Thus, the predistortion method improves the measured ACLR to have at least 12.6 dB of margin to the requirements [3GP, 3GPP]. The measured ACLR at 5 MHz is comparable to state-of-the-art WCDMA transceivers [Huang et al., 2010].

To compare the DPD performance to the achievable ACLR, a small simulation study has been performed. Assuming a PA with 35 dB of dynamic range (neglecting phase distortions), i.e. assuming $g_1 = 0.509$ and $g_2 = 0.491$, and a polynomial degree of $n = 5$, the computed achievable ACLR at 5 MHz and 10 MHz is ~3 dB better compared to the measurements with the WCDMA signal. Similarly, the computed achievable ACLR at 5 MHz is ~2 dB better compared to the measurements with the LTE signal.

Table 11.6: *Measured Spectral Performance at 1.95 GHz for WCDMA and LTE Uplink Signals with Predistortion (using $n = 5$) and without.*

	Measured Parameter	**Req**	**Without DPD**	**With DPD**
WCDMA	ACLR @ 5 MHz [dBc]	-33	-35.5	-46.3
	ACLR @ 10 MHz [dBc]	-43	-48.1	-55.6
LTE	ACLR @ 5 MHz [dBc]	-30	-34.1	-43.5

As discussed in Section 9.4 on page 143, the polynomial fit is best in the middle, and in intervals where there is most data points. For the signals in this thesis, that is in the center of the interval, see Figure 11.2 for the distribution of the different signal types used. As seen in Figures 11.7 and 11.9, this is where the predistorter improves the performance. The predistorter is based on inversion of the PA models estimated using least squares. Since the inversion is almost perfect, see Figure 10.2 for the analytical inversion, the misfit at the smallest and largest input amplitudes can be assumed to be correlated with the polynomial fit of the PA model. The nonlinearity functions can be compared for different signal types, and though the overall appearance is very similar, a small shift can be seen, such that the fit has been adapted to the signal type. That is, for an LTE signal, the functions $\hat{f}_k$ differ a bit from the ones estimated for a WCDMA signal. This can be seen for lower amplitudes in particular, where the LTE signal has a higher signal density than the WCDMA.

12 Concluding remarks

There are multiple applications where a model of an inverse is needed. These include power amplifier predistortion, sensor calibration, feedforward control and inverse kinematics in robotics. In this thesis we have discussed inverse system identification and predistortion for outphasing power amplifiers.

Inverse system identification The inverse models in this thesis have been estimated with the purpose of using them in cascade with the system itself, as an inverter. A good inverse model in this setting is one that, when used in series with the original system, reconstructs the original input.

In this thesis, a classification and analysis of various inverse model estimation approaches is provided, and a characterization of earlier work. In METHOD A a forward model should be found and inverted, analytically or numerically. Two possibilities exist for METHOD B– estimate a preinverse in series with either a model of the system (B1), or the system itself (B2). METHOD C describes a postinverse, where the output and the input change place in the estimation process. In power amplifier predistortion, there are two common methods to find the inverse, DLA corresponding to METHOD B1 and ILA corresponding to METHOD C. Rather often, one method is chosen and the preinverse is estimated and an improvement is shown in performance. One goal here has been to evaluate the methods themselves and to analyze the results from estimating inverse models in different ways.

Two special cases have been analyzed comparing the methods when the inverse is estimated directly (METHOD C) and when an inverse is based on a forward model (METHOD A). It is shown for linear time-invariant dynamical systems with noise-free measurements that the weighting of an approximate model will be adjusted to reflect the intended use as an inverse if METHOD C is used for the identification. This has also been illustrated by an example. For Hammerstein systems with a white input, approximate linear models from METHOD A and METHOD C are the same, up to a constant. For colored inputs and Wiener systems this does not hold and an extra weighting factor will be present, meaning that the inverted forward model and the inverse model will differ by more than a constant gain.

For a forward model and a postinverse $\mathcal{T}$, one set of measurements is enough to find the optimal model. For a preinverse $\mathcal{R}$ this is not true. Since the preinverse will change the characteristics of the signal and the true system is affected by noise, more information is needed. It has been shown that the problems of finding a preinverse $\mathcal{R}$ and a postinverse $\mathcal{T}$ are fundamentally different.

For noise-free data and a model structure that is flexible enough, the true inverse will reconstruct the input signal. However, when there is noise present, we have shown that the true inverse will not be optimal, and that other models and model structures can lead to a better preinverse and postinverse.

Since the preinverse changes the input, the original input to the system could be very different from the predistorted one (for power amplifiers the predistorted one is generally more broadband). The noise contribution should also be taken into account when the inverse is constructed. Therefore, it is necessary to use multiple measurements. METHOD B2 uses the system in measurements to construct a preinverse $\mathcal{R}$. The method demands multiple experiments but finds a preinverse that captures the systematic noise contributions and the changed characteristics of the predistorted input signal to the system.

The goal of this thesis has been to investigate the problems connected to inverse system identification and possibly to find the best method, but there are still many open questions. The different methods need to be evaluated and the differences investigated. Here, the special cases of LTI systems and block-oriented systems have been looked at, but more general results would be interesting. We know that there is not one method that will always be best, that the decision should be based on how the inverse should be used. Different noise levels and noise types should be investigated to make the basis of a more general framework. Also, different types of nonlinearities could be evaluated. This thesis covers the SISO case, the multiple input-multiple output MIMO case could be investigated. The method comparisons in this thesis and the references herein have been based on theory and simulations. To evaluate the theory in measurements for PA predistortion would also show the applicability of the results.

Outphasing power amplifier predistortion In this thesis, the *predistortion* problem has been investigated for a type of PA called outphasing power amplifier, where the input signal is decomposed into two branches that are amplified separately by highly efficient nonlinear amplifiers, and then recombined. If the decomposition and summation of the two parts are not perfect, nonlinear terms will be introduced in the output, and predistortion is needed. The goal is to obtain a predistorter that counteracts the nonlinearities introduced by the amplifier.

Here, a predistorter has been constructed based on a model of the PA. In a first method, the structure of the outphasing amplifier has been used to model the distortion, and from this model, a predistorter can be estimated. However, this involves solving two nonconvex optimization problems, and the risk of obtaining a suboptimal solution. Exploring the structure of the PA, the problem has been reformulated so that the PA modeling basically can be done by solving two least-squares (LS) problems, which are convex and can be solved efficiently. In a second step, an analytical description of an ideal predistorter can be used to obtain a predistorter estimate. Another possibility is to compute the predistorter without a PA model by estimating the inverse directly. The methods have been evaluated in simulations and in measurements, and it is shown that the predistortion improves the linearity of the overall power amplifier system.

Interesting expansions would be to add dynamics to the models. Two ways of adding dynamics have been evaluated which did not result in improved methods, but there are many other possibilities. The measurements did not indicate a large dynamic influence, however, extending the method to include possible dynamics would extend the field of application. Also the noise influence was minor, but less ideal conditions could be evaluated. Now, the noise has a large impact in the normalization and the estimation of gain factors in the two branches, since these depend on only one and two measurements, respectively. This could be made more robust by looking at multiple measurements.

A

Power amplifier implementation

The outphasing power amplifiers used for the measurements presented in Chapter 11 and the power amplifier modeling in Chapter 9 have been constructed by Jonas Fritzin, Christer Svensson and Atila Alvandpour at the Division of Electronic Devices, Linköping University, Linköping, Sweden. The results and pictures in this chapter are all measured and reproduced with the authors' permission and are published here for sake of completeness.

As described in Section 8.2, a power amplifier can be characterized by different measures, such as the efficiency and the gain. For the PA beginner, a quick review of these concepts and the others in Section 8.2 could be useful. See also the Glossary in the preamble (page xvi).

The power amplifiers are of outphasing-type. The amplifier in each branch is a Class D amplifier, based on inverters, that switches between V_{DD} and *GND*.

A.1 +10.3 dBm Class-D outphasing RF amplifier in 90 nm CMOS

The chip used for validation of the nonconvex method in Section 11.3 can be seen in the chip photo in Figure A.1 and the sketch in Figure A.2. The PA is a Class D outphasing amplifier with an inverter-based output stage and an on-chip transformer as power combiner. More specifics can be found in Fritzin [2011] and Fritzin et al. [2011a].

Figure A.3a shows the measured maximum output power (P_{out}), the *drain efficiency* (DE) and the *power-added efficiency* (PAE) over frequency for the power amplifier. V_{DD} and V_{bias} were 1.3 V and 0.65 V, respectively. The 3 dB bandwidth was 2 GHz (1-3 GHz). The output power at 2 GHz was +10.3 dBm with DE and PAE of 39 % and 33 %, respectively, with a gain of 23 dB from the buffers to the

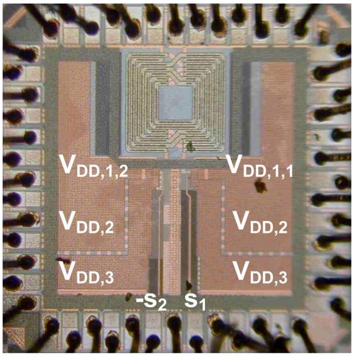


Figure A.1: *Photo of the chip with size 1x1mm^2.*

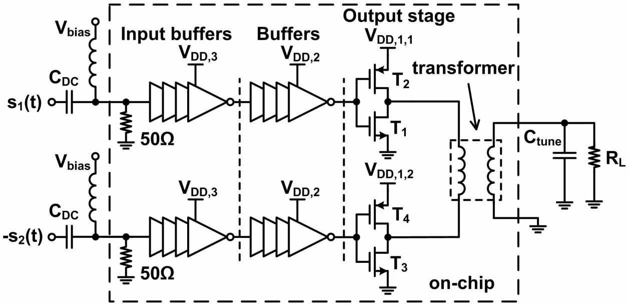
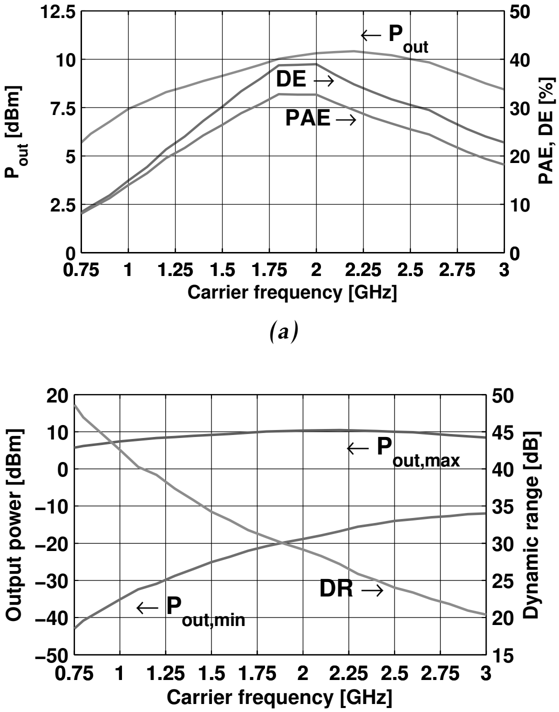


Figure A.2: *Implemented outphasing amplifier with inverters in the output stage.*



(a)

(b)

Figure A.3: *(a) Measured output power (P_{out}), DE and PAE over frequency. (b) Measured maximum output power, $P_{out,max}$, minimum output power, $P_{out,min}$, and dynamic range, DR, over frequency.*

output. The minimum and maximum output power and DR of the PA are plotted in Figure A.3b, where $P_{\text{out,max}} = P_{\text{out}}$ in Figure A.3a.

A.2 +30 dBm Class-D outphasing RF amplifier in 65 nm CMOS

The PA used for validation in Section 11.4 is described in more detail in Fritzin et al. [2011c] , but some basic characteristics can be found here. The chip photo can be seen in Figure A.4. Figure A.5 shows the outphasing PA, based on a Class D amplifier stage utilizing a cascode configuration illustrated in Figure A.6a. This configuration improves the life-time of the transistors by achieving a low on-resistance in the on-state and distributing the voltage stress in the off state which assures that the *root mean square* (RMS) electric fields across the gate oxide is kept low. The output stage is driven by an AC-coupled low-voltage driver operating at 1.3 V, V_{DD1}, to allow a 5.5 V, V_{DD2}, supply without excessive device voltage stress as discussed in Fritzin et al. [2011b] and Fritzin et al. [2011c]. The

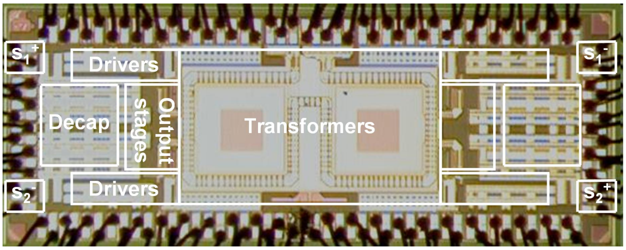


Figure A.4: *Photo of the chip with size 2.5x1.0mm^2. The photo has the same orientation as the simplified PA schematic in Figure A.5.*

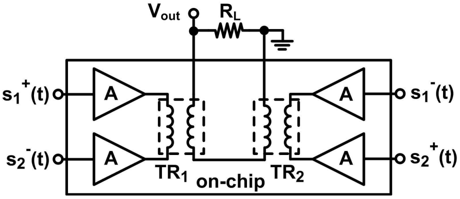


Figure A.5: *The implemented Class-D outphasing RF PA using two transformers to combine the outputs of four amplifier stages.*

chip was attached to an FR4 PCB and connected with bond-wires.

The measured output power, drain efficiency and power-added efficiency over frequency and outphasing angle, φ in (8.11) (where $\varphi = 2\Delta_\psi$), for $V_{\mathrm{DD1}} = 1.3\,\mathrm{V}$ and $V_{\mathrm{DD2}} = 5.5\,\mathrm{V}$ is shown in Figures A.7. The output power at 1.95 GHz was +29.7 dBm with a PAE of 26.6 % (in all drivers). The PA had a peak to minimum power ratio of ~35 dB and the gain was 26 dB from the drivers to the output. The DC power consumption of the smallest drivers was considered as input power.

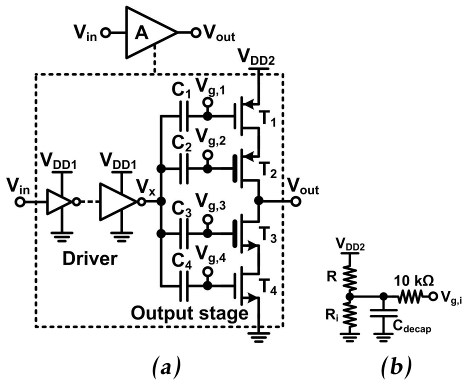
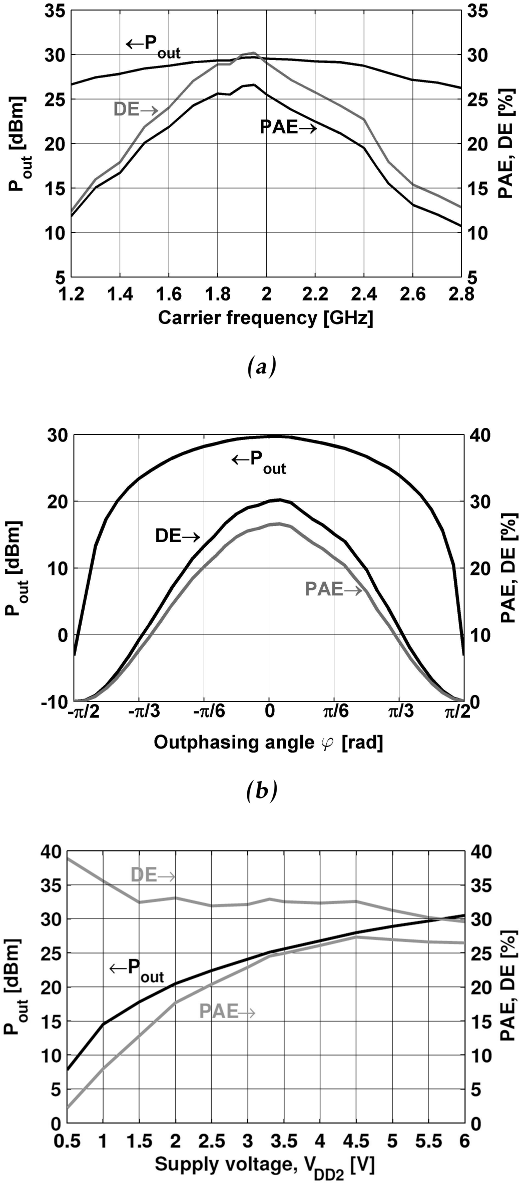


Figure A.6: *(a) The Class-D stage used in the outphasing PA Fritzin et al. [2011c]. C_1-C_4 are MIM capacitors. (b) Off-chip biasing resistors, R and R_i.*



(a)

(b)

(c)

Figure A.7: *Measured P_{out}, DE and PAE for $V_{DD1} = 1.3\,V$ and $V_{DD2} = 5.5\,V$ [Fritzin et al., 2011c]:*
(a) over carrier frequency.
(b) over outphasing angle, φ, at 1.95 GHz.
(c) Measured P_{out}, DE and PAE over V_{DD2} for $V_{DD1} = 1.3\,V$ at 1.95 GHz.

Bibliography

3GP. TS 25.101 v10.2.0 (2011-06). 3rd Generation Partnership Project; Technical specification group radio access network; user equipment (UE) radio transmission and reception (FDD), Release 10. Cited on page 179.

3GPP. TS 36.101 v10.3.0 (2011-06). 3rd Generation Partnership Project; Technical specification group radio access network; evolved universal terrestrial radio access (E-UTRA); user equipment (UE) radio transmission and reception, Release 10. Cited on page 179.

Emad Abd-Elrady, Li Gan, and Gernot Kubin. Direct and indirect learning methods for adaptive predistortion of IIR Hammerstein systems. *Elektrotechnik & Informationstechnik*, 125(4):126–131, April 2008. Cited on pages 30, 57, and 60.

Agilent. Agilent PN 89400-14, using error vector magnitude measurements to analyze and troubleshoot vector-modulated signals - *Product Note* 2000. http://cp.literature.agilent.com/litweb/pdf/5965-2898e.pdf. Accessed January, 2013. Cited on page 116.

Lars Ahlin, Jens Zander, and Ben Slimane. *Principles of Wireless Communications*. Studentlitteratur, 2006. ISBN 91-44-03080-0. Cited on page 165.

Shoaib Amin, Efrain Zenteno, Per N. Landin, Daniel Rönnow, Magnus Isaksson, and Peter Händel. Noise impact on the identification of digital predistorter parameters in the indirect learning architecture. In *Swedish Communication Technologies Workshop (SWE-CTW)*, pages 36–39, Lund, Sweden, October 2012. Cited on page 59.

Anritsu. Adjacent channel power ratio (ACPR) - *Application Note, Rev. A*. February 2001. http://www.us.anritsu.com/downloads/files/11410-00264.pdf, accessed January, 2013. Cited on page 115.

Marcus Arvidsson and Daniel Karlsson. Attenuation of harmonic distortion in loudspeakers using non-linear control. Master's thesis, Linköping University, 2012. LITH-ISY-EX–12/4579–SE. Cited on page 34.

Karl J. Åström and Pieter Eykhoff. System identification - a survey. *Automatica*, 7:123–162, 1971. Cited on pages 41 and 72.

Karl J. Åström and Tore Hägglund. *Advanced PID Control*. ISA - Instrumentation, Systems, and Automation Society, Second edition, 2005. ISBN 1-55617-942-1. Cited on pages 24 and 58.

Ahmed Birafane and Ammar B. Kouki. Phase-only predistortion for LINC amplifiers with Chireix-outphasing combiners. *IEEE Transactions on Microwave Theory and Techniques*, 53(6):2240–2250, June 2005. Cited on pages 122 and 126.

Ahmed Birafane, Mohamed El-Asmar, Ammar B. Kouki, Mohamed Helaoui, and Fadhel M. Ghannouchi. Analyzing LINC systems. *IEEE Microwave Magazine*, 11(5):59–71, August 2010. Cited on page 122.

André Carvalho Bittencourt, Patrik Axelsson, Ylva Jung, and Torgny Brogårdh. Modeling and identification of wear in a robot joint under temperature uncertainties. In *18th IFAC World Congress*, pages 10293–10299, Milan, Italy, August 2011. Not cited.

Åke Björck. *Numerical Methods for Least Squares Problems*. Siam, 1996. Cited on page 163.

Ylva Björk and Ebba Wilhelmsson. Linearisation of micro loudspeakers using adaptive control. Master's thesis, Linköping University, 2014. LITH-ISY-EX–13/4734–SE. Cited on page 2.

Lennart Blanken, Ids van den Meijdenberg, and Tom Oomen. Inverse system estimation for feedforward: A kernel-based approach for non-causal systems. In *18th IFAC Symposium on System Identification (SYSID)*, pages 1050–1055, Stockholm, Sweden, July 2018. Cited on page 71.

Frank Boeren, Dennis Bruijnen, Niels van Dijk, and Tom Oomen. Joint input shaping and feedforward for point-to-point motion: Automated tuning for an industrial nanopositioning system. *IFAC Mechatronics*, 24(6):572–581, September 2014. Cited on page 2.

Frank Boeren, Abhishek Bareja, Tom Kok, and Tom Oomen. Unified ILC framework for repeating and varying tasks: A frequency domain approach with application to a wire-bonder. In *54th IEEE Conference on Decision and Control (CDC)*, pages 6724–6729, Osaka, Japan, December 2015. Cited on page 27.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. Cited on page 146.

Julian J. Bussgang. Crosscorrelation functions of amplitude-distorted gaussian signals. Technical Report Technichal Report 216, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, March 1952. Cited on page 73.

Claudia Califano, Salvatore Monaco, and Dorothé Normand-Cyrot. On the discrete-time normal form. *IEEE Transactions on Automatic Control*, 43(11): 1654–1658, November 1998. Cited on page 28.

Jessica Chani-Cahuana, Christian Fager, and Thomas Eriksson. A new variant of the indirect learning architecture for the linearization of power amplifiers. In *10th European Microwave Integrated Circuits Conference*, pages 444–447, Paris, France, September 2015. Cited on page 84.

Jessica Chani-Cahuana, Per Niklas Landin, Christian Fager, and Thomas Eriksson. Iterative learning control for RF power amplifiers linearization. *IEEE Transactions on Microwave Theory and Techniques*, 64(9):2778–2789, September 2016. Cited on pages 60 and 61.

Tsan-Wen Chen, Ping-Yuan Tsai, Jui-Yuan Yu, and Chen-Yi Lee. A sub-mW all-digital signal component separator with branch mismatch compensation for OFDM LINC transmitters. *IEEE Journal of Solid-State Circuits*, 46(11):2514–2523, November 2011. Cited on page 126.

Hektbi Chireix. High power outphasing modulation. *IRE*, 23:1370–1392, November 1935. Cited on page 120.

Donald C. Cox. Linear amplification with nonlinear components. *IEEE Transactions on Communication*, COM-23:1942–1945, December 1974. Cited on page 120.

Steve C. Cripps. *RF Power Amplifiers for Wireless Communications*. Artech House, Second edition, 2006. ISBN 1-59693-018-7. Cited on pages 114 and 116.

Erik Dahlman, Stefan Parkvall, and Johan Sköld. *4G LTE/LTE-Advanced for Mobile Broadband*. Elsevier, 2011. ISBN 978-0-12-385489-6. Cited on page 165.

Germund Dahlquist and Åke Björck. *Numerical Methods in Scientific Computing, Vol I*. Siam, 2008. ISBN 978-0-898716-44. Cited on page 143.

Santosh Devasia. Technical notes and correspondance - should model-based inverse inputs be used as feedforward under plant uncertainty. *IEEE Transactions on Automatic Control*, 47(11):1865–1871, November 2002. Cited on page 59.

Lei Ding, G. Tong Zhou, Dennis R. Morgan, Zhengxiang Ma, J. Stevenson Kenney, Jaehyeong Kim, and Charles R. Giardina. A robust digital baseband predistorter constructed using memory polynomials. *IEEE Transactions on Communications*, 52(1):159–165, January 2004. Cited on pages 124 and 125.

Norman R. Draper and Harry Smith. *Applied Regression Analysis*. John Wiley & Sons, Third edition, 1998. ISBN 0-471-17082-8. Cited on page 16.

Martin Enqvist. *Linear Models of Nonlinear Systems*. Linköping Studies in Science and Technology. Dissertations. No 985, Linköping University, Linköping, Sweden, SE-581 83 Linköping, Sweden, December 2005. Cited on pages 54 and 55.

Martin Enqvist and Lennart Ljung. Linear approximations of nonlinear FIR systems for separable input processees. *Automatica*, 41(3):459–473, March 2005. Cited on pages 54, 55, and 73.

Louis E. Frenzel. *Principles of Electronic Communication Systems*. McGraw-Hill, Second edition, 2003. ISBN 0-07-828131-8. Cited on pages 110, 113, 117, 120, and 165.

Jonas Fritzin. *CMOS RF Power Amplifiers for Wireless Communications*. Linköping Studies in Science and Technology. Dissertations. No 1399, Linköping University, Linköping, Sweden, SE-581 83 Linköping, Sweden, November 2011. Cited on pages 115, 122, 123, 167, 168, and 185.

Jonas Fritzin, Ylva Jung, Per N. Landin, Peter Händel, Martin Enqvist, and Atila Alvandpour. Phase predistortion of a Class-D outphasing RF amplifier in 90nm CMOS. *IEEE Transactions on Circuits and Systems-II: Express Briefs*, 58(10): 642–646, October 2011a. Cited on pages 23, 60, 127, and 185.

Jonas Fritzin, Christer Svensson, and Atila Alvandpour. A +32dBm 1.85GHz Class-D outphasing RF PA in 130nm CMOS for WCDMA/LTE. In *IEEE European Solid-State Circuits Conference (ESSCIRC)*, pages 127–130, Helsinki, Finland, September 2011b. Cited on page 187.

Jonas Fritzin, Christer Svensson, and Atila Alvandpour. A wideband fully integrated +30dBm Class-D outphasing RF PA in 65nm CMOS. In *IEEE International Symposium on Integrated Circuits (ISIC)*, pages 25–28, Singapore, Singapore, December 2011c. Cited on pages 173, 187, 189, and 190.

Li Gan and Emad Abd-Elrady. Adaptive predistortion of IIR Hammerstein systems using the nonlinear filtered-x LMS algorithm. In *IEEE International Symposium on Wireless Communication Systems (ISWCS) 6th International Symposium on Communication Systems, Networks and Digital Signal Processing (CNSDSP 2008)*, pages 702–705, Graz, Austria, July 2008. Cited on page 125.

William A. Gardner. *Introduction to stochastic processes*. McGraw-Hill Book Co., Second edition, 1990. ISBN 0-07-022855-8. Cited on page 54.

Walter Gerhard and Reinhard Knöchel. Prediction of bandwidth requirements for a digitally based WCDMA phase modulated outphasing transmitter. In *The European Conference on Wireless Technology*, pages 97–100, Paris, France, October 2005a. Cited on page 167.

Walter Gerhard and Reinhard Knöchel. LINC digital component separator for single and multicarrier W-CDMA signals. *IEEE Transactions on Microwave Theory and Techniques*, 53(1):274–282, January 2005b. Cited on page 167.

Michel Gevers and Lennart Ljung. Optimal experimental designs, with respect to the intended model application. *Automatica*, 22(5):543–554, September 1986. Cited on page 57.

Fadhel M. Ghannouchi and Oualid Hammi. Behavioral modeling and predistortion. *IEEE Microwave magazine*, 10(7):52–64, December 2009. Cited on pages 60 and 124.

Pere L. Gilabert, Gabriel Montoro, and Eduard Bertran Alberti. On the Wiener and Hammerstein models for power amplifier predistorion. In *Asia-Pacific Conference Proceedings (APMC)*, Suzhou, China, December 2005. Cited on pages 60 and 125.

Pere L. Gilabert, Daniel D. Silveira, Gabriel Montoro, and Gottfried Magerl. RF-power amplifier modeling and predistortion based on a modular approach. In *European Microwave Integrated Circuits Conference*, pages 265–268, Manchester, UK, September 2006. Cited on page 125.

Pere L. Gilabert, Eduard Bertran, Gabriel Montoro, and Jordi Berenguer. FPGA implementation of an LMS-based real-time adaptive predistorter for power amplifiers. In *Circuits and Systems and TAISA Conference, 2009. NEWCAS-TAISA '09. Joint IEEE North-East Workshop on*, Toulouse, France, June-July 2009. Cited on page 163.

Lei Guan and Anding Zhu. Low-cost FPGA implementation of Volterra series-based digital predistorter for RF power amplifiers. *IEEE Transactions on Microwave Theory and Techniques*, 58(4):866–872, April 2010. Cited on pages 123 and 124.

Lei Guan and Anding Zhu. Green Communications: Digital predistortion for wideband RF power amplifiers. *IEEE Microwave Magazine*, 15(7):84–99, November 2014. Not cited.

Svante Gunnarsson, Ylva Jung, Clas Veibäck, and Torkel Glad. Io (implement and operate) first in an automatic control context. In *12th International CDIO Conference*, pages 238–249, Turku, Finland, June 2016. Not cited.

Mohamed Helaoui, Slim Boumaiza, and Fadhel M. Ghannouchi. On the outphasing power amplifier nonlinearity analysis and correction using digital predistortion technique. In *IEEE Radio and Wireless Symposium (RWS)*, pages 751–754, Orlando, FL, USA, January 2008. Cited on page 126.

Ronald M. Hirschorn. Invertibility of multivariable nonlinear control systems. *IEEE Transactions on Automatic Control*, 24(6):855–865, December 1979. Cited on pages 33 and 34.

Du Ho and Martin Enqvist. On the equivalence of forward and inverse IV estimators with application to quadcopter modeling. In *18th IFAC Symposium on System Identification (SYSID)*, pages 951–956, Stockholm, Sweden, July 2018. Cited on pages 72 and 75.

Qiuting Huang, Jürgen Rogin, Xinhua Chen, David Tschopp, Thomas Burger, Thomas Christen, Dimitris Papadopolous, Ilian Kouchev, Chiara Martelli, and Thomas Dellsperger. A tri-band SAW-less WCDMA/HSPA RF CMOS transceiver, with on-chip DC-DC converter connectable to battery. In *International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 60–61, San Fransisco, CA, USA, February 2010. Cited on pages 171 and 179.

Mazen Abi Hussein, Vivek Ashok Bohara, and Olivier Venard. On the system level convergence of ILA and DLA for digital predistortion. In *IEEE International Symposium on Wireless Communication Systems (ISWCS)*, pages 870–874, Paris, France, August 2012. Cited on pages 60 and 124.

Magnus Isaksson and Daniel Rönnow. A parameter-reduced Volterra model of dynamic RF power amplifier modeling based on orthonormal basis functions. *International Journal of RF and Microwave Computer-Aided Engineering*, 17 (6):542–551, November 2007. Cited on page 125.

Richard C. Jaeger and Travis N. Blalock. *Microelectronic Circuit Design*. McGraw-Hill, Third edition, 2008. ISBN 978-0-07-110203-2. Cited on pages 117 and 119.

Ylva Jung and Martin Enqvist. Estimating models of inverse systems. In *52nd IEEE Conference on Decision and Control (CDC)*, pages 7143–7148, Florence, Italy, December 2013. Cited on page 57.

Ylva Jung and Martin Enqvist. On estimation of approximate inverse models of block-oriented systems. In *17th IFAC Symposium on System Identification (SYSID)*, pages 1226–1231, Beijing, China, October 2015. Cited on page 57.

Ylva Jung, Jonas Fritzin, Martin Enqvist, and Atila Alvandpour. Least-squares phase predistortion of a +30dbm Class-D outphasing RF PA in 65nm CMOS. *IEEE Transactions on Circuits and Systems-I: Regular papers*, 60(7):1915–1928, July 2013. Cited on pages 127 and 133.

Thomas Kailath. *Linear Estimation*. Prentice Hall, 2000. ISBN 0-13-022464-2. Cited on page 73.

Mitsuo Kawato, Kazunori Furukawa, and Ryoji Suzuki. A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57(3):169–185, October 1987. Cited on page 3.

Peter B Kenington. *High-Linearity RF Amplifier Design*. Artech House, 2000. ISBN 1-58053-143-1. Cited on page 123.

Per N. Landin, Jonas Fritzin, Wendy Van Moer, Magnus Isaksson, and Atila Alvandpour. Modeling and digital predistortion of Class-D outphasing RF power amplifiers. *IEEE Transactions on Microwave Theory and Techniques*, 60(6): 1907–1915, June 2012. Cited on pages 127 and 133.

Per N. Landin, Annika E. Mayer, and Thomas Eriksson. MILA - A noise mitigation technique for RF power amplifier linearization. In *11th International Multi-Conference on Systems, Signals & Devices (SSD)*, Barcelona, Spain, February 2014. Cited on pages 61 and 124.

Jonas Linder and Martin Enqvist. Identification of systems with unknown inputs using indirect input measurements. *International Journal of Control*, 90(4): 729–745, 2017. Cited on page 3.

Lennart Ljung. *System Identification, Theory for the User*. Prentice Hall PTR, Second edition, 1999. ISBN 0-13-656695-2. Cited on pages 12, 13, 15, 16, 17, 18, 19, 41, 54, 57, 58, 71, 88, 132, 133, 135, and 163.

Lennart Ljung. Estimating linear time-invariant models of nonlinear time-varying systems. *European Journal of Control*, 7(2):203 – 219, 2001. Cited on page 54.

Lennart Ljung. *System Identification, Toolbox, User's Guide*. MathWorks, Sixth edition, 2003. Cited on page 78.

Pertti M. Mäkilä and Jonathan Partington. On linear models for nonlinear systems. *Automatica*, 39(1):1–13, January 2003. Cited on page 54.

Aarne Mämmelä. Commutation in linear and nonlinear systems. *Frequenz*, 60 (5-6):92–94, June 2006. Cited on page 30.

Ola Markusson. *Model and System Inversion with Applications in Nonlinear System Identification and Control*. TRITA-S3-REG-0201, Royal Institute of Technology, Stockholm, Sweden, SE-100 44 Stockholm, Sweden, 2001. Cited on pages 26, 27, 29, and 33.

Ola Markusson and Håkan Hjalmarsson. Iterative learning control of nonlinear non-minimum phase systems and its application to system and model inversion. In *Proc. of 40th IEEE Conference on Decision and Control (CDC)*, Orlando, FL, USA, December 2001. Cited on page 26.

Jaimin Mehta, Vasile Zoicas, Oren Eliezer, R. Bogdan Staszewski, Sameh Rezeq, Mitch Entezari, and Poras Bolsara. An efficient linearization scheme for a digital polar EDGE transmitter. *IEEE Transactions on Circuits and Systems-II: Express Briefs*, 57(3):193–197, March 2010. Cited on page 171.

Shervin Moloudi, Koji Takanami, Michael Youssef, Mohyee Mikhemar, and Asad Abidi. An outphasing power amplifier for software-defined radio transmitter. In *International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 568–569, San Fransisco, CA, USA, February 2008. Cited on page 126.

Gabriel Montoro, Pere L. Gilabert, Eduard Bertran, Albert Cesari, and José A. Garcia. An LMS-based adaptive predistorter for cancelling nonlinear memory effects in RF power amplifiers. In *Microwave Conference, 2007. APMC 2007. Asia-Pacific*, Bangkok, Thailand, December 2007. Cited on page 163.

Kevin L. Moore. *Iterative Learning Control for Deterministic Systems.* Springer Verlag, 1993. ISBN 978-1-4471-1914-2. Cited on page 26.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective.* The MIT Press, 2012. ISBN 978-0-262-01802-9. Cited on page 67.

Seong-Sik Myoung, Il-Kyoo Kee, Jong-Gwan Yook, Kyutae Lim, and Joy Laskar. Mismatch detection and compensation method for the LINC system using a closed-form expression. *IEEE Transactions on Microwave Theory and Techniques*, 56(12):3050–3057, December 2008. Cited on page 126.

Charles Nader, Per Niklas Landin, Wendy Van Moer, Niclas Björsell, Peter Händel, and Magnus Isaksson. Peak-to-average power reduction versus digital predistortion in OFDM based systems. In *IEEE MTT-S International Microwave Symposium Digest*, Baltimore, MD, USA, June 2011. Cited on page 125.

Henna Paaso and Aarne Mämmelä. Comparison of direct learning and indirect learning predistortion architechtures. In *IEEE International Syposium on Wireless Communication Systems (ISWCS)*, pages 309–313, Reykjavik, Iceland, October 2008. Cited on pages 30, 57, and 60.

Rik Pintelon and Johan Schoukens. *System Identification - A Frequency Domain Approach.* IEEE Press and John Wiley & Sons, Second edition, 2012. ISBN 978-0-470-64037-1. Cited on pages 12, 54, 55, 57, and 58.

Behzad Razavi. *RF Microelectronics.* Prentice Hall, 1998. ISBN 0-13-887571-5. Cited on pages 114 and 117.

Rohde & Schwarz. Application note, 1GP67: Phase adjustment of two MIMO signal sources with option B90. Cited on pages 167 and 173.

Luca Romanò, Luigi Panseri, Carlo Samori, and Andrea L. Lacaita. Matching requirements in LINC transmitters for OFDM signals. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 53(7):1572–1578, July 2006. Cited on pages 122 and 126.

Walter Rudin. *Principles of Mathematical Analysis.* McGraw-Hill Book Co., Third edition, 1976. ISBN 0-07-085613-3. Cited on pages 132 and 153.

Wilson J. Rugh. *Linear System Theory.* Prentice-Hall, Second edition, 1996. ISBN 0-13-441205-2. Cited on page 24.

Shankar Sastry. *Nonlinear Systems – Analysis, Stability and Control.* Springer Verlag, New York, 1999. ISBN 0-387-98513-1. Cited on pages 27 and 31.

Martin Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems.* John Wiley & Sons, New York, 1980. ISBN 0-471-04455-5. Cited on pages 31 and 32.

Johan Schoukens, Rik Pintelon, and Tadeusz Dobrowiecki. Parametric and nonparametric identification of linear systems in the presence of nonlinear distortions – A frequency domain approach. *IEEE Transactions on Automatic Control*, 43(2):176–190, February 1998. Cited on page 54.

Johan Schoukens, Rik Pintelon, Tadeusz Dobrowiecki, and Yves Rolain. Identification of linear systems with nonlinear distortions. *Automatica*, 41(3):491–504, March 2005. Cited on page 54.

Maarten Schoukens, Rik Pintelon, and Yves Rolain. Parametric identification of parallel Hammerstein systems. *IEEE Transactions on Automatic Control*, 60 (12):3931–3938, December 2011. Cited on page 53.

Maarten Schoukens, Jules Hammenecker, and Adam Cooman. Obtaining the preinverse of a power amplifier using iterative learning control. *IEEE Transactions on Microwave Theory and Techniques*, 65(11):4266–4273, Nov 2017. Cited on page 61.

Torsten Söderström and Petre Stoica. *System Identification*. Prentice Hall, 1989. ISBN 0-13-881236-5. Cited on page 12.

Michael Soudan and Christian Vogel. Correction structures for linear weakly time-varying systems. *IEEE Transactions on Circuits and Systems-I: Regular papers*, 59(9):2075–2084, September 2012. Cited on page 27.

Mehdi Tavan, Mahdi Aliyari Shoorehdeli, and Amir Reza Zare Bidaki. Stability of feedback error learning for linear systems. In *Proc. of 18th IFAC World Congress*, Milan, Italy, August 2011. Cited on page 3.

John Tsimbinos and Kenneth V. Lever. Computational complexity of Volterra based nonlinear compensators. *Electronic Letters*, 32(9):852–854, April 1996. Cited on page 124.

Murali Tummla, Michael T. Donovan, Bruce E. Watkins, and Robert North. Volterra series based modeling and compensation of nonlinearities in high power amplifiers. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2417–2420 vol. 3, Munich, Germany, April 1997. Cited on pages 32 and 124.

Johanna Wallén. *Estimation-Based Iterative Learning Control*. Linköping Studies in Science and Technology. Dissertations. No 1358, Linköping University, Linköping, Sweden, SE-581 83 Linköping, Sweden, February 2011. Cited on pages 26 and 27.

Gaoming Xu, Taijun Liu, Yan Ye, and Tiefeng Xu. FPGA implementation of augmented hammerstein predistorters for RF power amplifier linearization. In *Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, pages 481–484, Beijing, China, October 2009. Cited on page 125.

Hongtao Xu, Yorgos Palaskas, Ashoke Ravi, and Krishnamurthy Soumyanath. A highly linear 25dBm outphasing power amplifier in 32nm CMOS for WLAN application. In *IEEE European Solid-State Circuits Conference (ESSCIRC)*, pages 306–309, Seville, Spain, September 2010. Cited on page 123.

Jingshi Yao and Stephen I. Long. Power amplifier selection for LINC application. *IEEE Transactions on Circuits and Systems-II: Express Briefs*, 53(8):763–766, August 2006. Cited on page 123.

Xuejun Zhang, Lawrence E. Larson, Peter M. Asbeck, and Peter Nanawa. Gain/phase imbalance-minimization techniques for LINC transmitters. *IEEE Transactions on Microwave Theory and Techniques*, 49(12):2507–2516, June 2001. Cited on page 126.

Anding Zhu, Paul J. Draxler, Jonmei J. Yan, Thomas J. Brazil, Donald F. Kimball, and Peter M. Asbeck. Open-loop digital predistorter for RF power amplifiers using dynamic deviation reduction-based Volterra series. *IEEE Transactions on Microwave Theory and Techniques*, 56(7):1524–1534, July 2008. Cited on pages 32 and 124.

PhD Dissertations
Division of Automatic Control
Linköping University

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis No. 82, 1982. ISBN 91-7372-542-0.

A. J. M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis No. 86, 1982. ISBN 91-7372-586-2.

B. Bengtsson: On some control problems for queues. Thesis No. 87, 1982. ISBN 91-7372-593-5.

S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis No. 93, 1983. ISBN 91-7372-641-9.

H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis No. 104, 1983. ISBN 91-7372-718-0.

E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis No. 106, 1983. ISBN 91-7372-728-8.

K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis No. 162, 1987. ISBN 91-7870-170-8.

B. Wahlberg: On the identification and approximation of linear systems. Thesis No. 163, 1987. ISBN 91-7870-175-9.

S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis No. 194, 1988. ISBN 91-7870-380-8.

A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis No. 196, 1988. ISBN 91-7870-383-2.

M. Viberg: Subspace fitting concepts in sensor array processing. Thesis No. 217, 1989. ISBN 91-7870-529-0.

K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis No. 261, 1991. ISBN 91-7870-827-3.

F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis No. 271, 1992. ISBN 91-7870-876-1.

P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis No. 280, 1992. ISBN 91-7870-962-8.

T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis No. 285, 1992. ISBN 91-7870-989-X.

S. Andersson: On dimension reduction in sensor array signal processing. Thesis No. 290, 1992. ISBN 91-7871-015-4.

H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis No. 298, 1993. ISBN 91-7871-070-7.

I. Klein: Automatic synthesis of sequential control schemes. Thesis No. 305, 1993. ISBN 91-7871-090-1.

J.-E. Strömberg: A mode switching modelling philosophy. Thesis No. 353, 1994. ISBN 91-7871-430-3.

K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis No. 361, 1994. ISBN 91-7871-467-2.

T. McKelvey: Identification of state-space models from time and frequency data. Thesis No. 380, 1995. ISBN 91-7871-531-8.

J. Sjöberg: Non-linear system identification with neural networks. Thesis No. 381, 1995. ISBN 91-7871-534-2.

R. Germundsson: Symbolic systems – theory, computation and applications. Thesis No. 389, 1995. ISBN 91-7871-578-4.

P. Pucar: Modeling and segmentation using multiple models. Thesis No. 405, 1995. ISBN 91-7871-627-6.

H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis No. 407, 1995. ISBN 91-7871-629-2.

A. Helmersson: Methods for robust gain scheduling. Thesis No. 406, 1995. ISBN 91-7871-628-4.

P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis No. 436, 1996. ISBN 91-7871-424-8.

J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis No. 477, 1997. ISBN 91-7871-917-8.

M. Jirstrand: Constructive methods for inequality constraints in control. Thesis No. 527, 1998. ISBN 91-7219-187-2.

U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis No. 566, 1999. ISBN 91-7219-432-4.

A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis No. 571, 1999. ISBN 91-7219-450-2.

N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis No. 579, 1999. ISBN 91-7219-473-1.

K. Edström: Switched bond graphs: Simulation and analysis. Thesis No. 586, 1999. ISBN 91-7219-493-6.

M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis No. 608, 1999. ISBN 91-7219-615-5.

F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis No. 623, 2000. ISBN 91-7219-689-0.

V. Einarsson: Model checking methods for mode switching systems. Thesis No. 652, 2000. ISBN 91-7219-836-2.

M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis No. 653, 2000. ISBN 91-7219-837-0.

F. Tjärnström: Variance expressions and model reduction in system identification. Thesis No. 730, 2002. ISBN 91-7373-253-2.

J. Löfberg: Minimax approaches to robust model predictive control. Thesis No. 812, 2003. ISBN 91-7373-622-8.

J. Roll: Local and piecewise affine approaches to system identification. Thesis No. 802, 2003. ISBN 91-7373-608-2.

J. Elbornsson: Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis No. 811, 2003. ISBN 91-7373-621-X.

O. Härkegård: Backstepping and control allocation with applications to flight control. Thesis No. 820, 2003. ISBN 91-7373-647-3.

R. Wallin: Optimization algorithms for system analysis and identification. Thesis No. 919, 2004. ISBN 91-85297-19-4.

D. Lindgren: Projection methods for classification and identification. Thesis No. 915, 2005. ISBN 91-85297-06-2.

R. Karlsson: Particle Filtering for Positioning and Tracking Applications. Thesis No. 924, 2005. ISBN 91-85297-34-8.

J. Jansson: Collision Avoidance Theory with Applications to Automotive Collision Mitigation. Thesis No. 950, 2005. ISBN 91-85299-45-6.

E. Geijer Lundin: Uplink Load in CDMA Cellular Radio Systems. Thesis No. 977, 2005. ISBN 91-85457-49-3.

M. Enqvist: Linear Models of Nonlinear Systems. Thesis No. 985, 2005. ISBN 91-85457-64-7.

T. B. Schön: Estimation of Nonlinear Dynamic Systems — Theory and Applications. Thesis No. 998, 2006. ISBN 91-85497-03-7.

I. Lind: Regressor and Structure Selection — Uses of ANOVA in System Identification. Thesis No. 1012, 2006. ISBN 91-85523-98-4.

J. Gillberg: Frequency Domain Identification of Continuous-Time Systems Reconstruction and Robustness. Thesis No. 1031, 2006. ISBN 91-85523-34-8.

M. Gerdin: Identification and Estimation for Models Described by Differential-Algebraic Equations. Thesis No. 1046, 2006. ISBN 91-85643-87-4.

C. Grönwall: Ground Object Recognition using Laser Radar Data – Geometric Fitting, Performance Analysis, and Applications. Thesis No. 1055, 2006. ISBN 91-85643-53-X.

A. Eidehall: Tracking and threat assessment for automotive collision avoidance. Thesis No. 1066, 2007. ISBN 91-85643-10-6.

F. Eng: Non-Uniform Sampling in Statistical Signal Processing. Thesis No. 1082, 2007. ISBN 978-91-85715-49-7.

E. Wernholt: Multivariable Frequency-Domain Identification of Industrial Robots. Thesis No. 1138, 2007. ISBN 978-91-85895-72-4.

D. Axehill: Integer Quadratic Programming for Control and Communication. Thesis No. 1158, 2008. ISBN 978-91-85523-03-0.

G. Hendeby: Performance and Implementation Aspects of Nonlinear Filtering. Thesis No. 1161, 2008. ISBN 978-91-7393-979-9.

J. Sjöberg: Optimal Control and Model Reduction of Nonlinear DAE Models. Thesis No. 1166, 2008. ISBN 978-91-7393-964-5.

D. Törnqvist: Estimation and Detection with Applications to Navigation. Thesis No. 1216, 2008. ISBN 978-91-7393-785-6.

P-J. Nordlund: Efficient Estimation and Detection Methods for Airborne Applications. Thesis No. 1231, 2008. ISBN 978-91-7393-720-7.

H. Tidefelt: Differential-algebraic equations and matrix-valued singular perturbation. Thesis No. 1292, 2009. ISBN 978-91-7393-479-4.

H. Ohlsson: Regularization for Sparseness and Smoothness — Applications in System Identification and Signal Processing. Thesis No. 1351, 2010. ISBN 978-91-7393-287-5.

S. Moberg: Modeling and Control of Flexible Manipulators. Thesis No. 1349, 2010. ISBN 978-91-7393-289-9.

J. Wallén: Estimation-based iterative learning control. Thesis No. 1358, 2011. ISBN 978-91-7393-255-4.

J. D. Hol: Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS. Thesis No. 1368, 2011. ISBN 978-91-7393-197-7.

D. Ankelhed: On the Design of Low Order H-infinity Controllers. Thesis No. 1371, 2011. ISBN 978-91-7393-157-1.

C. Lundquist: Sensor Fusion for Automotive Applications. Thesis No. 1409, 2011. ISBN 978-91-7393-023-9.

P. Skoglar: Tracking and Planning for Surveillance Applications. Thesis No. 1432, 2012. ISBN 978-91-7519-941-2.

K. Granström: Extended target tracking using PHD filters. Thesis No. 1476, 2012. ISBN 978-91-7519-796-8.

C. Lyzell: Structural Reformulations in System Identification. Thesis No. 1475, 2012. ISBN 978-91-7519-800-2.

J. Callmer: Autonomous Localization in Unknown Environments. Thesis No. 1520, 2013. ISBN 978-91-7519-620-6.

D. Petersson: A Nonlinear Optimization Approach to H2-Optimal Modeling and Control. Thesis No. 1528, 2013. ISBN 978-91-7519-567-4.

Z. Sjanic: Navigation and Mapping for Aerial Vehicles Based on Inertial and Imaging Sensors. Thesis No. 1533, 2013. ISBN 978-91-7519-553-7.

F. Lindsten: Particle Filters and Markov Chains for Learning of Dynamical Systems. Thesis No. 1530, 2013. ISBN 978-91-7519-559-9.

P. Axelsson: Sensor Fusion and Control Applied to Industrial Manipulators. Thesis No. 1585, 2014. ISBN 978-91-7519-368-7.

A. Carvalho Bittencourt: Modeling and Diagnosis of Friction and Wear in Industrial Robots. Thesis No. 1617, 2014. ISBN 978-91-7519-251-2.

M. Skoglund: Inertial Navigation and Mapping for Autonomous Vehicles. Thesis No. 1623, 2014. ISBN 978-91-7519-233-8.

S. Khoshfetrat Pakazad: Divide and Conquer: Distributed Optimization and Robustness Analysis. Thesis No. 1676, 2015. ISBN 978-91-7519-050-1.

T. Ardeshiri: Analytical Approximations for Bayesian Inference. Thesis No. 1710, 2015. ISBN 978-91-7685-930-8.

N. Wahlström: Modeling of Magnetic Fields and Extended Objects for Localization Applications. Thesis No. 1723, 2015. ISBN 978-91-7685-903-2.

J. Dahlin: Accelerating Monte Carlo methods for Bayesian inference in dynamical models. Thesis No. 1754, 2016. ISBN 978-91-7685-797-7.

M. Kok: Probabilistic modeling for sensor fusion with inertial measurements. Thesis No. 1814, 2016. ISBN 978-91-7685-621-5.

J. Linder: Indirect System Identification for Unknown Input Problems: With Applications to Ships. Thesis No. 1829, 2017. ISBN 978-91-7685-588-1.

M. Roth: Advanced Kalman Filtering Approaches to Bayesian State Estimation. Thesis No. 1832, 2017. ISBN 978-91-7685-578-2.

I. Nielsen: Structure-Exploiting Numerical Algorithms for Optimal Control. Thesis No. 1848, 2017. ISBN 978-91-7685-528-7.

D. Simon: Fighter Aircraft Maneuver Limiting Using MPC: Theory and Application. Thesis No. 1881, 2017. ISBN 978-91-7685-450-1.

C. Veibäck: Tracking the Wanders of Nature. Thesis No. 1958, 2018. ISBN 978-91-7685-200-2.

C. Andersson Naesseth: Machine learning using approximate inference: Variational and sequential Monte Carlo methods. Thesis No. 1969, 2018. ISBN 978-91-7685-161-6.

FACULTY OF SCIENCE AND ENGINEERING

Linköping Studies in Science and Technology. Dissertations No. 1966, 2018
Department of Electrical Engineering

Linköping University
SE-581 83 Linköping, Sweden

www.liu.se

LINKÖPING
UNIVERSITY