BAYES MEETS BACH: APPLICATIONS OF BAYESIAN STATISTICS TO AUDIO RESTORATION

Hugo Tremonte de Carvalho

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientadores: Luiz Wagner Pereira Biscainho
Flávio Rainho Ávila

Rio de Janeiro
Janeiro de 2017

BAYES MEETS BACH: APPLICATIONS OF BAYESIAN STATISTICS TO AUDIO RESTORATION

Hugo Tremonte de Carvalho

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Luiz Wagner Pereira Biscainho, D.Sc.

Prof. Flávio Rainho Ávila, D.Sc.

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. Ralph dos Santos Silva, D.Sc.

Prof. Leonardo Tomazeli Duarte, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 2017

Tremonte de Carvalho, Hugo

Bayes meets Bach: Applications of Bayesian Statistics to Audio Restoration/Hugo Tremonte de Carvalho. – Rio de Janeiro: UFRJ/COPPE, 2017.

XXXII, 244 p.: il.; 29, 7cm.

Orientadores: Luiz Wagner Pereira Biscainho
Flávio Rainho Ávila

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2017.

Referências Bibliográficas: p. 235 – 244.

1. Audio Restoration. 2. Statistical Signal Processing. 3. Bayesian Statistics. 4. Monte Carlo Methods. I. Pereira Biscainho, Luiz Wagner *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

A Felipe Acker, por ter criado a Matemática Aplicada, "um curso para os desajustados [sic]", onde, ironicamente, me encaixei perfeitamente.

Agradecimentos

Para mim essa é uma das partes mais importantes da tese, pois aqui tenho a oportunidade de agradecer explicitamente a todos que me ajudaram, direta ou indiretamente, em toda minha trajetória de vida até esse ponto. Por isso, me reservei ao direito de escrevê-la em Português, pois acredito conseguir expressar melhor meus sentimentos em meu idioma nativo.

Iniciei minha trajetória acadêmica em 2007, cursando Graduação em Matemática Aplicada na UFRJ, e agora, dez anos depois, concluo uma etapa importante, o Doutorado. Não foi um caminho fácil, tive muito a aprender e muitas vezes de uma maneira um pouco assustadora. Tive que abrir mão de várias diversões e lazeres, mas hoje vejo que valeu a pena cada instante. Sem meus amigos, professores e alguns familiares próximos nada disso teria sido possível, e o mínimo que posso fazer para retribuir é esse breve agradecimento.

Essa lista não está em nenhuma ordem especial, e para complementá-la veja os agradecimentos na minha dissertação de Mestrado [1].

- Acredito que em primeiro lugar eu devo agradecer à vida, ao Universo e tudo o mais pela oportunidade que tive de estudar na UFRJ, bem como aos contribuintes do Brasil que ajudam a manter das Universidades públicas, e ter tido excelentes professores ao longo de toda a minha formação. Faço o meu melhor para honrar essa oportunidade que me foi dada.

- Agradeço aos meus pais pelo amor e carinho infinitos, mesmo achando que estou falando Grego quando falo sobre Matemática, o que não deixa de ser verdade! Sem o carinho e educação que vocês me deram eu jamais seria capaz de chegar até aqui.

- Agradeço aos meus orientadores Luiz Wagner e Flávio Ávila por me acompanharem e instruírem nessa importante etapa da minha formação, mesmo com as eventuais teimosias da minha parte, em particular no que diz respeito à informalidade da minha escrita. Vocês são muito mais que orientadores, mas sim grandes amigos, e espero que esses quatro anos de colaboração e amizade sejam

os primeiros de muitos que ainda virão pela frente! Agora os agradecimentos individuais:

  - Agradeço ao Luiz Wagner por todas as rimas sensacionais, trocadilhos infames, más-piadas, audições em sua casa e por ter quebrado meu preconceito com vários gêneros de música erudita, sem contar as breves avistadas a bandos de pegareios.
  - Agradeço ao Flávio pelas conversas sobre Ciência, seus breves ensinamentos sobre Economia nas horas de almoço na UERJ, nossas discussões saudáveis quando temos divergências de opiniões, e principalmente pelas visitas à casa da Magali.

- Agradeço aos membros da banca examinadora dessa tese, por aceitarem participar dessa avaliação, mesmo sendo necessário um grande deslocamento para isso da parte de alguns, e pelas excelentes sugestões para melhoria do trabalho.

- Agradeço ao professor Rolci Cipollati, chefe do Departamento de Matemática do IM/UFRJ no período em que lá fui Professor Substituto, principalmente por sua infinita paciência em lidar com as infinitas exigências de todos ao montar a grade de horários do semestre.

- Agradeço à Tia Deise, secretária academia da Graduação do IM/UFRJ, que resolveu infinitos pepinos meus na época da Graduação e Mestrado em Matemática Aplicada, e depois quando eu fui Professor Substituto. Ela faleceu no final de 2014, mas dedicou toda sua vida ao IM/UFRJ e seus alunos, trabalhando inclusive quando estava internada! Tive a honra imensa de tê-la presente em minha colação de grau.

- Agradeço a todos os meus alunos dos dois anos nos quais fui Professor Substituto do IM/UFRJ: duas turmas de Cálculo III em 2013/1 (Bacharelado em Química e Engenharia Mecânica), duas turmas de Cálculo I em 2013/2 (Bacharelado em Ciências Matemáticas e da Terra), Cálculo I (Engenharia de Bioprocessos e Alimentos) e Cálculo III (Bacharelado em Ciências Matemáticas e da Terra) em 2014/1 e Cálculo I (Bacharelado em Física, Física Médica, Geologia e Meteorologia) e Cálculo II (Engenharia de Bioprocessos de Alimentos) em 2014/2. Foi um prazer imenso participar minimamente da formação de todos vocês. Nesses dois anos como Professor eu tive a certeza da profissão que quero seguir.

- Da turma de Cálculo III de 2013/1 para a Engenharia Mecânica, gostaria de agradecer em particular aos alunos Caio, Henrique e Daniel, pessoas di-

vertidíssimas e muito interessadas em aprender, e eventualmente com alguns parafusos a menos na cabeça. Devíamos ter tomado cerveja mais vezes!

- Agradeço em particular aos alunos de Cálculo I e II durante o ano de 2014, que cursavam Engenharia de Alimentos e Bioprocessos. O abaixo-assinado que vocês fizeram em unanimidade em 2014/1 para que eu desse Cálculo II em 2014/2 foi muito marcante para mim. Vocês fizeram valer a pena todas as noites mal dormidas e os atrasos no meu Doutorado para preparar aulas, listas de exercícios, corrigir provas, etc. Fui recompensado com o carinho e amizade de vocês, e isso não tem preço. Jamais vou me esquecer da comemoração feita para mim no CAEng, após o término das aulas em 2014/2!

- Agradeço mais em particular ainda aos alunos dessa turma que me tornei mais próximo: Lucas (ou seria Robson?) Silva, Mateus Marinho, Rafael Ratier, Rayssa Guimarães, Kazumy Hamada e Beatriz Guerra (Bia), pela amizade que criamos.

- Devo destacar em especial meu agradecimento à Bia, que se tornou uma grande amiga ao longo desses anos, além de ter me apresentado pessoas interessantíssimas às quais também cabem aqui os devidos agradecimentos: Gabriel Raposo, Júlia Furtado, Thai, Jão, o divertidíssimo Mina, e vários outros.

- Agradeço também ao Pedro Foster, colega que conheci tocando violão com uma galera do IF e que hoje em dia integra, junto comigo e com a Bia, uma confraria cervejeira muito divertida!

- Agradeço à minha companheira nesses últimos dois anos Patrícia (*aka* Parocia) Andrade, por todo o carinho, amizade, companheirismo, gatices, retardos, cervejinhas, coquinhas, vídeos do Vsauce no almoço, e tudo de bom que passamos juntos. Você mora no meu coração. Pra sempre.

- Agradeço ao meu ex-aluno de aulas particulares, depois aluno "oficial" de Cálculo III e agora grande amigo André Guimarães. Foi muito legal acompanhar o seu crescimento ao longo desses anos, e você vai longe, rapaz!

- Agradeço ao meu também ex-aluno Renan Rios, pela sua crescente motivação e pela amizade cervejeira que criamos. Obrigado novamente por ter se desbancado da Taquara para Botafogo para ir na comemoração da defesa dessa tese!

- Agradeço também aos alunos de Graduação em Estatística, Ciências Atuariais e Ciência da Computação da UERJ que cursaram matérias comigo nos

períodos de 2015/2 e 2016/1, e peço desculpas pelas aulas eventualmente preparadas às pressas!

- Agradeço à banca do meu Exame de Qualificação (composta pelos professores Paulo Esquef, Amit Bhaya e meus orientadores) pelas diversas dicas para a melhoria do meu texto, em particular sobre alguns vacilos teóricos que lá estavam presentes!

- Agradeço também aos Professores Ralph Silva, Hélio Migon e Thais Fonseca (DME/IM-UFRJ) e meu colega Carlos Tadeu, na época mestrando em Estatística no DME/IM-UFRJ, por diversas sugestões do ponto de vista estatístico para esse trabalho, bem como algumas possíveis continuações.

- Agradeço aos Professores de Alemão que tive durante o período que estudei tal idioma, em particular o Professor Belino Reis.

- Apesar de jamais ser lido por ele, devo agradecer ao Thomas Bayes[1] pelo seu teorema, que permitiu que essa tese (e toda uma área da Estatística) exista. Na verdade, pelo Princípio de Arnol'd[2] possivelmente o teorema de Bayes não foi descoberto por ele, e mais detalhes podem ser vistos em [2].

- Outro agradecimento que jamais será lido pelo homenageado é direcionado a Johann Sebastian Bach[3], sem a menor sombra de dúvida o meu compositor favorito. Tamanha é minha admiração por ele que decidi homenageá-lo colocando seu nome no título da tese, fazendo um trocadilho divertido. Ironicamente, nenhum dos testes com sinais reais nesse trabalho foi realizado com uma obra sua! Porém, diversos dos algoritmos aqui apresentados foram escritos ao som de sua obra. Curiosamente, o prédio no qual eu moro agora se chama "Edifício Johan Sebastian Bach" (sim, escreveram o primeiro nome dele errado...).

- Continuando com os agradecimentos jamais lidos pelos homenageados, agradeço imensamente aos meus melhores amigos quase-imaginários que me acompanham diariamente: Robert de Visèe, Marin Marais, Agustín Barrios, Sylvius Leopold Weiß, Fernando Sor, Gaspar Sanz, Luyz de Narváez, Isaac Albéniz, Joaquin Rodrigo, Paulinho Nogueira, Alonso Mudarra, Giovanni Girolamo Kapsberger, e seus intérpretes Fábio Zanon, Julian Bream, David

[1] Thomas Bayes, London, c. 1701 – Tunbridge Wells, April 7, 1761.

[2] Vladimir Igorevich Arnol'd, Odessa, 12 de Junho de 1937 – Paris, 3 de Junho de 2010. O princípio tem duas partes: 1) Se algo leva o nome de uma pessoa, então tal nome não é o nome de quem descobriu isto; 2) O Princípio de Arnol'd se aplica a si mesmo. É creditado a Arnol'd por aparecer em um trabalho controverso sobre ensino de Matemática Pura, *"On teaching mathematics"*.

[3] Johann Sebastian Bach, Eisenach, 31 de Março de 1685 – Leipzig, 28 de Julho de 1750.

Russel, Andrés Segóvia, Hopkinson Smith, Paul O'Dette, Rolf Lislevand, José Miguel Moreno, Lutz Kirchoff, Göran Söllscher, Jakob Lindberd, Hespèrion XX (atualmente Hespèrion XXI), Jordi Savall, Montserrat Figueras, Hille Perl, Lee Santana, João Carlos Martins, Viktoria Mullova, Mstislav Rostorpovich, Jacqueline du Pré, Gergely Sárközy, Nigel North, Andrew Lawrence-King, Xavier Díaz-Latorre, Pedro Estevan, Paco de Lucia, Sergio e Eduardo Abreu, e muitos, muitos outros que escuto diariamente.

- Devo acrescentar à lista acima algumas bandas do meu novo vício musical, rock progressivo: Camel, Gentle Giant, Liquid Tension Experiment, Mike Oldfield, Symphony X, Bacamarte, Banco Del Mutuo Soccorso, Blind Faith, Crimson Jazz Trio, Emerson, Lake & Palmer, Focus, Frank Zappa, Gryphon, Jethro Tull, King Crimson, Pink Floyd (em particular por curar minhas febres, ao escutar "The Dark Side of the Moon"), Premiata Forneria Marconi, Quaterna Requiem, Renaissance, Rick Wakeman, Snarky Puppy, The Moody Blues, Triumvirat, Van der Graaf Generator, Yes, e outros...

- Agradeço às minhas gatas Mila e Ada, pelos passeios no teclado, cochilos nos meus rascunhos, sumiços misteriosos de objetos, arranhões nas pernas, e outras gatices. E agradeço também obviamente ao querido Anjo, *in memoriam*, pela companhia e carinho no breve tempo em que viveu lá em casa.

- Agradeço a meus amigos Renato e Krystal, à sua mais recente aquisição Nina, e às suas respectivas famílias por toda a amizade e carinho. Apesar de não termos nos visto com tanta regularidade ultimamente, vocês moram no meu coração!

- Agradeço aos amigos de loucura, Antônio e Adrielly (*aka* Antonielly) pelos sushis de queijo canastra, idas ao Mondial de la Bière, cafés espressos preparados em Júpiter, lembranças de certos passados obscuros, "não nãos", cânticos sucessivos de "Parabéns à você" e vergonhas alheias em certos bares da Tijuca. Infelizmente não posso agradecer ao *alter ego* Breno pois ainda não o conheci. Estão me devendo essa.

- Agradeço ao Fefo pelos feijões, pizzas e coincidências inusitadas em Coelho Neto, e obviamente os abraços e massagens destruidoras!

- Agradeço aos funcionários e colegas do SMT, em particular àqueles que me tornei mais próximo: Isabela, Luís Felipe, Maurício, Igor, Renam, Wallace, Iker e Andreas.

- Em particular agradeço ao Luís Felipe, que me sugeriu fazer o curso de Processamento de Imagens com o Professor Eduardo Silva para conhecer mais a área

de Processamento de Sinais, enquanto eu ainda estava no Mestrado. Graças a isso resolvi fazer Doutorado nessa área, e ironicamente o co-orientei em um projeto final de Graduação!

- Agradeço aos meus colegas Flávio, Lisandro e Michel do PROSAICO na UERJ, que abriram as portas do laboratório para me acolher quando de lá me tornei Professor.

- Agradeço à Edinalva, a senhora responsável pela limpeza das dependências do SMT, por manter um ambiente impecavelmente limpo!

- Mais geralmente, agradeço a todos os funcionários da limpeza tanto da UFRJ quanto da UERJ, que mesmo com todas as agruras e vicissitudes dos últimos tempos, zelam por um bom ambiente para todos.

- Como de costume, agradeço ao meu irmão de consideração Cláudio e sua família, por terem me acolhido em um momento que muito precisei e pelas infinitas conversas sobre Matemática na madrugada (mesmo tendo aula no dia seguinte às 8:00). Sem a sua companhia minha formação (tanto matemática quanto pessoal) certamente não seria a mesma.

- Agradeço ao Filipe, integrante do *Trio Ternura* comigo e com o Cláudio na minha época de Graduação e Mestrado pelas diversas loucuras!

- Agradeço à minha tia Cida por ter me ajudado a ir no EUSIPCO'2015. Sem sua ajuda essa viagem não seria possível naquela época.

- Mesmo tendo pouco contato nesses últimos quatro anos, agradeço a meus colegas da ABC-116 durante meu período de Graduação e Mestrado. Foi uma época muito divertida da minha vida!

- Agradeço ao Lucas e à Renata, meus amigos mais próximos da minha turma de Graduação, em particular por todos os *róiles* que usamos para programar, erros na linha -108 do código em certos *softwares* de procedência altamente duvidosa e doses de Sigmatil que tivemos que tomar nos cursos de Probabilidade.

- Agradeço ao Bruno Gusmão, meu instrutor de Krav Maga, e a todos os meus colegas de lá, pelos bons momentos que passamos juntos treinando. Que eles aconteçam por muitos e muitos anos! *Kida.*

- Agradeço a certos servidores de Internet itinerantes porém com sede sempre em países exóticos como Niue, Samoa Ocidental, Ilhas Cocos, Belize e Rússia. Sua importância para todos nós dispensa comentários.

- Agradeço a Jimmy Wales e Larry Sanger por terem idealizado e criado a fantástica Wikipedia, companheira de estudos de todos nós diariamente.

- Agradeço a Donald Knuth e Leslie Lamport por terem criado o TEX e o LATEX, respectivamente, que com suas fontes maravilhosas permite diagramar magistralmente documentos científicos. Sem isso, teríamos que usar editores de texto que dispensam palavras...

- Agradeço ao Michael Stevens, por ter criado o fantástico canal do YouTube, Vsauce, crucial para pessoas curiosas para com o mundo.

- Agradeço a Adam D'Angelo e Charlie Cheever por terem criado o Quora, uma sensacional rede social de perguntas e respostas.

- Agradeço a Vincent Connare, por ter criado a fonte Comic Sans MS, que me proporciona momentos divertidíssimos de caçada por seus usos inapropriados pela cidade.

- Agradeço aos mestres-cervejeiros de todo o mundo (em particular os belgas), por produzirem essa maravilhosa bebida, que tento modestamente reproduzir de modo caseiro, eventualmente com algum sucesso.

- Voltando aos agradecimentos impossíveis de serem lidos pelos homenageados, gostaria de agradecer às minhas maiores inspirações científicas: Kolmogorov[4], Feynman[5] e von Neumann[6]. Sem vocês, certamente o mundo hoje não seria o mesmo.

- Agradeço à Professora Alexandra Schmidt por ter me ensinado diversas nuances do pensamento estatístico, mesmo nas poucas porções de seus cursos que pude assistir como ouvinte. Tais ensinamentos foram muito importantes para mim ao longo do Doutorado.

- Agradeço ao Reinaldo e à Camilla, meus colegas do IF, pelas conversas sobre Ciência que tivemos. Infelizmente foram poucas, mas espero que ao longo dos próximos anos elas se tornem mais frequentes.

- Agradeço aos Professores Fábio Ramos, Heudson Mirandola, Marcelo Tavares, Bernardo Costa e Amit Bhaya, pela amizade e apoio nesses últimos anos.

- Agradeço aos Professores Flávio Dickstein e Paulo Goldfeld pela confiança depositada em mim ao me convidarem para participar do projeto junto com a Petrobrás.

[4] Andrey Nikolaevich Kolmogorov, Tambov, April 25, 1903 – Moscow, October 20, 1987.
[5] Richard Phillips Feynman, New York, May 11, 1918 – Los Angeles, February 15, 1988.
[6] John von Neumann, Budapest, December 28, 1903 – Washington, D.C., February 8, 1957.

- Agradeço ao Betão, proprietário de uma cantina no DAEQ (Diretório Acadêmico da Escola de Química), que serve o melhor croissant de chocolate da UFRJ, além de ser o único lugar do CT que vende uma cervejinha pra acompanhar o almoço do proletariado.

- Agradeço aos criadores do CoppeTeX (`http://coppetex.sourceforge.net/`), por terem criado essa classe do LaTeX que em tanto facilita a nossa vida ao escrever a tese!

Peço infinitas desculpas se alguém ficou de fora dessa lista. Tomei o maior cuidado do mundo para que isso não acontecesse, mas infelizmente vacilos acontecem.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

BAYES ENCONTRA BACH: APLICAÇÕES DE ESTATÍSTICA BAYESIANA A RESTAURAÇÃO DE ÁUDIO

Hugo Tremonte de Carvalho

Janeiro/2017

Orientadores: Luiz Wagner Pereira Biscainho
Flávio Rainho Ávila

Programa: Engenharia Elétrica

Distorções não-lineares podem aparecer em sinais de áudio desde o momento da sua gravação até a posterior reprodução: equipamentos precários ou operados de maneira indevida, mídias fisicamente degradadas e baixa qualidade dos aparelhos de reprodução são somente alguns exemplos onde não-linearidades podem aparecer de modo natural.

Outro defeito bastante comum em gravações antigas são os pulsos longos, em geral causados pela reprodução de discos com arranhões muito profundos ou fitas magnéticas severamente degradadas. Tais defeitos são caracterizados por uma descontinuidade inicial na forma de onda, seguida de um transitório de baixa frequência e longa duração.

Em ambos os casos, artefatos auditivos podem ser criados, causando assim uma experiência ruim para o ouvinte. É importante então desenvolver técnicas para mitigar tais efeitos, tendo como base somente uma versão do sinal degradado, de modo a recuperar o sinal original não degradado.

Nessa tese são apresentadas técnicas para lidar com esses dois problemas: o problema de restaurar gravações corrompidas com distorções não-lineares é abordado em um contexto bayesiano, considerando tanto modelos autorregressivos quanto de esparsidade no domínio da DCT para o sinal original, bem como por uma solução determinística também usando esparsidade; para a supressão de pulsos longos, uma abordagem paramétrica é revisitada, junto com o acréscimo de um eficiente procedimento de inicialização, sendo também apresentada uma abordagem não-paramétrica usando processos gaussianos.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

BAYES MEETS BACH: APPLICATIONS OF BAYESIAN STATISTICS TO AUDIO RESTORATION

Hugo Tremonte de Carvalho

January/2017

Advisors: Luiz Wagner Pereira Biscainho
Flávio Rainho Ávila

Department: Electrical Engineering

Memoryless nonlinear distortion can be present in audio signals, from recording to reproduction: bad quality or amateurishly operated equipments, physically degraded media and low quality reproducing devices are some examples where nonlinearities can naturally appear.

Another quite common defect in old recordings are the long pulses, caused in general by the reproduction of disks with deep scratches or severely degraded magnetic tapes. Such defects are characterized by an initial discontinuity in the waveform, followed by a low-frequency transient of long duration.

In both cases audible artifacts can be created, causing an unpleasant experience to the listener. It is then important to develop techniques to mitigate such defects, having at hand only the degraded signal, in a way to recover the original signal.

In this thesis, techniques to deal with both problems are presented: the restoration of nonlinearly degraded recordings is tackled in a Bayesian context, considering both autoregressive models and sparsity in the DCT domain for the original signal, as well as through a deterministic solution also based on sparsity; for the suppression of long pulses, a parametric approach is revisited with the addition of an efficient initialization procedure, and a nonparametric modeling via Gaussian process is also presented.

Contents

List of Figures

List of Tables

Part I

FOUNDATIONS

Chapter 1

Introduction

"Abandon all hope, ye who enter here."

– Dante Alighieri[1], *Divine Comedy*

The human being has always been concerned with the preservation of important moments for posterior admiration: the prehistoric man carved figures in the rocks, mainly as part of its rituals; the creation of the written language allowed the recording of informations in a less pictorial way than primitive drawings; and with the development of better tools it was possible to manufacture canvas, inks and brushes, and with some technical effort the paintings became more and more precise and realistic. All these aspects are about the recording of visual information, and the recording of auditory information needed to wait until the technological advances of recent times.

One of the earliest forms of "recording" audio information, at least in a graphical way, dates back to about two millennia BCE[2], where tables with cuneiform symbols represented the melodic line of some songs. More sophisticated notations were developed along time, in particular the modern staff notation, whose rudimentary ideas date back to the 11th. Century, with Guido d'Arezzo[3]. However, the recording of audio information in written form possesses a severe and obvious drawback: in order to be heard, it must be played by a person or group of people! Moreover, there must be another person able to transcribe the audio information to a written format, not an easy task.

The earliest known device to record an audio information without being transcribed by a human was the *phonautograph*, created and patented by Édouard-Léon Scott de Martinville[4] in 1857. Built analogously to some structures of the human ear, this device was able to transcribe sound waves in a pictorial information, its

[1]Durante degli Alighieri, Florence, c. 1265 – Ravenna, September 1321.
[2]Abbreviation to "Before Common Era" or "Before Current Era".
[3]Guido d'Arezzo, c. 991 – after 1033.
[4]Édouard-Léon Scott de Martinville, April 25, 1817 – April 26, 1879.

corresponding waveform. It was intended not for playing back the sound, but only to create a visual representation of the audible information. However, in 2008 researchers were able to play these waveforms, recorded more than 150 years ago [3].

Some years later, in 1877, Thomas Edison[5] invented and patented the *phonograph*, the first device able to record and reproduce the recorded sound. Like the phonautograph, it used a conical horn to conduct the air pressure to a membrane, which moved an attached needle back and forth, engraving the waveform in a metal, later replaced by a wax, cylinder. The playback of the sound was done in a dual way, by using the needle to trace the groove, causing the membrane to vibrate and emit sound. Some years later, the standard recording media became a disk instead of a cylinder.

Fast-forwarding some years, around 1950 recording in magnetic tapes became very popular, since it allowed to record continuously for a longer duration and with much higher fidelity than before. Another advantage of using magnetic tapes was allowing the easy manipulation of the recorded information, by physically combining audio excerpts recorded in more than one tape.

And finally, around 1975 began the digital era of recorded audio, the most rapid and far-reaching series of changes in the history of recorded audio until now. Digital recording and reproducing media, like the DAT (Digital Audio Tape) and the CD (Compact Disc), became the standard, both for the industry and the consumer.

In all these cases, the recorded information is sensitive to the preservation state of the physical media; and for the oldest recordings it is almost certain that the media is degraded in some way, leading to audible artifacts that impair the quality of perceived information. For example, a cylinder or disk could be scratched (or even broken), causing deviations along the needle's path and disturbances in the arm response, creating then sounds like *clicks* or *thumps*, technically known as *impulsive noise* and *long pulse*, respectively; the puncture on the disk might not be well centralized, and the disk might be bent if subject to high temperatures, creating then variations of speed during reproduction; the process of recording and reproducing information in a magnetic tape is inherently nonlinear, due to the physical process involved in the production of a magnetic flux from an electrical current, causing nonlinear modifications to the recorded audio.

It is then important to be able to "undo" some of these impairments, returning back to the original recorded information, and this is the main focus of this thesis. After the emergence and popularization of the digital computer, restoration procedures on acoustical information from degraded media became more efficient and easy to perform, despite not being straightforwardly designed.

The defects in an audio recording can be broadly divided in *localized* or *dis-*

[5]Thomas Alva Edison, Milan, February 11, 1847 – West Orange, October 18, 1931.

tributed. A localized degradation is one that affects only short sections of the signal, like the *clicks* heard in a scratched disk. On the other hand, distributed defects, like background noise, nonlinear distortions in magnetic tapes, and variations of speed, for example, impact the entire signal or a large portion of it. It is not easier to restore one or another category of degradations: each one possesses its particularities that must be exploited in order to create an effective restoration procedure.

In this work, we propose solutions to some of these problems. The restoration procedure is performed in a digital computer, with a digitized version of the degraded signal and via statistical methods for their description, its respective degradations and its estimation. Each of these theoretical frameworks and procedures are described in detail along the text. This thesis is structured in the following way:

- This first part presents very briefly the theoretical foundations that are needed to tackle the problem: Chapter 2 recalls some basic tools and definitions from Probability and Statistics, followed by the Bayesian computational methods used in most of the proposed algorithms in Chapter 3. Finally, in Chapter 4 the autoregressive model for audio signals is presented and discussed in detail.

- The second part presents solutions to the restoration of audio signals with nonlinear distortions when the original signal is modeled via the AR model. Chapter 5 is a brief introduction to nonlinearities in audio signals, and in Chapter 6 some possibilities to model nonlinear phenomena are presented. Solutions to the restoration of audio signals degraded by nonlinear distortion without and with memory are presented in Chapters 7 and 8, respectively, and its results and related future works are described in Chapter 9.

- In the third part, the problem of nonlinearly distorted audio signals is tackled from a more recent perspective: the original undistorted signal is modeled as approximately sparse in the DCT domain. In Chapter 10, an adaptation of [4] to the case of audio signals is presented, where the sparsity of the signal is deterministically modeled. On the other hand, Chapter 11 models the sparsity in a Bayesian context via Laplace priors in the DCT domain.

- In the fourth part, the problem of degradation by low frequency decaying pulses is treated. In Chapter 12, a parametric model for the pulse is presented together with an estimation procedure for their parameters, while in Chapter 13 the pulse is modeled in a non-parametric way via a Gaussian Process, and also an efficient initialization procedure for the variables of the initial discontinuity is presented.

- Finally, the fifth and last part of the text discusses the results obtained and indicates future works in Chapter 14.

Publications associated with this thesis are [5–7].

Whenever possible, historical information is provided, and essential concepts are discussed in an intuitive way before presenting their more technical aspects. I hope that its improved readability justifies the extended length of this text.

Chapter 2

Probability and Statistics

"God does not play dice with the universe."

– Albert Einstein[1]

2.1 Probability: quantifying uncertainty since antiquity

The human being has always been concerned with uncertainty. Since the beginning of the civilizations, quantifying and (trying to) reduce the uncertainty of natural phenomena is important: the first farmers in the Fertile Crescent were concerned whether it would rain or not, and they prayed and did offerings to the gods in order to reduce the uncertainty in the climate; there are historical reports of gambling games in ancient times, from Egypt to Greece, in which the player could use dishonest artifacts to reduce the uncertainty on their odds of winning; in the 9th. Century Al-Kindi[2] used rudimentary techniques of Statistics to create the first known code breaking algorithm, thus reducing the uncertainty about the information contained in an encrypted message [8, 9].

Fast-forwarding to more recent times, Probability and Statistics theory are widely employed in our world, in almost every scientific field of knowledge. For example, it is unimaginable to publish a scientific paper reporting the result of some experiment without a statistical analysis of the data, leading to a "proof" of the correctness (or not) of the proposed claim; investment funds employ very sophisticated techniques of Probability theory with the goal of predicting the behavior of the financial market; Netflix uses statistical algorithms to predict which films you will like to watch, based on information about films already watched and rated.

[1] Albert Einstein, Ulm, March 14, 1879 – Princeton, April 18, 1955.

[2] Abu Yūsuf Ya'qūb ibn 'Isḥāq aṣ-Ṣabbāḥ al-Kindī, Basra, c. 801 – Baghdad, c. 873.

Unfortunately not every person is aware of Probability and Statistics theory, and sometimes many are fooled by banks, lotteries and casinos.

The main usage of Probability and Statistics in this thesis will not have any of the motivations above; they will serve as an abstract and very powerful tool to describe audio signals and other quantities related to them. This chapter is not an introduction to Probability and Statistics, but only a collection of ideas and methods that will be used during this text. As an introduction to the subject we strongly recommend the book [10].

2.2 The relationship between Probability and Statistics

In the last section we talked about Probability and Statistics not in a symmetric way. In fact, these fields are very different, despite this fact being not so clear. In order to explain this, take a look on the diagram below, strongly inspired by [11]:

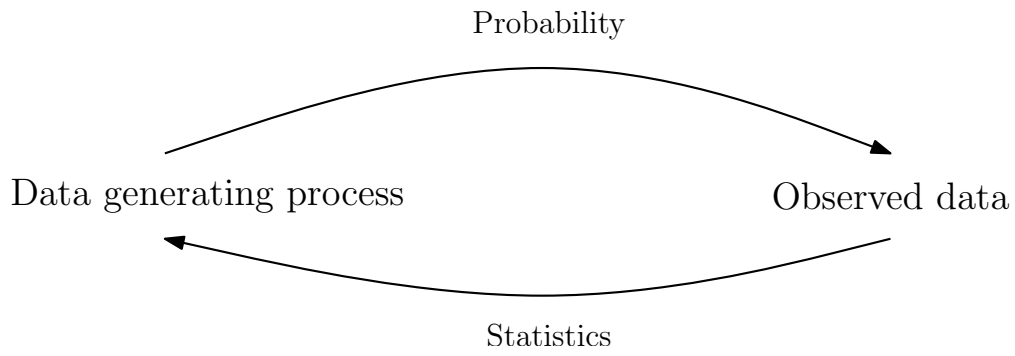

Figure 2.1: Probability and Statistics.

Probability theory is concerned with obtaining information from the data being generated by some well-known data generating process. For example, one could be interested in computing the expected number of heads until the first tail appears in a tossing coin game, knowing beforehand that the coin is fair and having a model for the process being studied.

On the other hand, Statistics theory does the opposite, that is, it tries to infer informations about the data generating process, analyzing the available data. Following the example in the last paragraph, a gambler could be interested in discovering whether the coin is fair, based on a sequence of trials of the coin tossing game.

This difference should always be clear when dealing with probabilistic and statistical concepts, in order to guide our intuition. In this work we will be mainly interested in creating a probabilistic model for audio signals and their respective

defects, depending on some parameters, and estimating these quantities from the data available in order to restore the degraded signal. So, in the figure presented above, we will be mainly interested in the arrow from right to left.

2.3 Interpretations of Probability

Probability theory is quite an old subject in Mathematics, being studied regularly since Renaissance by famous mathematicians and philosophers like Cardano[3], de Moivre[4], Pascal[5], Fermat[6], the Bernoulli family[7], and many others [9, 12]. Since the subject involves much more than technical effort in order to understand its basis due to its philosophical content, it is then natural that some discordances about the interpretations of probability have emerged along time. For example, how does one interpret the following sentences?

- The probability of obtaining the number 2 when throwing a balanced dice is 1/6, approximately 16.67%.
- The probability of raining tomorrow in Rio de Janeiro is 32%.

Intuitively, the first one is quite simple, since we can imagine that if one throws the dice several times, in approximately 1/6 of the trials the number 2 will be obtained[8]. But what about the second sentence? If one tries to interpret it in the same fashion as the first one, something very weird will happen! There is *no* way of repeating the "tomorrow" several times and computing the proportion of "tomorrows" in which will rain in Rio de Janeiro, unless we accept the hypothesis of multiple parallel universes in Physics!

In fact, these two sentences lead to very different interpretations of probability: the *frequentist* and the *subjective* ones. The frequentist interpretation is exactly what we proposed in the beginning of the last paragraph: the probability of an event occurring is its relative frequency when the number of trials is big enough. On the other hand, the subjective interpretation is related to the degree of plausibility one associates with the occurrence of some event. Note that the subjective interpretation is much more reasonable than the frequentist one in the second sentence: based on methods for weather forecasting, the degree of belief of someone in the hypothesis

[3]Gerolamo Cardano, Pavia, September 24, 1501 – Rome, September 21 1576.

[4]Abraham de Moivre, Vitry-le-François, May 26, 1667 – London, 27 November, 1754.

[5]Blaise Pascal, Clermont-Ferrand, June 19, 1623 – Paris, August 19, 1662.

[6]Pierre de Fermat, Beaumont-de-Lomagne, August 17, 1601 or 1607 – January 12, 1665.

[7]A family of merchants and scholars from Switzerland.

[8]In fact this is not so simple as stated. What do we mean by "several times" and "approximately"? Our mind can accept this inaccuracy, but rigorously there is something missing. Nowadays, with the modern foundations of Probability theory, this can be viewed as a family of theorems, the Laws of Large Numbers.

of raining tomorrow in Rio de Janeiro might be not high enough to make him worry about a flood, but sufficiently high to make him carry an umbrella inside his backpack.

There are other possible interpretations of Probability [13], and this is always a cause of huge controversies in the scientific community. Although very interesting, this philosophical aspect of the theory is not the main focus of this work.

2.4 Probability spaces and random variables

The axiomatization of Probability theory we use nowadays is due to Kolmogorov, and was first published in 1933 (English translation can be found in [14]). Despite the previous discussion, the axioms postulated by Kolmogorov and stated below are independent of any interpretation of Probability. The main ingredients to do Probability are[9]:

- A set Ω, called the *sample space*;
- A σ-algebra $\mathcal{F}$ defined over Ω, called the *set of events*;
- A function $\mathbb{P} : \mathcal{F} \to [0, 1]$, the *probability measure*, satisfying $\mathbb{P}(\Omega) = 1$ and $\mathbb{P}(\cup_{n=1}^{\infty} E_n) = \sum_{n=1}^{\infty} \mathbb{P}(E_n)$, if the sequence $(E_n)_{n\in\mathbb{N}} \subset \mathcal{F}$ is disjoint.

The triple $(\Omega, \mathcal{F}, \mathbb{P})$ is called a *probability space*. Intuitively, the set Ω represents the possible outcomes of an experiment, and $\mathcal{F}$ consists of the subsets of Ω we judge "reasonable" to measure. Finally, the function $\mathbb{P}(.)$ assigns a number from 0 to 1 to any of these "reasonable" sets, representing its probability. The function $\mathbb{P}(.)$ and the σ-algebra must satisfy some compatibility conditions that are beyond the scope of this text.

One of the main advantages of using this framework is that it allows the use of the very powerful tools of Calculus. This is done via the concept of a *random variable*, which means nothing more than a function $X : \Omega \to \mathbb{R}$ satisfying the condition $X^{-1}(-\infty, a) \in \mathcal{F}$, for all $a \in \mathbb{R}$. Despite its apparent obscurity, a random variable is only a way of summarizing numerically the possible results of an experiment, ignoring all the detailed information contained in Ω. The above condition that $X(.)$ must satisfy is a very technical one and is called *measurability*. It guarantees essentially that the random variable does not require any information that is not available in the σ-algebra $\mathcal{F}$[10].

[9] For a precise definition of these entities, see [15, 16].

[10] The name "random variable" is a very misleading one, since there is nothing more deterministic than a function that assigns one number to another object in a well defined way, and since it is a function it is not a variable anymore! Anyway, this is the best known way to express uncertainty in terms of real numbers, and unfortunately it received this name, which is widely used in the literature.

Associated with each random variable there is its respective *distribution*, being essentially a function representing how the probability is spread among the image of the function $X(.)$: since we are assigning a real number to each possible outcome in Ω, it is reasonable to assign probabilities to these real numbers as well. The two main types of random variables are the *discrete* and *continuous* ones. The image of a discrete random variable is a discrete subset of $\mathbb{R}$, denoted by $\{x_1, x_2, \dots\}$, and for each x_i there is a number p_i associated, representing its probability of occurrence, that is, $p_i = \mathbb{P}(X = x_i) := \mathbb{P}(X^{-1}(x_i))$. The sequence $p_1, p_2, \dots$ is called the *probability mass function* (PMF) of X. Continuous random variables are characterized by the existence of a *probability density function* (PDF), an integrable positive function $f : \mathbb{R} \to \mathbb{R}$ satisfying $\mathbb{P}(X \in (a,b)) := \mathbb{P}(X^{-1}(a,b)) = \int_a^b f(x)\, dx$. And finally, independently of being discrete or continuous, we can define the *cumulative distribution function* (CDF) by $F(x) = \mathbb{P}(X \in (-\infty, x))$, for all $x \in \mathbb{R}$.

Random variables can also be vector-valued, that is, a measurable function $\mathbf{X} : \Omega \to \mathbb{R}^n$, for some $n \in \mathbb{N}$. The random vector $\mathbf{X}$ is composed of n random variables, denoted by $X_1, \dots, X_n$, and can also be discrete or continuous. We will be only interested on continuous random vectors. Analogously to the univariate case, this situation is characterized by the existence of an integrable function $f : \mathbb{R}^n \to \mathbb{R}$ such that $\mathbb{P}(X \in D) := \mathbb{P}(X^{-1}(D)) = \int_D f(\mathbf{x})\, d\mathbf{x}$, where $D \subset \mathbb{R}^n$. This function is also called the probability density function.

We note that independently of the random variable being discrete or continuous, we will always refer to its PMF or the PDF only as its distribution. Since in this work we will deal essentially with continuous distributions, this should cause no confusion. Some results will be then stated only for continuous distributions, but a discrete counterpart always does exist. There are some well-known family of distributions, and for a non-extensive list see Chapter 5 of [10].

A very important theorem we must use along the text is the *change of variables for multidimensional random variables*, since sometimes we will need to compute the PDF of a function of a random vector. We state this theorem here, without proof.

Theorem 2.4.1 (Change of variables for multidimensional random variables)**.** *Let the random vector $\mathbf{X} = (X_1, \dots X_N)$ have a continuous joint distribution for which the joint PDF is given by $f_{\mathbf{X}}$, whose support is the set $S \subset \mathbb{R}^N$. Define a new random vector $\mathbf{Y} = (Y_1, \dots Y_N)$ as:*

$$\begin{aligned} Y_1 &= r_1(X_1, \dots, X_N); \\ &\vdots \\ Y_N &= r_N(X_1, \dots, X_N). \end{aligned} \tag{2.1}$$

Assume that the functions $r_1, \ldots, r_N$ define an one-to-one and differentiable transformation of S onto the set $T \subset \mathbb{R}^N$, whose inverse is given by

$$\begin{aligned} x_1 &= s_1(\mathbf{y}); \\ &\vdots \\ x_N &= s_N(\mathbf{y}), \end{aligned} \tag{2.2}$$

where $\mathbf{y} = (y_1, \ldots, y_N) \in T$. Then the joint PDF of $\mathbf{Y}$, denoted by $f_{\mathbf{Y}}$, is given by

$$f_{\mathbf{Y}}(\mathbf{y}) = \begin{cases} f_{\mathbf{X}}(s_1(\mathbf{y}), \ldots, s_N(\mathbf{y}))|J|, & \text{for } \mathbf{y} \in T; \\ 0, & \text{otherwise}, \end{cases} \tag{2.3}$$

where J is the determinant

$$J = \det \begin{bmatrix} \dfrac{\partial s_1}{\partial y_1} & \cdots & \dfrac{\partial s_1}{\partial y_N} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial s_N}{\partial y_1} & \cdots & \dfrac{\partial s_N}{\partial y_N} \end{bmatrix}. \tag{2.4}$$

2.4.1 An example

We provide now a short example to clarify the meaning of the above defined objects. Suppose that a fair coin is tossed 10 times. Then the sample space Ω consists of the 2^{10} possible outcomes of this experiment, that is, all the sequences of length 10 consisting of head and tails. Since any subset of Ω is reasonable to be measured, meaning that at least intuitively we can assign probability to all of them, the σ-algebra $\mathcal{F}$ is the power set of Ω. Under the hypothesis of fairness of the coin, it is reasonable to assign probability $1/2^{10}$ to any element of Ω, that is, $\mathbb{P}(\omega) = 1/2^{10}$, for all $\omega \in \Omega$. It can be proven that this choice of $(\Omega, \mathcal{F}, \mathbb{P})$ is in fact a probability space.

Now imagine that one is interested not in all the sequences of heads and tails obtained in a trial of the experiment, but only in the number of heads obtained. This clearly defines a function $X : \Omega \to \mathbb{R}$, that can be proven to satisfy the conditions to be a random variable. Since the number of heads in a trial can be only $0, 1, \ldots, 10$, this is a discrete random variable, and the associated probabilities p_i are easily computable.

Sometimes, mainly when considering continuous random variables, the sample space is some Euclidean space $\mathbb{R}^n$ itself, and the considered random variable is the identity function. In this cases, we can "forget" about the sample space and probability measure, since all the required informations are encoded in the PDF.

2.5 Statistical inference: retrieving information from data

When we assign a probability model to some real world phenomena, we are almost always ignoring the underlying sample space, σ-algebra and probability measure, but simply associating a distribution to them. Every useful family of distribution consists of a PMF or a PDF and its respective *parameters*[11], denoted in both cases by $f(\mathbf{x}|\boldsymbol{\theta})$, where $\mathbf{x}$ is the argument of the function and $\boldsymbol{\theta}$ is a vector containing the parameters of the distribution. The main problem in statistical inference is to estimate $\boldsymbol{\theta}$ from observations of the corresponding distribution. There are several tools for estimating the parameters of a probability distribution [17], and now the distinction between the frequentist and subjective views of Probability arises again.

From the frequentist point of view, the parameters are an unknown and fixed quantity, and one of the most popular ways of estimating $\boldsymbol{\theta}$ in this context is the *maximum likelihood* method, briefly described here.

The *likelihood function* is defined as

$$\ell(\boldsymbol{\theta}; \mathbf{x}) = f(\mathbf{x}|\boldsymbol{\theta}), \tag{2.5}$$

considered as a function of $\boldsymbol{\theta}$ when the observed data are fixed and equal to $\mathbf{x}$. The *maximum likelihood estimator* of $\boldsymbol{\theta}$ is then the value that maximizes the function $\ell(.; \mathbf{x})$, that is,

$$\boldsymbol{\theta}^{\mathrm{ML}} = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}}\, \ell(\boldsymbol{\theta}; \mathbf{x}), \tag{2.6}$$

where the set Θ is the set of possible values of $\boldsymbol{\theta}$. This estimate is interpreted as the parameters value for which the observed data are the most probable ones. There are several ways of computing this maximum (or these maxima) [18], but sometimes the likelihood function is very complicated to maximize, even via sophisticated numerical methods, and this will be exactly our main problem in using it in this work. We must resort to more advanced techniques, presented in the next section.

2.6 Parameters as random quantities: Bayes' theorem, or "inverse probability"

When doing some calculations in the eighteenth Century, the Presbyterian minister Thomas Bayes certainly did not expect to be as famous as he is today. In that time,

[11] From the Ancient Greek $\pi\alpha\rho\alpha$, "para", meaning "beside, subsidiary, auxiliary", and $\mu\epsilon\tau\rho o\nu$, "metron", meaning "measure". Therefore, the word "parameter" can be understood as a subsidiary- or quasi-measurement.

he discovered a particular case of the now called *Bayes' Theorem*, independently rediscovered and generalized some years later by Laplace[12] [9]. This result remained controversial in the mathematical community for a long time, and was also known as *inverse probability*. For a history of Bayesian Statistics, see [19]. The importance of Bayesian methods in this work is the shift of paradigm it provides to the problem of statistical inference.

Recall that in the previous section the parameter $\boldsymbol{\theta}$ was considered a fixed and unknown vector that we want to estimate from the observed data. Using Bayes' Theorem, it is possible to assign probabilities to $\boldsymbol{\theta}$ as well:

$$f(\boldsymbol{\theta}|\mathbf{x}) = \frac{f(\mathbf{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})}{f(\mathbf{x})}. \tag{2.7}$$

But the equation above says that we must pay some price in order to do this assignment: we must create a *prior distribution* for $\boldsymbol{\theta}$, a function $f(\boldsymbol{\theta})$ representing our initial knowledge about the parameters. For example, if one knows that its components are all positive and that they are all close to zero, this information can be encoded in $f(\boldsymbol{\theta})$ in order to assign probabilities directly to $\boldsymbol{\theta}$.

Note that the denominator in Equation 2.7 is a fixed number, since it depends only on the observed data. It can be viewed as a constant guaranteeing that $f(\boldsymbol{\theta}|\mathbf{x})$ integrates to one. Another form of the Bayes' Theorem that we will use in this work ignores this constant, and states that

$$f(\boldsymbol{\theta}|\mathbf{x}) \propto f(\mathbf{x}|\boldsymbol{\theta})f(\boldsymbol{\theta}). \tag{2.8}$$

The function $f(\boldsymbol{\theta}|\mathbf{x})$ is called the *posterior distribution* of $\boldsymbol{\theta}$, and can be understood as the degree of belief we have about any particular choice of $\boldsymbol{\theta}$. Several reasonable estimators can be constructed using this distribution, in particular using the notion of a *loss function*. Very briefly, given a estimator $\hat{\boldsymbol{\theta}}$ for the parameter $\boldsymbol{\theta}$, we can define a loss function, denoted by $L(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$, representing the loss of choosing $\hat{\boldsymbol{\theta}}$ when the true value of the parameter is $\boldsymbol{\theta}$. Some common loss functions are given by:

- $L_p(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_p$
- $L_\epsilon(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \begin{cases} 0, & \text{if } \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\| \leq \epsilon; \\ 1, & \text{otherwise.} \end{cases}$

The *Bayes estimator* is the estimator that minimizes the expected posterior loss, given by

$$r(\hat{\boldsymbol{\theta}}|\mathbf{x}) = \int_{-\infty}^{+\infty} L(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) f(\boldsymbol{\theta}|\mathbf{x}) \, d\boldsymbol{\theta}. \tag{2.9}$$

[12] Pierre-Simon Laplace, Beaumont-en-Auge, March 23, 1749 – Paris, March 5, 1827.

For a single real parameter and using the L_p loss function with $p = 1$, the Bayes estimator is the median of the posterior distribution, whereas with $p = 1$ it is the mean of the posterior distribution. The L_ϵ loss function with small ϵ leads to the maximum of the posterior distribution, given by:

$$\boldsymbol{\theta}^{\mathrm{MAP}} = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}}\, f(\boldsymbol{\theta}|\mathbf{x}). \tag{2.10}$$

This estimator is interpreted, in contrast to the maximum likelihood, as the most probable $\boldsymbol{\theta}$ for the observed data. For more details, see [11].

Recall that in the previous section we pointed out that maximizing the likelihood function can be a very difficult procedure, even with the aid of sophisticated numerical methods, and the posterior distribution for $\boldsymbol{\theta}$ can be even more complicated, since it is essentially the product of the likelihood with the prior distribution. The main advantage now is that $\boldsymbol{\theta}$ possesses a probability distribution, and if we are able to sample from this distribution, we are perhaps able to estimate quantities of interest from it. How to sample from probability distributions is the subject of the next chapter.

Chapter 3

Monte Carlo methods

> *"Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin. For, as has been pointed out several times, there is no such thing as a random number – there are only methods to produce random numbers, and a strict arithmetic procedure of course is not such a method."*
>
> – John von Neumann

In this chapter we briefly recall some solutions to the problem of obtaining samples from some particular probability distribution, with special emphasis on Markov Chain Monte Carlo (MCMC) methods, since they will be widely used in the rest of the text. The chapter is organized as follows: in Section 3.1 some motivations and the earlier solutions to the sampling problem are presented, and Section 3.2 introduces the MCMC methods used in this text, in particular the Metropolis-Hastings algorithm (3.2.2), the Gibbs sampler (3.2.3) and the Metropolis within Gibbs (3.2.4). We close the chapter discussing some convergence issues about these algorithms in Section 3.2.5.

3.1 Motivation and early methods for sampling

Every random experiment we perform can be regarded as a sampling procedure from some probability distribution. For example, if one tosses a fair coin and associates the number 1 to tail and 0 to head, this person is performing a sample from the Bernoulli distribution with parameter 1/2; if one spins a roulette and observes the angle of the pointer with the x axis, a sample from the uniform distribution in the interval $[0, 2\pi)$ is obtained. But imagine now a person throwing a dart on a target and computing the distance from its center: it is not obvious from which probability distribution this sample is. On the other hand, if one wants to obtain a sample from a Beta distribution with parameters 0.7 and 0.59, it is not clear which procedure

must be adopted in order to perform this task. Even if such procedure is created, if one wishes to generate a large number of samples it can be very tedious to repeat the procedure several times. Imagine, for example, tossing a coin a million times to obtain a million samples from the Bernoulli distribution with parameter 1/2: even if tossing one coin per second, the entire procedure will last for approximately 11 days and 13 hours!

The first solutions to this problem appeared around 1940 in correspondences between von Neumann and Stanislaw Ulam[1] [20], and we recall them briefly here. For an excellent historical presentation of Monte Carlo methods see [21].

3.1.1 Inverse probability transform and the accept-reject method

The problem motivating von Neumann and Ulam was of practical interest: they were working together at Los Alamos after World War II, and they needed to compute quantities like the mean distance a neutron travels during some nuclear reaction before it collides with another atomic nucleus, and how much energy the neutron loses after some collision. The physicists at Los Alamos were unable to solve this problem analytically, since it required a lot of very difficult computations. The idea of using random numerical experiments to obtain the required quantities was from Ulam, when convalescing from an illness and playing solitaire! The following quotation is a remark from 1983 by Stan Ulam [20]:

> *"The first thoughts and attempts I made to practice [the Monte Carlo Method] were suggested by a question which occurred to me in 1946 as I was convalescing from an illness and playing solitaires. The question was what are the chances that a Canfield solitaire laid out with 52 cards will come out successfully? After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than "abstract thinking" might not be to lay it out say one hundred times and simply observe and count the number of successful plays. This was already possible to envisage with the beginning of the new era of fast computers, and I immediately thought of problems of neutron diffusion and other questions of mathematical physics, and more generally how to change processes described by certain differential equations into an equivalent form interpretable as a succession of random operations. Later [in 1946], I described the idea to John von Neumann, and we began to plan actual calculations."*

[1] Stanislaw Marcin Ulam, Lemberg, April 13, 1909 – Santa Fe, May 13, 1984.

In a letter to Ulam, von Neumann proposed the inverse probability transform and the accept-reject methods as a partial solution to this problem, and also pointed out some of its problems.

The inverse probability transform is a way of generating samples from virtually any distribution from samples of the uniform distribution on the interval $[0, 1]$, as stated in the following theorem:

Theorem 3.1.1 (Inverse Probability Transform)**.** *Let X be a continuous random variable with PDF $f(x)$ and CDF $F(x)$, and let $Y = F(X)$. Then the distribution of Y is the uniform distribution on the interval* $[0, 1]$.

Its proof is very simple, and can be found in [10] or [22]. To illustrate, imagine that one is interested in generating samples from X following the exponential distribution with parameter 1. It is well known that the PDF and CDF of this distribution are given, respectively, by $f(x) = e^{-x}$ and $F(x) = 1 - e^{-x}$, for $x > 0$. The inverse of $F(.)$ is given by $F^{-1}(y) = -\ln(1 - y)$, for $y \in [0, 1]$. The theorem states that $Y = F(X)$ is uniformly distributed in the interval $[0, 1]$, and a straightforward computation shows that $F^{-1}(Y)$ is exponentially distributed with parameter 1. If $y_1, \ldots, y_n$ are samples from the uniform distribution on the interval $[0, 1]$, then $-\ln(1 - y_1), \ldots, -\ln(1 - y_n)$ are samples from the exponential distribution with parameter 1.

The other method proposed by von Neumann is the accept-reject method, briefly explained now. Suppose one wishes to generate sample from a random variable X with PDF $f(.)$, called the *target distribution*, and one is unable to sample directly from $f(.)$. But suppose that an auxiliary random variable Y is available, whose PDF $g(.)$ is easy to sample from. The basic idea of the method consists in generating samples from Y and "accept" these samples as genuine samples of X if they passes through some test. More precisely, $f(.)$ and $g(.)$ must satisfy two conditions:

i) $f(.)$ and $g(.)$ have compatible supports, that is, $g(x) > 0$ when $f(x) > 0$;

ii) There is a constant M such that $f(x)/g(x) \leq M$, for all x.

Being these two conditions satisfied, we generate a sample y from $g(.)$ and independently generate u from a uniform distribution in the interval $[0, 1]$. If

$$u \leq \frac{1}{M} \frac{f(y)}{g(y)}, \tag{3.1}$$

we accept y as a genuine sample from $f(.)$, and reject it otherwise.

It is not intuitive to see that this method in fact provides exact samples of $f(.)$, and a proof can be found in [22]. We provide an intuitive explanation based on Figure 3.1.

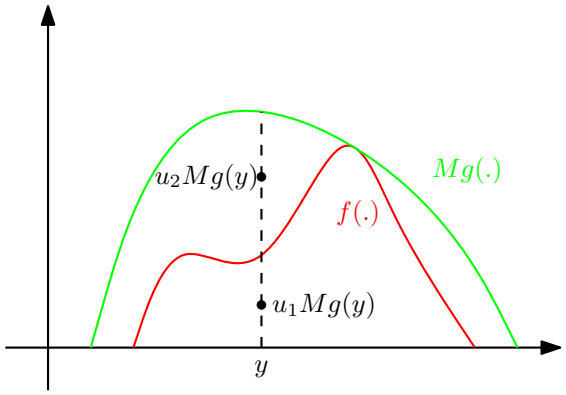

Figure 3.1: Illustration of the accept-reject method.

The condition to accept y as a genuine sample from $f(.)$ in Equation 3.1 can be rewritten as

$$uMg(y) \leq f(y), \tag{3.2}$$

that is, the product $uMg(y)$ falls under the graph of $f(.)$. In the figure above, the depicted value of y was sampled from $g(.)$, and two samples from the uniform distribution on the interval $[0, 1]$, denoted by u_1 and u_2, are shown. In the first case, the product $u_1Mg(y)$ is under the graph of $f(.)$, so this value of y is accepted as a sample from $f(.)$. In the second case, the product $u_2Mg(y)$ is above the graph of $f(.)$ and the sample y is rejected as a genuine sample from $f(.)$. It is now clear the requirement of existing some M satisfying $f(x)/g(x) \leq M$, for all x: otherwise, it will be impossible to perform the comparison just described, and there is no means of deciding to accept or reject y.

It can be proven that the probability of acceptance in the accept-reject method is $1/M$ [22]. Therefore, the efficiency of the method depends on finding an auxiliary distribution that is close to the target one, in the sense that $M \approx 1$, and that it is also easy to sample from. It is not so obvious which auxiliary distribution is the best one. Another drawback of the method is that its efficiency in high dimensions is very poor [22].

Note that we learned how to generate samples from virtually *any* probability distribution, but we always needed at least samples from the uniform distribution on the interval $[0, 1]$. In the same correspondence cited in the beginning of this section, von Neumann also proposed a method to generate these samples using the *middle-square method*, a deterministic algorithm that generates numbers following approximately this distribution. He noted that the sequences generated are periodic with a very short period, and do not pass through many statistical tests of randomness. Nevertheless, he wrote that this method was good enough for his purposes. Nowadays one of the most widely used methods of generating pseudo-random num-

bers from a uniform distribution is the much more efficient *Mersenne Twister* [23]. But if one is interested in possessing genuine uniform random numbers, one can buy them [24]!

Being the development of these methods secret, von Neumann and Ulam needed a code to denote and talk about it. The name *Monte Carlo methods* was suggested by a colleague of them, Nicholas Metropolis[2], referring to the Monte Carlo Casinos in Monaco, where Ulam's uncle used to gamble. Nowadays, the Monte Carlo methods are a very wide class of algorithms, and a particular class that will be of great importance for us is the *Markov Chain Monte Carlo* methods, that we will see in the next sections.

3.2 Markov Chain Monte Carlo (MCMC) methods

Recall that in the previous section we pointed out some problems of the accept-reject method: the difficulty of creating a good auxiliary distribution and its inefficiency in higher dimensions. On the other hand, the samples it generates are exact and independent, allowing us to use the Central Limit Theorems to compute quantities related to the target distribution. However, in 1906 Andrey Markov[3] published a work [25] introducing a new class of stochastic process, described by dependent random variables with a well-behaved and very simple dependence relation, for which a version of the Central Limit Theorems is also valid, leading to a class of theorems known as *ergodic theorems*. This process is now known as *Markov chain*, and curiously at that time it was introduced without any practical purpose, but only to serve as a counterexample! The first practical application of this concept dates back to 1917, when Erlang[4] used the Markov chains to model call loss and waiting time in the telephone network of Denmark [26]. The basic concepts of Markov chain are beyond the scope of this text. A very succinct exposition leading directly to the MCMC methods can be found in [11], and a much more technical and detailed text is [27].

3.2.1 From Monte Carlo to MCMC methods

Some years after the correspondence between von Neumann and Ulam, again at Los Alamos Nicholas Metropolis needed to compute the free energy of a complicated thermodynamic system. Being unable to do the calculation analytically, the

[2]Nicholas Constantine Metropolis, Chicago, 11 June, 1915 – Los Alamos, October 17, 1999.
[3]Andrey Andreyevich Markov, Ryazan, June 14, 1856 – Petrograd, July 22, 1922.
[4]Agner Krarup Erlang, Lønborg, January 1, 1878 – Copenhagen, February 3, 1929.

solution given by his friends to approximate the computation by some stochastic simulation seemed to be a good idea. But now the scenario was quite different: a thermodynamic system consists of a lot of particles, and the probability distributions they needed to deal with were very high-dimensional, making impossible the efficient use of any known sampling method. However, after some hard work the problem was solved by Metropolis together with Arianna W. Rosenbluth[5], Marshall N. Rosenbluth[6], Augusta H. Teller[7], and Edward Teller[8] and published in 1953 [28], leading to the *Metropolis algorithm*[9]. The main idea of the proposed solution is to construct not a sequence of independent and exact samples from a target distribution, but at each iteration sample from different random variables forming a Markov chain whose stationary distribution is the target distribution. Therefore, the samples obtained are not exactly from the desired distribution, but only from an approximation of it. It is expected that if the algorithm runs for sufficiently long time, the output of the chain can be considered as samples from the exact distribution, after some iterations called the *burn-in time*, where the chain does not converged. The convergence of Markov chains is very difficult to assess, and we will return to this point later in this chapter.

The Metropolis algorithm was generalized several years later in 1970 by Hastings[10] [29], leading to the *Metropolis-Hastings algorithm*[11], which we present in the next section.

3.2.2 The Metropolis-Hastings algorithm

The idea of this algorithm is similar to the accept-reject method: a sample from some auxiliary distribution is generated and accepted as a genuine sample of the target distribution if some criteria is satisfied. More precisely, let $f(.)$ be the (possibly multidimensional) target distribution and let $\mathbf{x}^{(i)}$ be the current state of the chain, that is, the last sampled and accepted value for $f(.)$. A new value $\mathbf{x}^*$ is then sampled from an auxiliary distribution, called the *proposal distribution* and denoted by $q(\cdot; \mathbf{x}^{(i)})$. Note that the proposal distribution is allowed to depend on $\mathbf{x}^{(i)}$, as expected when working with Markov chains. This sample is accepted as a genuine

[5]Information not found.

[6]Marshall Nicholas Rosenbluth, Albany, February 5, 1927 – San Diego, September 28, 2003.

[7]Information not found.

[8]Edward Teller, Budapest, January 15, 1908 – Stanford, September 9, 2003.

[9]In fairness to the other authors, the algorithm should rather be known as the *Metropolis-Rosenbluth-Rosenbluth-Teller-Teller algorithm.*

[10]W. K. Hastings, Toronto, July 21, 1930.

[11]Followig the same reasoning, this algorithm should be called the *Metropolis-Rosenbluth-Rosenbluth-Teller-Teller-Hastings algorithm.*

sample from $f(.)$ with probability

$$\alpha(\mathbf{x}^{(i)}, \mathbf{x}^*) = \min\left(1, \frac{f(\mathbf{x}^*)q(\mathbf{x}^{(i)}; \mathbf{x}^*)}{f(\mathbf{x}^{(i)})q(\mathbf{x}^*; \mathbf{x}^{(i)})}\right). \tag{3.3}$$

If the proposed sample is accepted, the new state of the chain is $\mathbf{x}^{(i+1)} = \mathbf{x}^*$, and $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)}$ otherwise. The initialization of the algorithm, i.e. the determination of $\mathbf{x}^{(0)}$, is left at the choice of the user.

Note that the condition to accept the proposed sample in Equation 3.3 is much more complicated that the one in the accept-reject method in Equation 3.1. This is to ensure that the corresponding Markov chain satisfies the *detailed balance* equation [11, 27], a sufficient condition for $f(.)$ to be the stationary distribution of the chain. If this and two more technical conditions[12] are satisfied, the Markov chain converges to $f(.)$, in the sense that samples from it are closer and closer to samples from $f(.)$.

The efficiency of the algorithm depends on the choice of the proposal distribution. A popular choice is a Gaussian distribution centered at the previous value of the chain, that is, $q(\mathbf{x}; \mathbf{x}^{(i)}) \sim N(\mathbf{x}; \mathbf{x}^{(i)}, \mathbf{D})$, where $\mathbf{D}$ is a diagonal matrix of the same dimension as $\mathbf{x}$. How to choose the diagonal terms in $\mathbf{D}$ is a delicate task: a broad proposal distribution is very likely to propose values where $f(.)$ has very small values, leading to a large amount of rejected values; on the other hand, a more concentrated proposal distribution does not explore well the sample space of $f(.)$. Some authors suggest tuning the variance until 40% of samples are accepted, as a compromise between both situations [30].

3.2.3 The Gibbs sampler

Fourteen years after the generalization of the Metropolis algorithm by Hastings, in 1984 the brothers Stuart[13] and Donald Geman[14] were working with restoration of degraded monochromatic images. Inspired by Statistical Mechanics, they proposed a mathematical model for the images that was very similar to the Ising[15] model, widely used in Ferromagnetism. To restore images via their method, it was necessary again to sample from very high-dimensional probability distributions, and unaware of the previous work by Metropolis et. al. and Hastings, they proposed a new method to generate samples from multidimensional probability distributions, known as the *Gibbs sampler*[16] [31].

The main idea of the Gibbs sampler is to split a multivariate distribution in

[12] Namely, *aperiodicity* and *irreducibility*. For more details, see [27].

[13] Stuart Alan Geman, Chicago, c. 1949.

[14] Donald Jay Geman, Chicago, September 20, 1943.

[15] Ernst Ising, Cologne, May 10, 1900 – Peoria, May 11, 1998.

[16] Josiah Willard Gibbs, born in February 11, 1839 in New Haven, Connecticut, died in April 28, 1903, more than eighty years before the emergence of the method!

several components, and sample from each component conditioned on the value of the others. More precisely, let $f(\mathbf{x})$ be the target distribution we want to sample from, and partition the $\mathbf{x}$ variable in k subsets, $\mathbf{x}_1, \ldots, \mathbf{x}_k$. Denote by $\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_k^{(i)}$ the current sampled value of each sub-variable. The next iteration of the algorithm is then given by:

$$\mathbf{x}_1^{(i+1)} \sim f(\mathbf{x}_1 | \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_k^{(i)}) \tag{3.4}$$

$$\mathbf{x}_2^{(i+1)} \sim f(\mathbf{x}_2 | \mathbf{x}_1^{(i+1)}, \mathbf{x}_3^{(i)}, \ldots, \mathbf{x}_k^{(i)}) \tag{3.5}$$

$$\vdots$$

$$\mathbf{x}_k^{(i+1)} \sim f(\mathbf{x}_k | \mathbf{x}_1^{(i+1)}, \ldots, \mathbf{x}_{k-1}^{(i+1)}), \tag{3.6}$$

where the symbol $\mathbf{y} \sim g(.)$ means "sample $\mathbf{y}$ from the distribution $g(.)$". Note that in each step of this iteration the sampled values of the previous variables must be updated in the respective conditional distributions. The initial set of samples $\mathbf{x}_1^{(0)}, \ldots, \mathbf{x}_k^{(0)}$ can be constructed in any way the user desires.

It can be proven that this algorithm in fact creates a Markov chain whose stationary distribution is $f(.)$ [27], and although the Geman brothers did not know the Metropolis-Hastings algorithm at that time, it can also be proven that the Gibbs sampler is a particular case of it [27, 30]. Both methods became popular in the statistical community after the work [32].

3.2.4 Metropolis within Gibbs

In order to apply the Gibbs sampler, one must be able to sample from all conditional distributions described above. But sometimes we are quite unlucky, and some of these distributions are not easy to sample from. However, there is a simple solution to this problem, leading to the algorithm known as *Metropolis within Gibbs* [30].

Suppose, for example, that we desire to sample from a bivariate distribution $f(x_1, x_2)$, with conditionals given by $f(x_1|x_2)$ and $f(x_2|x_1)$. Furthermore, suppose that only the first conditional distribution is easy to sample from. Instead of directly sampling from $f(x_2|x_1)$ when required, we can sample a value x_2^* from a proposal distribution $q(\cdot; x_1)$ and accept this value as a genuine sample from $f(x_2|x_1)$ with probability given by Equation 3.3. This procedure is called a *Metropolis step* inside the Gibbs sampler, and although it looks like an heuristic solution to the problem, the convergence of this modified Gibbs sampler can also be proven [27].

3.2.5 Convergence issues

All MCMC methods presented above have the drawback of generating only approximate samples from the target distributions; moreover, such samples are also dependent on each other. However, for most purposes, the second problem can be circumvented by the ergodic theorems, but the first one is quite more complicated to ignore. Recall that in all cases the corresponding Markov chain is constructed in order to set the target distribution as the stationary distribution. Moreover, the chain also satisfies conditions that guarantee its convergence to the stationary distribution. But how long one must wait in order to be sufficiently close to the target distribution? This is not an easy question to answer, and a lot of study has been done in this direction. There are some partial results, mainly if the state space of the chain is discrete [33]. Even without being able to give a definitive answer to this question, there are a lot of convergence diagnostic tools [34], but all of them possess some serious drawback.

In this work, we will adopt a heuristic practice that showed to be very effective. We will define a certain number of iterations, called the *burn-in time* and denoted by $N_{\text{burn-in}}$, such that the samples before $N_{\text{burn-in}}$ are discarded since the chain is still "moving" towards its stationary distribution, and the samples after $N_{\text{burn-in}}$ are used to compute some value of interest about the distribution. This procedure will be performed visually, by looking at the plots of the iterations and searching for some point where the chain exhibits a constant and regular pattern. It is important to note that this is not the usual procedure in the statistics community: some statisticians recommend running two or more chains in order to correctly access the convergence, instead of looking at only a single chain. See [30] for a more extensive discussion.

Chapter 4

Autoregressive models for audio signals

"Essentially, all models are wrong, but some are useful."

– George E. P. Box[1]

Many audio signals we hear everyday are produced by either musical instruments or human voice – both having some well defined (but not necessarily well known) physical and acoustic properties, or by synthesizers, which by means of electric and/or electronic devices can mimic real instruments or create entirely new sounds. Even though the later does not necessarily follow the same physical principles of the former, a great variety of both kinds of sounds share a common property: our ears and brain can extract some "useful" information from them. The music being played can have a weird melodic or harmonic structure, and even if one does not appreciate it, the music does not sound like random noise.

The above discussion suggests that the majority of audio signals we perceive and understand possess some underlying structure. The goal of this chapter is to present and discuss a mathematical model for this structure. In the next sections we will convince ourselves that a deterministic model does not fit properly, since we want to deal with a large class of audio signals. We then present the autoregressive (AR) model and discuss its interpretations from both the statistical and signal processing viewpoints. Finally, in the last section we discuss to which extent the AR model is valid for audio signals.

4.1 The autoregressive (AR) model

It was mentioned above that most of the audio signals we hear and interpret, in the sense that they have some useful information, possess some internal coherence

[1]George Edward Pelham Box, Gravesend, October 18, 1919 – Madison, March 28, 2013.

that can be exploited in order to describe them. Roughly this means that these signals are not just noise, or simply random amplitudes drawn from some probability distribution.

Indeed, this fact can be extended to all the information we can perceive and comprehend: our brain prefers structured information to pure noise. This is depicted in Figure (4.1). On the left, we see one of the most famous paintings in the world, the Mona Lisa. Clearly, to paint it, Leonardo da Vinci have not picked random brushes, random colors and made random strokes until "converge" to the Mona Lisa. On the other hand, the figure on the right was created in this fashion: a random matrix was constructed, each entry of which is zero (represented as black) or one (represented as white) with probability 1/2 and associated the zeros to the black color and the ones to the white color.

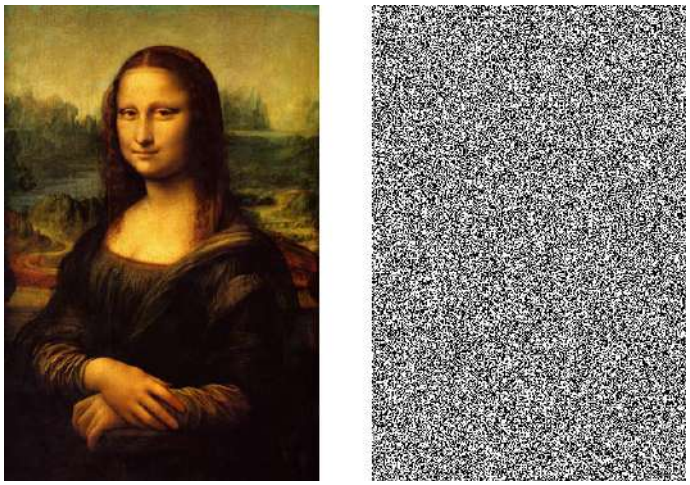

Figure 4.1: Mona Lisa and a random image.

If you look at a pixel in the forehead of Mrs. Lisa del Giocondo, you can expect that its neighbor pixels are similar, in some sense. However, in the noisy image this does not occur, since the color of one pixel does not depend on its neighbors. Now that we have gained some intuition, let us bring this idea to the audio context.

Let $\mathbf{x}$ be a vector containing some generic audio signal, properly time-sampled[2]. Due to the enormous variety of audio signals, we cannot expect to create a reasonable deterministic model to all of them: if this model exists, it will be so complex, with so many variables, that it cannot be practical. So we can suppose that each time sample x_n from the signal $\mathbf{x}$ is a sample from some abstract random variable[3], which

[2]By this we mean that the original analog signal was adequately filtered with an anti-aliasing filter and sampled at a sampling rate in such a way that allows us to recover the original signal, e.g. 44,100 Hz. We will always make this hypothesis when talking about audio signals.

[3]Pay attention to the use of the word "sample" here! The first use was to denote a sample of a discrete-time signal, while the second denotes the act of taking a (pseudo)random number following some probability distribution.

we will denote as X_n. Then the set $\{X_n\}$ is a *stochastic process*, and the vector $\mathbf{x}$ is a *time series*, that is, a sample – in the statistical sense – of the underlying process $\{X_n\}$. Obviously there are infinitely many ways of translating the internal structure of the signal into the description of the stochastic process, but the temporal order of audio signals tells us that it is reasonable that every random variable X_n depends in some way on the precedent ones X_k, for $k < n$. But in principle X_n will have little relation with some distant X_k, for $k \ll n$[4].

Thus, we can say that X_n is a function of $X_{n-1}, X_{n-2}, \ldots, X_{n-P}$, for some P[5]. A class of functions that keeps some compromise between simplicity and reality is the affine one, that is, the $n-$th random variable of the stochastic process is a linear combination of the P precedent random variables plus an error, called the *innovation error*:

$$X_n = \sum_{i=1}^{P} a_i X_{n-i} + E_n. \tag{4.1}$$

We assemble the coefficients $a_1, \ldots, a_P$ into a column vector $\mathbf{a}$. The innovation error must include every information within the signal that cannot be well explained linearly by the previous time samples. In order to allow E_n to contain the greatest amount of information possible, it is usually modeled as independent and identically distributed (iid) Gaussian random variables with zero mean and variance σ_e^2, independent of X_n. This choice of distribution is justified by the Principle of Maximum Entropy [13]:

Theorem 4.1.1 (Principle of Maximum Entropy)**.** *Let E be a random variable with probability density function f whose support is $\mathbb{R}$, and define the differential entropy of E as $h(E) = -\int_{-\infty}^{+\infty} f(x)\ln(f(x))\,dx$. Suppose that the mean and variance of E exist, are finite and equal to μ and σ^2, respectively. Then the distribution that maximizes $h(E)$ is a Gaussian with mean μ and variance σ^2.*

Despite appearing mysterious, this result possesses a nice interpretation. The differential entropy measures how much "information" a continuous probability distribution contains, in the sense of measuring its uncertainty: the more uncertain is

[4]One can argue that a violin playing some note for a long time breaks this hypothesis. Or even worse, if there is the exact repetition of some fragment within a piece, this hypothesis is broken in an even more radical way. But note that these two cases represent long-term correlations within an audio signal, and this is not the goal of the AR model. Indeed, as will be more clear in Section 4.3, this model is adequate to represent aspects of the physical creation of the sound, related with resonant frequencies of the corresponding instrument or voice. Although more easily interpretable in the frequency domain, it is easier to describe these features in the time domain via the AR model.

[5]The correlation between two distant time samples of the signal depends not only on P, but also on the position of poles of the linear filter associated with the AR model, to be discussed in more detail in Section 4.3.

the distribution, the greatest is its entropy [35][6]. This is more easily understood if the support of the distribution is a closed interval $[a, b]$. In this case, it can be shown that the distribution of maximum entropy in this domain is the uniform [35]. This is in accordance with our interpretation, since this distribution does not prioritize any region of its support, leading to the greatest amount of uncertainty, in the intuitive meaning of the word. Following this reasoning, one expects that the maximum entropy distribution over the entire real line is as most uniform as possible, since the uniform "distribution" is not a probability distribution over $\mathbb{R}$! The above results tells us that the most uniform distribution over $\mathbb{R}$ (that is, the maximum entropy distribution) is a Gaussian.

If a stochastic process satisfies Equation 4.1, we say that it is an *autoregressive process of order* P, abbreviated by AR(P). We will suppose that short sections of audio signals are well described by this model, in the sense that the time samples contained in each of these blocks of signal can be supposed to be samples from a stochastic process satisfying Equation 4.1, for a particular set of coefficients $\mathbf{a}$ and σ_e^2. The order P and how large can a block be for the model be considered sufficiently accurate will be discussed in more detail later.

This model is widely studied in Statistics, because of its simple form and its ability to fit several phenomena of importance. It can be generalized in several ways, usually described in the literature by acronyms like ARMA (autoregressive moving-average), TVAR (time-varying autoregressive), ARMAX (autoregressive moving-average with exogenous inputs), ARIMA (autoregressive integrated moving average), ARFIMA (autoregressive fractionally integrated moving average), SARIMA (seasonal autoregressive integrated moving average), SARIMAX (seasonal autoregressive integrate moving average with exogenous inputs), and so on [36].

Perhaps the complexity of these models increases proportionally with the size of the corresponding acronym and, on the other hand, the intuition about them decreases in inverse proportion. It is natural to ask whether the use of a more complex model allows for a better representation of audio signals. Indeed, in Section 5.2.4 of [37], the ARMA model for interpolation of missing time samples in audio signals is explored, and the conclusion is that in this particular case the increase in accuracy does not justify the increase in complexity. The TVAR seems to be a natural extension of the AR model for audio signals, and it is presented in this context in [38], with good results being reported. The use of this extended model

[6]The concept of entropy of a random variable was first defined by Shannon (Claude Elwood Shannon, Petoskey, April 30, 1916 – Medford, February 24, 2001), and this name was suggested to him by von Neumann: *"You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name, so it already has a name. In the second place, and more important, no one really knows what entropy really is, so in a debate you will always have the advantage."*.

to perform restoration procedures is one of the possible future works in this thesis. We will convince ourselves in Section 4.3 that the AR model can provide a very accurate representation for audio signals.

4.1.1 How large a block and the order must be?

If the audio signal is time-sampled at 44,100 Hz, the CD sampling rate, a monophonic audio signal which is 4 minutes long is represented by a vector $\mathbf{x}$ of dimension 10,584,000. It is not reasonable to assert that all these random variables are well described by a single set of parameters through Equation 4.1: since the structure of the signal may change along the time, the parameters representing the dependence of time samples on the previous ones also could change.

But one does not need to look at the audio signal in such a big time scale to note that a single set of parameters is not enough. Indeed, it is enough to consider one set of parameters $\mathbf{a}$ and σ_e^2 for each block of approximately 20 milliseconds (ms), about 1,000 time samples at the above sampling rate, and an order around 40. The size of 1,000 for a block is the approximate extent to which audio signals can be considered as stationary, and the order of 40 is enough to model the frequencies and resonances within the considered block (for more details, see [37] and references therein, and [39] for a discussion from the lossless audio compressing viewpoint). Obviously some exceptions can occur, for example, the beginning of a plosive phoneme in a speech signal or the attack of a percussion instrument, and in this work we will consider other values for these parameters as well. These numbers should be considered only as general guidelines, not strict rules to be followed[7].

A possible explanation from the psychoacoustical viewpoint for the size of the block is now presented. Suppose that a digital audio signal consists of single pulses equally spaced with separation equal to L time samples, that is, every time sample of the considered signal is zero, except the time samples indexed by integer multiples of L, that are equal to one. If one plays this signal by considering its sampling rate of 44,100 Hz, for L greater than a value of approximately 1,000 it will be possible to distinguish the individual pulses, whereas for L smaller than this value, a single tone will be heard. This experiment indicates that this number is related with some kind of "resolution" in our process of hearing, and blocks smaller than this size are essentially indistinguishable.

[7] Recall the quote at the beginning of this chapter!

4.2 Statistical description of audio signals using the AR model

Since we have a probability distribution for the innovation error, we can deduce one for the audio signal as well. Let $\mathbf{x} = [x_1\ x_2\ \ldots\ x_N]^T$ be a block of N time samples from an audio signal. For simplicity, we partition this vector in two sub-vectors of size P and $(N-P)$, respectively:

$$\mathbf{x}_0 = [x_1\ x_2\ \ldots\ x_P]^T \quad \text{and} \quad \mathbf{x}_1 = [x_{P+1}\ \ldots\ x_N]^T. \tag{4.2}$$

For now consider that the vector $\mathbf{x}_0$ is known, since to compute some x_n we need to know the previous P entries of $\mathbf{x}$. In particular, to compute x_{P+1} we need the information in $\mathbf{x}_0$, and since our signal begins at x_1, there is nothing more we can resort to in order to compute x_{P+1}.

We can also assemble the innovation errors from $(P+1)$ to N into a vector:

$$\mathbf{e} = [e_{P+1}\ e_{P+2}\ \ldots\ e_N]^T. \tag{4.3}$$

Recall that the innovation error was modeled as iid Gaussian random variables with mean 0 and variance σ_e^2, that is,

$$p_{E_n}(e_n) = \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\left(-\frac{1}{2\sigma_e^2} e_n^2\right), \text{ for } n = P+1, \ldots, N, \tag{4.4}$$

or equivalently, by taking the product of the above pdf's,

$$p_{\mathbf{E}}(\mathbf{e}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2} \mathbf{e}^T\mathbf{e}\right). \tag{4.5}$$

If we write the equations for the AR model for $x_{P+1}, \ldots, x_N$, we have

$$\begin{cases} x_{P+1} &= a_1 x_P + a_2 x_{P-1} + \cdots + a_P x_1 + e_{P+1} \\ x_{P+2} &= a_1 x_{P+1} + a_2 x_P + \cdots + a_P x_2 + e_{P+2} \\ \quad \vdots & \\ x_N &= a_1 x_{N-1} + a_2 x_{N-2} + \cdots + a_P x_{N-P} + e_N, \end{cases} \tag{4.6}$$

and after solving the above equations for $e_{P+1}, \ldots, e_N$, we obtain:

$$\begin{cases} e_{P+1} &= x_{P+1} - a_1 x_P - a_2 x_{P-1} - \cdots - a_P x_1 \\ e_{P+2} &= x_{P+2} - a_1 x_{P+1} - a_2 x_P - \cdots - a_P x_2 \\ \quad \vdots & \\ e_N &= x_N - a_1 x_{N-1} - a_2 x_{N-2} - \cdots - a_p x_{N-P}. \end{cases} \tag{4.7}$$

Since we are supposing that $\mathbf{x}_0$ is known, the above equations describe a linear change of variables from $\mathbf{x}_1$ to $\mathbf{e}$, which has unitary Jacobian. This is easily seen by noting that the Jacobian matrix is given by the $(N-P) \times (N-P)$ matrix below:

$$\mathbf{J}_{ij} = \left(\frac{\partial e_i}{\partial x_j} \right) = \begin{bmatrix} 1 & 0 & \ldots & \ldots & \ldots & \ldots & 0 \\ -a_1 & 1 & 0 & \ldots & \ldots & \ldots & 0 \\ -a_2 & -a_1 & 1 & 0 & \ldots & \ldots & 0 \\ -a_3 & -a_2 & -a_1 & 1 & 0 & \ldots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & -a_P & \ldots & -a_1 & 1 & 0 \\ 0 & 0 & \ldots & -a_P & \ldots & -a_1 & 1 \end{bmatrix}, \tag{4.8}$$

whose determinant equals 1, since it is upper triangular with ones on the diagonal. In matrix terms, we can write Equation 4.7 in two different ways:

$$\mathbf{e} = \mathbf{A}\mathbf{x} \quad \text{or} \quad \mathbf{e} = \mathbf{x}_1 - \mathbf{X}\mathbf{a}, \tag{4.9}$$

where the matrices $\mathbf{A}$, of size $(N-P) \times N$, and $\mathbf{X}$, of size $(N-P) \times P$ are given respectively by

$$\mathbf{A} = \begin{bmatrix} -a_P & \ldots & -a_1 & 1 & 0 & \ldots & 0 \\ 0 & -a_P & \ldots & -a_1 & 1 & \vdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & -a_P & \ldots & -a_1 & 1 \end{bmatrix}, \tag{4.10}$$

and

$$\mathbf{X} = \begin{bmatrix} x_P & x_{P-1} & \ldots & \ldots & x_2 & x_1 \\ x_{P+1} & x_P & \ldots & \ldots & x_3 & x_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ x_{N-2} & x_{N-3} & \ldots & \ldots & x_{N-P} & x_{N-P-1} \\ x_{N-1} & x_{N-2} & \ldots & \ldots & x_{N-P+1} & x_{N-P} \end{bmatrix}. \tag{4.11}$$

We can then write down a statistical description of $\mathbf{x}_1$, supported by Theorem 2.4.1:

$$p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{a}, \sigma_e^2) = p_{\mathbf{E}}(\mathbf{A}\mathbf{x}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2}\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}\right), \tag{4.12}$$

or equivalently,

$$\begin{aligned} p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{a}, \sigma_e^2) &= p_{\mathbf{E}}(\mathbf{x}_1 - \mathbf{X}\mathbf{a}) \\ &= \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2}(\mathbf{x}_1 - \mathbf{X}\mathbf{a})^T(\mathbf{x}_1 - \mathbf{X}\mathbf{a})\right). \end{aligned} \tag{4.13}$$

Note that the first equation is quadratic in $\mathbf{x}$, while the second one is quadratic in $\mathbf{a}$. This fact will be explored in later chapters to obtain some likelihoods and conditional distributions.

The Equations 4.12 and 4.13 are called the *conditional likelihood* of parameters $\mathbf{a}$ and σ_e^2. This name derives from the conditioning on $\mathbf{x}_0$, which was essential to the described procedure.

If one is performing a block-wise restoration procedure, an overlap of P time samples between adjacent blocks solves the conditioning problem. The only block that must be treated separately is the first one, since there are no time samples on which to condition. On the other hand, if one is interested in estimating the parameters $\mathbf{a}$ and σ_e^2 by maximum likelihood, a potentially more reliable estimate can be obtained without the conditioning, since the statistical information about the parameters that is contained in $\mathbf{x}_0$ is being ignored. To surpass this, one must use the *exact likelihood* $p(\mathbf{x}|\mathbf{a}, \sigma_e^2)$, which is given by $p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{a}, \sigma_e^2)p(\mathbf{x}_0|\mathbf{a}, \sigma_e^2)$. Usually this is not performed, because the computation of the additional term $p(\mathbf{x}_0|\mathbf{a}, \sigma_e^2)$ is quite complicated, as seen in Appendix C of [37]. Clearly this is not a rigorous justification for not using the exact likelihood instead of the conditional one, but an acceptable excuse. A better reason would be a Theorem asserting that ignoring this term is not so bad. In fact, one can prove that for $N \gg P$ the following approximation is true [40]:

$$p(\mathbf{x}|\mathbf{a}, \sigma_e^2) \approx p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{a}, \sigma_e^2), \tag{4.14}$$

that is, the exact likelihood is approximately equal to the conditional likelihood, if we have enough time samples. Intuitively, if we have a sufficient number of time samples such that for the majority of time samples its dependency on the first P ones is "lost", then the exact likelihood carries just a little more information in comparison to the approximated one.

4.3 Interpretation of the AR model in the frequency domain

When we introduced the AR model in Equation 4.1, there was no clue on how to interpret it in the frequency domain, a tool widely employed in signal processing, specially audio processing. In this section we first interpret the AR model as a filter acting on a signal, giving as output the desired audio excerpt. Then we analyze the frequency content of this last signal in order to get a new intuition about the model.

Let us remember Equation 4.1, that defines a stochastic process following the AR model:

$$X_n = \sum_{i=1}^{P} a_i X_{n-i} + E_n. \tag{4.15}$$

Since we are considering contiguous short sections of an audio signal as a realization of a particular stochastic process of this kind, when such a signal is available we can exchange the capital letters in the above equation (except for P, of course!) for small ones:

$$x_n = \sum_{i=1}^{P} a_i x_{n-i} + e_n. \tag{4.16}$$

Now, the above equation is written in terms of time samples from a signal, not random variables anymore. We can take the z-transform on both sides of the equation, and we have[8]:

$$X(z) = X(z) \sum_{i=1}^{P} a_i z^{-i} + E(z), \tag{4.17}$$

and we can rewrite this equation as:

$$X(z) \left(1 - \sum_{i=1}^{P} a_i z^{-i} \right) = E(z). \tag{4.18}$$

If we think that the signal $\mathbf{e}$ is an input for some filter whose output is $\mathbf{x}$, the

[8]To be coherent with almost all signal processing books, we denote the z-transform of a signal by its correspondent capital letter followed by (z), indicating that it is a function of the complex variable z. If this signal arises from observations of a stochastic process $\{X_n\}$ this must cause no confusion, despite both being represented by the capital letter X.

above equation gives us the transfer function of this filter. More precisely[9]:

$$A(z) := \frac{X(z)}{E(z)} = \frac{1}{\left(1 - \sum_{i=1}^{P} a_i z^{-i}\right)}. \tag{4.19}$$

This analysis provides a new insight about the AR model: a time-series that is well-explained by an AR model is the output of an all-pole filter, whose input is the random sequence $\mathbf{e}$. We now interpret this fact in the frequency domain, where it is even more interesting. First, we must recall two definitions and some facts from random signal processing theory.

Definition 4.3.1. *A stochastic process $\{X_n\}$ is wide-sense stationary (WSS) if its mean and auto-correlation function satisfy, respectively, the conditions below:*

- $\mathbb{E}(X_n) = \mu_X$ *is independent of n, and*
- $\mathbb{E}(X_n X_{n+k}) = r_{XX}(k)$ *is only dependent on the lag k between two time samples, and not on their absolute positions.*

Observation 4.3.1. *One can also define a strict-sense stationary (SSS) stochastic process. We will not need this definition here. It is common both in the Statistics and Signal Processing literature to refer to WSS processes simply as "stationary", and we will adopt this practice here.*

In other words, the first- and second-order statistics of a stationary stochastic process are time-invariant.

It is possible to measure the frequency content of a stochastic process. Unfortunately one cannot simply compute the discrete Fourier transform of realizations of this process, since it is very likely that these signals do not satisfy the conditions to possess a well-defined transform. In practice this is not a big issue here, since in the end we will be dealing only with finite-dimensional sequences. The following point is more important: even if one is very lucky and the given realization can be transformed, the computation will reveal the frequency content only of this realization, providing no information about the others. There are several ways of defining a mathematical entity containing the required information, and we will adopt the one below. The motivation for this definition can be found in the very clear exposition in Section 7.1 of [41].

[9]It is more usual in the signal processing literature to denote transfer functions of linear filters by $H(z)$, but since the filter related to the AR model is quite special and will appear several times during this text, we will denote it by a special letter.

Definition 4.3.2. *The power spectral density (PSD) of a stationary stochastic process $\{X_n\}$ is given by the discrete Fourier transform of its auto-correlation function:*

$$P_{XX}(\omega) = \sum_{k=-\infty}^{+\infty} r_{XX}(k) e^{-j2\pi\omega k}, \tag{4.20}$$

where ω is measured in radians per time sample.

Observation 4.3.2. *There is a more general definition of power spectral density of stochastic processes, which is valid for other classes than the stationary one. This definition is outside the scope of this thesis, and we recommend the book [40] for a complete treatment of spectral theory of time series. The definition given above is in fact a (very surprising) theorem, known as the Wiener[10]-Khinchin[11] theorem[12].*

To compute the PSD of an AR process, we have some obstacles: 1) to use the PSD definition given above, we must ensure that the process is stationary, a condition equivalent to the stability of the filter whose transfer function is $A(z)$; and 2) we must compute its auto-correlation function, a task that can be tedious. We first tackle the second obstacle and for the moment suppose stationarity. Since $\mathbf{x}$ is the output of the filter whose transfer function is $A(z)$ when the input is $\mathbf{e}$, the relation between their respective PSDs is given by:

$$P_{XX}(\omega) = |A(e^{j2\pi\omega})|^2 P_{EE}(\omega). \tag{4.21}$$

Observation 4.3.3. *This relation also is valid only for stationary processes, and a proof can be found in Section 8.4 of [41]. Note that the process $\{E_n\}$ is WSS.*

Since the process $\{E_n\}$ is iid Gaussian with mean 0 and variance σ_e^2, its PSD is given by σ_e^2. Then, the above formula reduces to:

$$P_{XX}(\omega) = \frac{\sigma_e^2}{\left|1 - \sum_{i=1}^{P} a_i e^{-j2\pi\omega i}\right|^2}. \tag{4.22}$$

Now, with the PSD of an AR process at hand, we can continue with its interpretation in the frequency domain. Since we are assuming stationarity, the causal filter whose transfer function is $A(z)$ is stable. This is equivalent to saying that its poles are all inside the unit circle[13]. If we write the poles of $A(z)$ in polar form as

[10] Norbert Wiener, Columbia November 26, 1894 – Stockholm March 18, 1964.

[11] Aleksandr Yakovlevich Khinchin, Kondrovo, July 19, 1894 – Moscow, November 18, 1959.

[12] This theorem is also known as the *Wiener-Khinchin-Einstein* or *Khinchin-Kolmogorov* theorem. The history of Science does not allow us to ignore the name of both great scientists, so for a more complete exposition of the history of this result, see the references within the Wikipedia page of the theorem [42].

[13] For a proof of this result, see [43].

$r_i e^{j\theta_i}$, for $i = 1, \ldots, P$, then one expects $P_{XX}(\omega)$ to be peakier for ω around θ_i if the corresponding radius r_i is close to one. Since $P_{XX}(\omega)$ contains the information on frequencies present in $\mathbf{x}$, the arguments of the poles of $A(z)$ are related to these frequencies and their respective radii measure how much this frequency is being excited.

Note that this interpretation is perfectly correlated with our experience in hearing music! At each instant of time there is a set of frequencies being played: for example, the fundamental notes present in some chord and its harmonics (due to the physical properties of the instrument being played), and each frequency possesses its own amplitude, which means that some frequencies are being more excited than others.

Recall that Equation 4.1 at the beginning of this chapter was proposed only as a way to translate into mathematical terms some internal correlation present in audio signals, and its interpretation there was very loose. We have now seen that this very simple structure is intimately related to fundamental properties of audio signals. From the signal processing viewpoint, the generation of pitched sounds is related with resonances, well modeled by poles in its generating system.

To close this section, it is interesting to note that the AR model was born in an entirely different context: the first time it appeared in history, explicitly used to model time series[14], was in a work of Yule[15] from 1927 [44], where he applied the AR model to study the Wolf's Sunspot Number, one of the most widely studied time series in Statistics. These data are being collected on a regular basis since 1750, and this quantity measures the number of sunspots and groups of sunspots present on the surface of the Sun. It is important to understand the periodicity behavior of this time series, since sunspots are related to solar storms, that can in turn disturb the operation of electromagnetic devices here on Earth. These phenomena correspond to concentrations of magnetic field flux, and the details of its generation are still not well understood.

[14] A mathematical formulation similar to the AR model appeared before in recursive least square filters, firstly proposed by Gauss in the 19th. Century.

[15] George Udny Yule, Morham February 18, 1871 - Cambridge June 26, 1951.

Part II

RESTORATION OF AUDIO SIGNALS WITH NONLINEAR DISTORTIONS VIA THE AR MODEL

Chapter 5

Introduction

"Happy families are all alike; every unhappy family is unhappy in its own way."

– Lev Tolstoy[1], *Anna Karenina*

This second part of the thesis deals with the restoration of nonlinearly distorted audio signals, a class of degradations whose mechanism of creation is not well approximated by a linear filter acting on the original signal. We need more sophisticated mathematical structures to represent this class of distortions. This part is structured in the following way: in this chapter we introduce the concept of nonlinear distortion, pointing out its differences from the linear ones in Sections 5.1 and 5.2, and finally discussing its effects on audio signals on Section 5.3; Chapter 6 presents some ways to describe the nonlinear world, introducing the Volterra[2] series model and some of its particular cases, finishing with some applications of these concepts in audio processing; the most technical Chapters, 7 and 8, deal with the restoration of audio signals degraded by nonlinear distortions without and with memory, respectively. These two chapters contain extensions of the work initiated in [45]. Finally, in Chapter 9, the results and future extensions of the work are presented.

5.1 What is a "distortion"?

A search for the word "distortion" in Google returns about 59,900,000 results. If one wants to briefly understand the term, Google's first result, the corresponding Wikipedia article in English [46], is the best option. When consulted in November 2015, it defines the term in the following way:

[1]Lev Nikolayevich Tolstoy, Yasnaya Polyana, September 9, 1828 – Astapovo, November 20, 1910.

[2]Vito Volterra, Ancona, May 3, 1860 – Rome, October 11, 1940.

Distortion is the alteration of the original shape (or other characteristic) of something, such as an object, image, sound or waveform.

Let us understand what this means. First of all, note that this is a very general definition – so general that includes cases not usually associated with distortion, in the intuitive sense of the word. For example, if you turn down the volume knob of your audio player, or if you use an equalizer to modify the frequency content of the music you hear, you are distorting the signal. Even the procedure of recording and playing an audio signal includes some degree of distortion. And if you think more carefully, even the mere act of *hearing* some music being played involves some distortion, which depends on the acoustics of the room, the physical properties of head, ears and body. All these features together modify the "original shape" of the music being played.

At this point, the reader may ask himself what we mean by the "original shape" of an audio signal, since almost *everything* in the environment can distort it, in the sense defined above. Perhaps the best definition of "original shape", in our particular case of audio restoration, is *the sound produced at the exact moment and exact location of the recording*. Every procedure done with the signal after this moment introduces distortions, in the sense defined above: inaccuracies of the recording equipment, posterior edition and mixing of the signal, injuries on the media (CD, vinyl disk, magnetic tapes), bad quality of playing devices, headphones and speakers, equalization by the listener, and so on.

Of course we will not adopt a definition as broad as the previous one. The word *distortion* brings to mind something bad, unwanted. In the context of audio restoration, it means some modification of the audio signal that is potentially unpleasant to the listener. To illustrate, not all the procedures exemplified on the last paragraph modify the signal in an annoying way: the edition and mixing of the recorded signal can be performed to provide a better experience to the listener, the inaccuracies of the recording equipment can be barely noticeable, the equalization can be done in order to make the music more pleasant to the listener and some distortions in electric guitar signals could also be pleasant. On the other hand, if you play some piece of Bach in a very poor quality speaker with a very loud volume, the listening experience will be horrible! Even if the listener has access to a high-end playing equipment, the media can be severely damaged, greatly modifying the original audio signal. Only the second kind of modification deserves to be called *distortion*: some special change on the original signal that produces a potentially annoying, unpleasant experience to the listener.

Note that this definition is personal, in the sense that something that sounds annoying for one person is not necessarily annoying to another one. There is some kind of "common sense", but disagreements occur very frequently: for example, a

distorted guitar in a heavy metal band can be very annoying for someone, but not for fans of heavy metal. However, the perception of distortions, specially the nonlinear distortions to be discussed in the next section, is a well-studied subject, and objective metrics inspired by psychoacoustics that measures the amount of nonlinearities in an audio signal exists [47]. This metric, called Rnonlin, will appear in future chapters.

5.2 And what is a "nonlinear" distortion?

Recall from the courses of Signal Processing [43] that a *linear system* is an operator $\mathcal{H}$ between two spaces of functions[3] satisfying some special conditions. If x_1 and x_2 are two admissible inputs to the system[4], let us call $y_1 = \mathcal{H}\{x_1\}$ and $y_2 = \mathcal{H}\{x_2\}$ the respective outputs. The special condition that $\mathcal{H}$ must satisfy to be called *linear* is the following equation:

$$\mathcal{H}\{ax_1 + bx_2\} = ay_1 + by_2, \tag{5.1}$$

for all real numbers a and b[5].

So, the most obvious way to define a *nonlinear system* is an operator $\mathcal{H}$ between two spaces of functions that *does not* satisfy Equation 5.1. But maybe this definition lacks utility in this simple form. It says nothing about the nature of the system, and Lev Tolstoy would say that *linear systems are all alike; every nonlinear system is nonlinear in its own way.* To illustrate this, recall that the graphs of linear functions are straight lines (and in higher dimensions are planes, hyper-planes, and so on) but the graph of every other kind of function may look very weird, even with the assumptions of continuity, differentiability, analyticity, and so on.

In order to study the effects of nonlinear systems on audio signals, we must restrict ourselves to some special cases, since the enormous variety of this kind of system does now allow us to create a generic approach that works in *all* cases. The description of such cases is the subject of the next chapter. For now, let us close this chapter with a brief description of an interesting interpretation of nonlinearity in the frequency domain.

[3]For the time being, it is not necessary to describe more precisely these spaces.

[4]This means to say that both functions belongs to the domain of the operator $\mathcal{H}$.

[5]The x and y functions do not possess the arguments (t) or $[n]$, usually employed to denote continuous- or discrete-time signals because the above definition applies to both cases. We will emphasize the specific case when it is necessary.

5.3 Effects of linear and nonlinear transformations in the frequency domain

In the beginning of this chapter we defined a generic linear system. Now, we restrict ourselves to the discrete-time case. To be consistent with the literature, we will not write the signal as $x(t)$, but as $x[n]$, where n can assume integer values. If this system is time-invariant [43], we can write the action of $\mathcal{H}$ on some signal in the time domain as

$$y[n] = (h * x)[n], \tag{5.2}$$

where $*$ denotes the convolution operation and $h[n]$ is the *impulse response* of the system $\mathcal{H}$. If the sequence $x[n] = e^{k\omega n}$ is given as input to the considered linear system, we have that:

$$\begin{aligned} y[n] &= (h * x)[n] && (5.3)\\ &= \sum_{k=-\infty}^{+\infty} h[k]x[n-k] && (5.4)\\ &= \sum_{k=-\infty}^{+\infty} h[k]e^{j\omega(n-k)} && (5.5)\\ &= \underbrace{e^{j\omega n}}_{x[n]} \underbrace{\sum_{k=-\infty}^{+\infty} h[k]e^{-j\omega k}}_{H(e^{j\omega})}, && (5.6) \end{aligned}$$

where we define $H(e^{j\omega})$ as the *frequency response* of the filter $\mathcal{H}$. Intuitively, this quantity measures how the filter responds to the input of a signal $x[n]$ containing a single frequency.

Unfortunately, this situation is impossible when dealing with real signals: all the considered signals have bounded support, implying that the support of their Fourier transform is unbounded. However, the majority of signals we deal with contains some frequencies that are much more prominent than others[6]. So we can bury our heads in the sand for a moment, ignore the mathematical technicalities and approximate a real signal by a superimposition of a finite set of frequencies $e^{j\omega_1 n}, \ldots, e^{j\omega_k n}$. For simplicity, consider again a single frequency $e^{j\omega n}$ and now let $\mathcal{H}$ be a nonlinear system. Since it is quite hard to model nonlinear systems (see next chapter for more details), consider a simplified one: each time sample $y[n]$ of the output signal is given by $f(x[n])$, where $f(.)$ is an arbitrary nonlinear function[7]. If

[6]This hypothesis will be largely explored in Part III of the thesis.

[7]This kind of nonlinear systems will be studied in more details in Part II and III of the thesis.

we expand this function in a Taylor series around the origin (supposing that it is analytic), we obtain

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \cdots + \frac{f^{(m)}(0)}{m!}x^m + \ldots. \tag{5.7}$$

Therefore, the response of this system to the input $x[n] = e^{j\omega n}$ is given by

$$y[n] = f(e^{j\omega n}) = f(0) + f'(0)e^{j\omega n} + \frac{f''(0)}{2!}e^{j2\omega n} + \cdots + \frac{f^{(m)}(0)}{m!}e^{jm\omega n} + \ldots, \tag{5.8}$$

which contains a superimposition of all frequencies that are integer multiples of ω, unless the function $f(.)$ is so well-chosen such that the coefficients of the Taylor series prevent the creation of these additional frequencies[8].

This fact provides a very interesting interpretation of nonlinearities in audio signals, usually stated simply as "nonlinear distortions *creates* new frequencies within a signal". Usually this fact is taken as the definition of a nonlinear system, which is quite useful albeit not correct from the logical viewpoint. This definition is also as general as saying that the filter does not satisfy Equation 5.1, but it provides nice interpretation and intuition about nonlinearities in the frequency domain. Indeed, there is also a measure of nonlinearity inspired by this fact, called the *total harmonic distortion* (THD). For more details, see Chapter 4 of [48].

Figure 5.1: Audio chain, from emission to reproduction.

Nonlinear distortions can be present at every stage of the chain illustrated in Figure 5.1, from the recording to the posterior reproduction by the listener, and can be divided in *intentional* and *unintentional* distortions. Examples of the intentional ones are: compression during recording, in order to improve the SNR; and guitar pedal effecs and tube amplifiers, which shape the timbre of the original instrument's sound. On the other hand, examples of unintentional nonlinear distortions could be over-compression and accidental saturation during recording; damage of the media in which the signal is stored; and reproduction outside the loudspeaker's linear range.

Thus, nonlinear distortions are not necessarily annoying to the listener, and sometimes are even essential in the process of audio recording. Since this thesis is concerned with the restoration of degraded audio signals, we will be mainly interested in "undoing" an unintentional (and then potentially annoying)[9] nonlinear

[8]When dealing with nonlinear distortions in Part II and III of this thesis we will be interested exactly in this case, in order to mitigate the nonlinearity within an audio signal.

[9]But not every unintentional nonlinear distortion is necessarily annoying! The Wikipedia page

distortion that might be present in an audio signal. For this, we first must have a way to describe nonlinear systems, subject of the next chapter.

We conclude by saying that the matter of deciding if a distortion is intentional or not, annoying or not is purely subjective and depends on the user of the proposed methods. There is this loose correspondence between unintentional and annoying nonlinearities but this is not a general rule to be followed.

of the album "In The Court of The Crimson King" [49] by the legendary progressive rock band King Crimson, reports that the stereo master recorder used during the mixing stage had incorrectly aligned heads, and this misalignment resulted in a loss of high-frequencies and introduced some nonlinear distortions in the entire album, more perceived in the song "21st Century Schizoid Man". Since the original tapes were considered lost, even after some edition of the audio this problem was present in the first editions of the vinyl, and it was solved only around 2003, when the original tapes were found. I was not able to find a more reliable source to this fact, like an interview of some member of the band, but this appears to be a well-known fact in the progressive rock community, as I inferred from reading topics in some forums. The version I have of this album is one of the post-2003 ones, so I am also unable to compare with the original recording.

This history illustrates an unintentional nonlinearity that was not necessarily annoying, since its artistic impact was consistent with the purpose of the song.

Chapter 6

Trying to describe the nonlinear world

Perhaps the most popular and studied way of describing nonlinear systems is the *Volterra series*, because of its flexibility to model several real-world phenomena and relatively easy description [50], even though it was not created specifically with this goal in mind. This chapter was written with the aim of introducing this tool and contextualizing in this more general scenario the models we use. It is organized as follows: Section 6.1 briefly explains the birth of the Volterra series in the context of integral equations and explains its link with nonlinear systems in both continuous- and discrete-time cases; in Section 6.2 we restrict the general definition of a Volterra model to some particular cases of interest, parameterized by a finite number of coefficients; Section 6.3 presents the *block-oriented models*, an important class of nonlinear systems that are special cases of Volterra models; and finally, Section 6.4 reports some early applications of Volterra models in audio processing, not only for restoration, but also for modeling and recreating an intentional nonlinear distortion.

6.1 Volterra Series

Vito Volterra was an Italian mathematician and physicist who, among other things, is known for his contributions to the theory of integral equations [51]. This kind of equations arise quite naturally when studying, for example, the existence and uniqueness of solutions of ordinary differential equations[1].

[1]Of course, integral equations should not be taken as mere ancillary objects to the study of differential equations. Indeed, some physical laws like the Maxwell equations can be written both in the differential of integral forms, each one possessing its own advantages and beauty. There is also the field of study of *integro-differential equations*, very important to Statistical Mechanics and electrical circuit analysis [52].

For example, consider the differential equation below:

$$\begin{cases} \dfrac{dy}{dx} = F(x, y(x)); \\ y(x_0) = y_0, \end{cases} \tag{6.1}$$

for $x \in (x_0 - \varepsilon, x_0 + \varepsilon)$. If we integrate this equation from x_0 to x, we obtain:

$$y(x) = y_0 + \int_{x_0}^{x} F(s, y(s))\ ds. \tag{6.2}$$

If one defines $\phi_0(x) = y_0$ and considers the sequence

$$\phi_n(x) = y_0 + \int_{x_0}^{x} F(s, \phi_{n-1}(s))\ ds \tag{6.3}$$

for $n \geq 1$, it can be shown that under some mild conditions on F [53][2] that sequence $\phi_n(t)$ defined above converges to the unique solution of Equation 6.1, as $n \to \infty$. This is the well-known *Picard-Lindelöf theorem*.

If one considers the slightly more general integral equation given by

$$\phi(x) = f(x) + \int_{0}^{x} F(x, y, \phi(y))\ dy, \tag{6.4}$$

where the functions $f(x)$ and $F(x, y, z)$ are known and $\phi(x)$ is the unknown function, it can be shown [54] that the iterative sequence defined by

$$\phi_n(x) = f(x) + \int_{0}^{x} F(x, y, \phi_{n-1}(y))\ dy \tag{6.5}$$

for $n \geq 1$ and $\phi_0(x) = f(x)$ converges to the desired solution $\phi(x)$ as $n \to \infty$.

And finally considering the special case where $F(x, y, \phi(y)) = \lambda K(x, y)\phi(y)$, where K satisfies $K(x, y) = 0$ if $y > x$, called the *Volterra equation of the second kind* and very common in Physics [52], the procedure considered above leads to the following series expansion [54]:

$$\phi(x) = f(x) + \sum_{n=1}^{\infty} \lambda^n \int_{0}^{x} K_n(x, y) f(y)\ dy, \tag{6.6}$$

where $K_n(x, y)$ is called the *iterated kernel* defined by $K_1(x, y) = K(x, y)$ and for $n \geq 2$

$$K_n(x, y) = \int_{0}^{x} K(x, z) K_{n-1}(z, y)\ dz. \tag{6.7}$$

Equation 6.6 is the link between integral equations and nonlinear systems. This

[2] It is enough to consider F uniformly Lipschitz continuous on the second variable.

link appears for the first time in history in a wartime report from 1942 [55] by Norbert Wiener, and it was made public after the war. A more detailed exposition of this report can be found in [56]. We briefly transcribe here a derivation of this link below, following Chapter 1 of [50].

6.1.1 Relation of Volterra series with nonlinear systems

In the continuous-time scenario, the definition of a linear system in Equation 5.1 implies that the output $y(.)$ is given by the convolution between the input $u(.)$ and the impulse response of the system, denoted by $h(.)$:

$$y(t) = \int_{-\infty}^{+\infty} h(s)u(t-s)\ ds = \int_{-\infty}^{+\infty} h(t-s)u(s)\ ds. \tag{6.8}$$

If one considers also that the system is *causal* and *relaxed*, this implies that $h(t) = 0$, for $t < 0$, $u(t) = 0$ for $t \leq 0$, $y(0) = 0$ and the convolution in Equation 6.8 above reduces to

$$y(t) = \int_{0}^{t} h(s)u(t-s)\ ds = \int_{0}^{t} h(t-s)u(s)\ ds. \tag{6.9}$$

Now if one defines $K(x,y) = K_1(x,y) = h(x-y)u(x)$ and considers $f(x) = 1$, the Equation 6.9 is essentially the term for $n = 1$ in the series in Equation 6.6. Considering high-order terms, it is possible to model more general relations between the input $u(.)$ and the output $y(.)$. In particular, the terms in the series for $n = 2$ and $n = 3$ are, respectively, given by:

$$\int_{0}^{x}\int_{0}^{x} h(x-z)h(z-y)u(z)u(y)\ dzdy \tag{6.10}$$

and

$$\int_{0}^{x}\int_{0}^{x}\int_{0}^{x} h(x-z)h(z-s)h(s-y)u(z)u(s)u(y)\ dzdsdy, \tag{6.11}$$

which can be clearly seen as generalizations of the usual convolution in Equation 6.9.

Now it is no longer necessary to restrict ourselves to any framework imposed by the Volterra equation of second kind, and we are free to define the *continuous-time Volterra model* to a nonlinear system, relating its input $u(.)$ and its output $y(.)$ according to the infinite series

$$y(t) = y_0 + y_1(t) + y_2(t) + y_3(t) + \ldots, \tag{6.12}$$

where y_0 is a constant, $y_1(t)$ is the first-order term defined in Equation 6.8, and for

$n \geq 2$, $y_n(t)$ is defined by

$$y_n(t) = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_n(s_1, \dots, s_n) u(t - s_1) \dots u(t - s_n)\, ds_1 \dots ds_n, \tag{6.13}$$

where each h_n is a function with a similar role as the impulse response of the system.

The more important point we must learn with this discussion is that since the Volterra model defined above is inspired by a method for solving integral equations that are ubiquitous in Physics to describe inherently nonlinear real world phenomena, it is a good candidate to model generic nonlinear systems.

6.1.2 Discrete-time Volterra series model

Now it is quite simple to define a model relating the input and output of a nonlinear system, inspired by the discussion above. If $u[.]$ and $y[.]$ are, respectively, the input and output of a nonlinear system, we can suppose that the relation between them is given by

$$y[k] = y_0 + y_1[k] + y_2[k] + y_3[k] + \dots, \tag{6.14}$$

where the first term is a constant, the second one is the usual convolution between the input and the impulse response

$$y_1[k] = \sum_{l=-\infty}^{+\infty} h_1[l] u[k - l], \tag{6.15}$$

and for $n \geq 2$, $y_n[k]$ is given by

$$y_n[k] = \sum_{l_1=-\infty}^{+\infty} \dots \sum_{l_n=-\infty}^{+\infty} h_n[l_1, \dots, l_n] u[k - l_1] \dots u[k - l_n], \tag{6.16}$$

where again each h_n is a function with a similar role as the impulse response of the system[3].

This is the *discrete-time Volterra model* of a nonlinear system, and in the next section we point out some of its applications in audio processing. The main reference for this Chapter [50][4] presents a lot of applications to diverse fields of science.

[3] We will not consider convergence issues for now, since the great generality of the model turns very difficult to assert convergence in the general case. In next sections we will restrict the model to specific cases where this point will be enlightened.

[4] I thank very much Professor Amit Bhaya for recommending me this great book!

6.2 Particular classes of Volterra models

Equations 6.14, 6.15 and 6.16, which define the discrete-time Volterra model, are very general since all implied sums are infinite. First we consider the finite-dimensional models, defined by

$$y[k] = y_0 + \sum_{n=1}^{N} y_n^M[k], \text{ where} \tag{6.17}$$
$$y_n^M[k] = \sum_{l_1=0}^{M} \dots \sum_{l_N=0}^{M} h_n[l_1, \dots, l_n] u[k-l_1] \dots u[k-l_n].$$

Note that it is reasonable to call this class of models *causal*, since the time sample $y[k]$ is determined only by the "present" $u[k]$ and the "past" $u[k-l]$, for $l > 0$.

The class of model defined by Equations 6.17 is denoted by $\mathcal{V}_{(N,M)}$, where the first parameter N is called the *nonlinear degree* and the second one M the *dynamic order* of the model. Indeed, if we take $y_0 = 0$ and $N = 1$, we obtain the description of a linear FIR filter, since Equation 6.17 reduces to

$$y[k] = \sum_{l_1=0}^{M} h_1[l_1] u[k-l_1]. \tag{6.18}$$

On the other hand, considering $M = 0$, the relation between signals $u[.]$ and $y[.]$ is a static memoryless polynomial nonlinearity of degree N:

$$y[k] = y_0 + \sum_{n=1}^{N} h_n[0, \dots, 0] u[k]^n. \tag{6.19}$$

Nothing prevents us from considering the infinite dimensional limiting cases $\mathcal{V}_{(\infty,M)}$, $\mathcal{V}_{(N,\infty)}$ and $\mathcal{V}_{(\infty,\infty)}$, and these cases will be indeed studied later.

6.3 Block-oriented models

Despite being finite-dimensional, the class $\mathcal{V}_{(N,M)}$ carries a problem: in order to uniquely specify a model within it, a large number of coefficients is necessary. To understand this, note that each function $h_n[l_1, \dots, l_n]$ is specified by $(M+1)^n$ real numbers, since each l_i can assume the values from 0 to M ($M+1$ possibilities) and there are n such l_i's. Since n varies from 1 to N, the total number of coefficients necessary to specify a model is given by

$$1 + \sum_{n=1}^{N} (M+1)^n = \sum_{n=0}^{N} (M+1)^n = \frac{(M+1)^{N+1} - 1}{M} \approx M^N. \tag{6.20}$$

When one needs to identify such a system, the potentially great amount of coefficients can lead to several problems. This motivates us to restrict even more the class $\mathcal{V}_{(N,M)}$. Recall that the simplest finite-dimensional Volterra models are the classes $\mathcal{V}_{(1,M)}$ with $y_0 = 0$ and $\mathcal{V}_{(N,0)}$, containing the FIR filters and the static memoryless polynomial nonlinearities, respectively. If we combine these linear filters and static nonlinearities in series and/or parallel, we are considering the *block-oriented* models. This name is motivated by the fact that these models are graphically represented by a block diagram, each block containing the filter or the static memoryless nonlinearity. In this thesis, the nonlinear distortions studied will be modeled in this way. For now, we just consider the theoretical relationship between block-oriented models and the classes $\mathcal{V}_{(N,M)}$, and in the next sections we will present three particular block-oriented models.

6.3.1 Hammerstein model

This model consists of a single static memoryless nonlinearity $f(.)$ followed by a linear filter with transfer function $B(z)$, as illustrated in Figure 6.1. For the moment we consider $f(.)$ as a polynomial of degree N and the filter as an FIR one of order M, leading to the class $\mathcal{H}_{(N,M)}$ of *finite-dimensional Hammerstein models*.

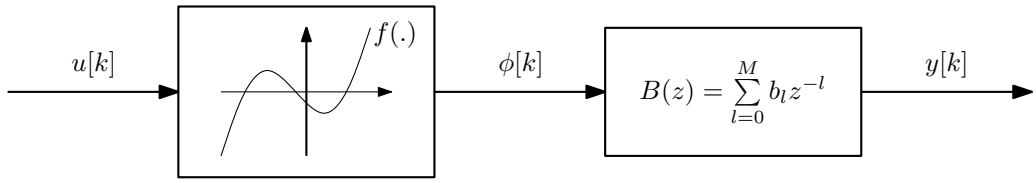

Figure 6.1: Hammerstein model.

We now prove that $\mathcal{H}_{(N,M)} \subset \mathcal{V}_{(N,M)}$. Let the polynomial $f(.)$ be given by $f(u) = \sum_{n=0}^{N} a_n u^n$, the transfer function $B(z) = \sum_{l=0}^{M} b_l z^{-l}$ and as indicated in Figure 6.1, we denote the intermediate output of the static nonlinearity by $\phi[k]$. Note that

$$\phi[k] = f(u[k]) = \sum_{n=0}^{N} a_n u[k]^n, \tag{6.21}$$

and the overall output of the model is given by

$$y[k] = \sum_{l=0}^{M} b_l \phi[k-l]. \tag{6.22}$$

These two equations can be combined to generate a single formula, like the one

given by Equation 6.17. After substituting and rearranging the terms, we have that:

$$y[k] = \underbrace{a_0 \sum_{l=0}^{M} b_l}_{y_0} + \sum_{n=1}^{N} \underbrace{\sum_{l=0}^{M} a_n b_l u[k-l]^n}_{y_n^M[k]}. \tag{6.23}$$

By examining the term $y_n^M[k]$ above, we conclude that

$$h_n[l_1, \ldots, l_n] = \begin{cases} a_n b_l, & \text{if } l_1 = \cdots = l_n = l; \\ 0, & \text{otherwise.} \end{cases} \tag{6.24}$$

Because of that, we can view the Hammerstein models as *diagonal Volterra models*, since only the "diagonal" terms in each function h_n can be different from zero. But note that the converse is not true: not every diagonal Volterra model is a Hammerstein one, since the diagonal terms in the latter case is proportional to the impulse response of the linear filter $B(z)$, as is clearly seen in Equation 6.24, and this imposes an additional constraint to the model.

Since the orders of the polynomial nonlinearity and of the linear filter are N and M, respectively, a Hammerstein model is uniquely determined by $N + M + 2$ coefficients, a number much smaller than M^N, the approximate number of coefficients necessary to specify a generic Volterra model in $\mathcal{V}_{(N,M)}$.

6.3.1.1 More general linear filters

The hypothesis of an FIR filter in the Hammerstein model is quite restrictive, and if one wants to use a filter with a more general impulse response, the finite-dimensional Hammerstein and Volterra models are not enough. If the desired linear filter is causal and stable, let us denote its impulse response by b_l, for $l \geq 0$. Now, Equation 6.23 can be adapted to this case by considering the limiting case $M \to \infty$:

$$y[k] = \underbrace{a_0 \sum_{l=0}^{\infty} b_l}_{y_0} + \sum_{n=1}^{N} \underbrace{\sum_{l=0}^{\infty} a_n b_l u[k-l]^n}_{y_n^M[k]}. \tag{6.25}$$

Note that this representation is only valid if the series above converge. A sufficient condition for this convergence is the BIBO stability of the linear filter in the model. In fact, recall that for linear and time-invariant filters the BIBO stability is

equivalent to its impulse response being absolutely summable[5] [43]:

$$\sum_{l=0}^{\infty} |b_l| < \infty. \tag{6.26}$$

This implies immediately that the first sum in Equation 6.25, given by a multiple of $\sum_{l=0}^{\infty} b_l$, converges. Now, for each fixed $n = 1, \ldots, N$, consider that the input to the linear filter is the sequence defined by $a_n u[k]^n$, for $k \in \mathbb{Z}$, which is bounded if the sequence $u[k]$ also is bounded. Then each term of the second sum in Equation 6.25, given by $\sum_{l=0}^{\infty} a_n b_l u[k-l]^n$, is just the filtering of this sequence by the linear filter. This sum exists and is finite for any $k \in \mathbb{Z}$, since we are supposing the linear filter to be BIBO stable.

This particular class of Hammerstein models is denoted by $\mathcal{H}_{(N,\infty)}$, and it is clearly a subset of the class $\mathcal{V}_{(N,\infty)}$. In its most general form, the class $\mathcal{H}_{(N,\infty)}$ is infinite-dimensional, but we will be interested in some finite-dimensional sub-classes. For example, if the linear filter is an all-pole one and causal, its impulse response is determined by the finite number of coefficients in the denominator of its transfer function, and the model is then determined by a finite number of coefficients. But since the filter is not FIR anymore, we need all the terms in the series in Equation 6.25 to fully represent this model in the framework we constructed here.

6.3.1.2 More general nonlinear distortions

If one desires to use a more general nonlinear distortion, a natural extension of the cases presented here is to consider analytical nonlinear distortions, given by its Taylor series centered in zero

$$f(u) = \sum_{n=0}^{\infty} a_n u^n, \tag{6.27}$$

provided the series converges at least in a small interval. The overall output of the model is given now by

$$y[k] = \underbrace{a_0 \sum_{l=0}^{M} b_l}_{y_0} + \sum_{n=1}^{\infty} \underbrace{\sum_{l=0}^{M} a_n b_l u[k-l]^n}_{y_n^M[k]}. \tag{6.28}$$

Note that the convergence condition in the above series is that all the terms of

[5] The actual definition of a BIBO-stable system is: if the input is a sequence in ℓ_∞, its respective output is also in ℓ_∞. The acronym BIBO means *bounded input bounded output*. For more details, see [43].

the input sequence $u[.]$ belong to the convergence interval of the Taylor series. This gives rise to the class $\mathcal{H}_{(\infty,M)}$.

6.3.1.3 More general linear filters and nonlinear distortions

Of course we can consider both analytical nonlinear distortion and an IIR filter in the same model, giving rise to the even more general relationship between input $u[.]$ and output $y[.]$:

$$y[k] = \underbrace{a_0 \sum_{l=0}^{\infty} b_l}_{y_0} + \sum_{n=1}^{\infty} \underbrace{\sum_{l=0}^{\infty} a_n b_l u[k-l]^n}_{y_n^M[k]} . \tag{6.29}$$

These models generate the class $\mathcal{H}_{(\infty,\infty)}$. Sufficient conditions to the convergence of the above series are the intersection of the conditions to assert the convergence of the series in Equations 6.25 and 6.28, namely the BIBO-stability of the linear filter and every term of the input sequence $u[.]$ belonging to the convergence interval of the Taylor series of the analytical nonlinear distortion [50].

6.3.2 Wiener model

This model consists of the same components of the Hammerstein one, but in reverse order: first the linear filter whose transfer function is $B(z)$ acts on the input $u[.]$, followed by the static memoryless nonlinearity denoted by $f(.)$, as indicated in Figure 6.2 below:

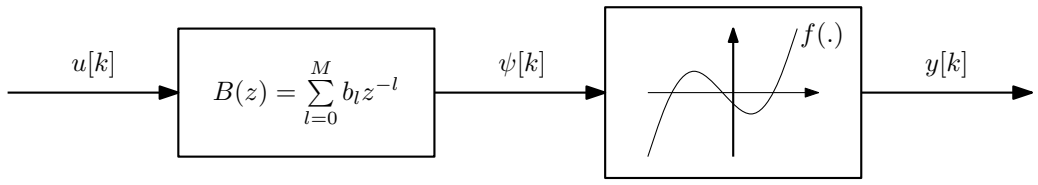

Figure 6.2: Wiener model.

Analogously to the Hammerstein model, if we consider $f(.)$ as a polynomial of degree N and the filter as an FIR one of order M, we obtain the class of the *finite-dimensional Wiener models*, denoted by $\mathcal{W}_{(N,M)}$. It is also possible to show that $\mathcal{W}_{(N,M)} \subset \mathcal{V}_{(N,M)}$, but this demonstration is very tedious and will be omitted here.

6.3.3 Wiener-Hammerstein model

Finally, the last model of interest for us is the *Wiener-Hammerstein model*, consisting of a linear filter, a static memoryless nonlinear distortion and another linear filter, connected in cascade, as shown in Figure 6.3 below:

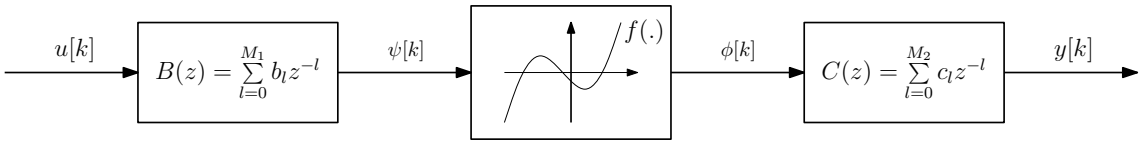

Figure 6.3: Wiener-Hammerstein model.

Considering $f(.)$ as a polynomial of degree N and both filters as FIR of orders M_1 and M_2 respectively, as indicated in Figure 6.3, we have the class of *finite-dimensional Wiener-Hammerstein models*, denoted by $\mathcal{WH}_{(N,M_1,M_2)}$, that is contained in the class $\mathcal{V}_{(N,M)}$, for some value of M, but again the details will be omitted.

6.4 Applications of nonlinear models in audio processing

The nonlinear distortions in audio signals can be divided in intentional and unintentional ones, as we have seen in Chapter 5. Therefore, early works in this field are divided mainly in two categories: *simulation* and *compensation* of nonlinear distortions, briefly explained here: the goal of the simulation of a nonlinear distortion is to identify an intentional effect in order to reproduce it in another situation. For example, one could try to infer the parameters of a Volterra model associated with an historical guitar valve amplifier, and apply this effect into a new sound; on the other hand, the restoration procedures are concerned with the identification of the nonlinearity and its inversion, in order to mitigate it and recover the original sound.

Although philosophically different, the technical procedures are very similar, since both requires the identification of some model for the nonlinearity. Naturally, the Volterra model, or some block-oriented model, was widely used in this task, and here we list (non extensively) some of the previous work in this field.

The earliest proposed technique for the restoration of nonlinear distorted audio appeared in 1982 [57], and it did not use the Volterra model. Instead, it was based on "equalizing" the histogram of speech signals, in order to recover its original distribution. The works [58–60] deal with more specific sources of nonlinearity: magnetic recordings, horn loudspeakers and distortions usually present in old motion-pictures, respectively.

A more general approach to the restoration procedure is the Ph.D. thesis [61], where a coupling of AR and Volterra models is performed in order to model the entire audio signals. Associated publications expanding the work from this thesis

are [62, 63].

Finally, some applications of Volterra models for simulation of nonlinear devices in audio can be found in [64, 65].

Chapter 7

Invertible memoryless nonlinear distortion

Now that we have discussed some ways of modeling nonlinearities, we apply them to the problem of audio restoration. In this chapter we tackle the case of memoryless nonlinear distortions, with the additional assumption of invertibility. In Section 7.1 we present the model used to describe the nonlinear distortion; Section 7.2 presents a particular model where the inverse of the static nonlinearity is approximated by its Taylor series centered at zero, leading to a model in class $\mathcal{V}_{(\infty,0)}$; in Section 7.3 we approximate the static nonlinearity by a piecewise linear function, which does not comply to any class of Volterra models presented; we close the chapter with Section 7.4, where the estimation procedure for the undistorted signal is briefly discussed.

Of the references given at the end of the last chapter, none is tailored to treat generic memoryless distortions in audio signals. However, some of them are specific to some situation, like nonlinearities present in old motion-pictures [60] or caused by horn loudspeakers [59]. Despite this, the problem of memoryless nonlinear distortion was tackled in other contexts [4, 66], both supposing sparsity of the underlying original signal.

The contributions of the thesis to this topic will be stated at each section where a solution to the problem is given.

7.1 Describing the model

The relation between the original audio signal $\mathbf{x}$ and the distorted one $\mathbf{y}$ is shown in Figure 7.1 below:

We only have access to a single copy of the distorted signal, and we wish to recover the original one. Naturally we need some assumptions about the undistorted signal in order to recover it; otherwise the problem is ill-posed. As argued in Chapter

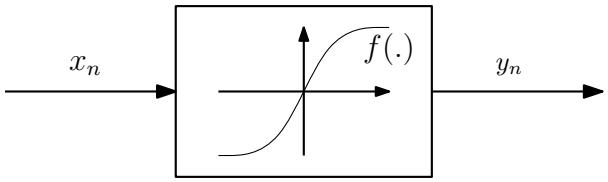

Figure 7.1: Non-linear memoryless distortion model.

4, the AR model is convenient to describe audio signals, and it is our choice here. Short excerpts of the degraded signal are described by the diagram in Figure 7.2, where e_n is the Gaussian excitation sequence and $A(z)$ is the corresponding all-pole filter of the AR model, as discussed in Section 4.3.

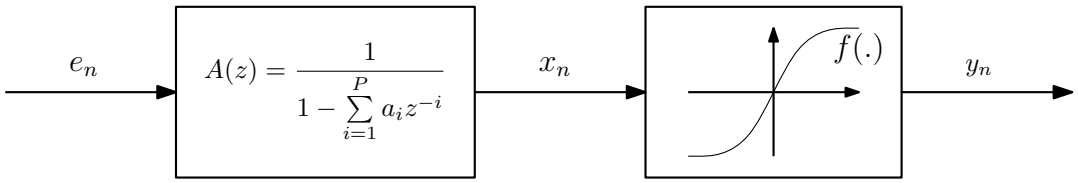

Figure 7.2: Complete non-linear memoryless distortion model.

Actually, this is the Wiener model, as described in Section 6.3.2. Since we are mainly interested in estimating and inverting only the distortion, it is more convenient to focus on the memoryless nonlinear part, and treat the parameters of the AR model as *nuisance*[1]. Since we do not have access to the input of this model (this is exactly what we want to recover), this is usually called a *blind identification problem.*

7.2 Polynomial approximation to the inverse of the nonlinear distortion

In this section, we present a Bayesian solution when the inverse of the nonlinear distortion is approximated by its Taylor series centered at zero. This solution was firstly proposed in [45], and the contributions of the thesis to the topic were some modifications on the proposal distribution for sampling the coefficients of the polynomial, and additional tests with real signals corrupted by artificial distortions, both polynomial and more general ones.

[1]This is a recurrent term in Bayesian Statistics, representing a parameter whose estimation is not the main goal, but which must be taken into account to estimate the desired parameters.

7.2.1 Description of the approximation

We impose additional assumptions over the nonlinearity $f(.)$, in order to enable its estimation and posterior inversion: anti-symmetric, invertible, and with $f'(0) = 1$. The unitary derivative around the origin may appear quite mysterious here, but it acts like a "regularization condition": since the problem is inherently ill-posed, several solutions would exist without some additional condition. In particular, the recovered signal can have arbitrarily high amplitude and the coefficients of $f(.)$ can be chosen in order to cancel out this high amplitude when computing the distorted signal. The condition $f'(0) = 1$ then says that the original and distorted signal are "similar" when their amplitudes are close to zero, preventing the amplitude of the recovered signal from being arbitrarily high.

As it will be clear in the next paragraphs, it is more convenient to write the formulas with the inverse of $f(.)$, which we will call $g(.)$ instead of $f^{-1}(.)$, to avoid cumbersome notations. In terms of the function g, the aforementioned hypotheses can be written as:

- $g(y) = -g(-y)$, for all $y \in \mathbb{R}$;
- $g'(y) > 0$, for all $y \in \mathbb{R}$;
- $g'(0) = 1$.

If $g(.)$ is approximated by its Taylor series and one wishes to incorporate the above restrictions[2] on the approximation, we can write it as:

$$x = g(y) = y + m_1 y^3 + m_2 y^5 + \cdots + m_M y^{2M+1}. \tag{7.1}$$

In this first part of the work, the value of M, related to the order of the polynomial $g(.)$, is unknown and can be tuned by the user to obtain the desired result. This is a drawback of the method that will be addressed in future works by estimating M via a reversible Jump MCMC algorithm [27].

Since this equation is linear in the coefficients of the polynomial, it can be written in matrix form as:

$$\mathbf{x} = \mathbf{y} + \mathbf{Y}\mathbf{m}, \tag{7.2}$$

[2]Except for the invertibility, which is more difficult to assure. One possibility is to consider only positive coefficients, but this is very restrictive. We will soon see how to numerically verify this.

for a block of N time samples from signal $\mathbf{y}$, where

$$\mathbf{Y} = \begin{bmatrix} y_1^3 & y_1^5 & \dots & y_1^{2M+1} \\ y_2^3 & y_2^5 & \dots & y_2^{2M+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_N^3 & y_N^5 & \dots & y_N^{2M+1} \end{bmatrix} \tag{7.3}$$

and

$$m = [m_1 \; m_2 \; \dots \; m_M]^T. \tag{7.4}$$

This linearity in $\mathbf{m}$ will turn the procedure of estimation much easier, as we will see.

7.2.2 Computation of the likelihood

In order to estimate vector $\mathbf{m}$, we must write the likelihood function for the parameters, given the observed signal. In Chapter 4, we considered $\mathbf{x}$ as an excerpt of the original signal of length N, and split this vector into its first P time samples ($\mathbf{x}_0 = [x_1 \; \dots \; x_P]^T$) and the remaining $(N-P)$ samples ($\mathbf{x}_1 = [x_{P+1} \; \dots \; x_N]^T$). The corresponding excitation signal is denoted by $\mathbf{e} = [e_{P+1} \; \dots \; e_N]^T$, and finally the corresponding distorted signal $\mathbf{y}$ is also split into its first P time samples ($\mathbf{y}_0 = [y_1 \; \dots \; y_P]^T$) and the remaining $(N-P)$ samples ($\mathbf{y}_1 = [y_{P+1} \; \dots \; y_N]^T$). The exact likelihood of the parameters $\mathbf{a}$ and σ_e^2 is approximated by

$$\begin{aligned} p(\mathbf{x}|\mathbf{a}, \sigma_e^2) \approx p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{a}, \sigma_e^2) &= \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2}\mathbf{e}^T\mathbf{e}\right) \\ &= \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2}\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}\right) \\ &= \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2}(\mathbf{x}_1 - \mathbf{X}\mathbf{a})^T(\mathbf{x}_1 - \mathbf{X}\mathbf{a})\right), \end{aligned} \tag{7.5}$$

where matrices $\mathbf{A}$ and $\mathbf{X}$ were defined in Chapter 4. From now on, by an abuse of notation, we write an equality sign instead of the approximation sign, as discussed in Chapter 4.

The change of variables in Equation 7.1 does not have unitary Jacobian, and in order to obtain the likelihood of the parameters given the vector $\mathbf{y}$ we must use Theorem 2.4.1.

In our case, function g already defines functions s_n, for $n = P+1, \dots N$, by

$$x_n = s_n(y_1, \dots, y_N) = g(y_n) = y_n + m_1 y_n^3 + m_2 y_n^5 + \dots + m_M y_n^{2M+1}, \tag{7.6}$$

and its partial derivatives are given by

$$\frac{\partial s_n}{\partial y_l} = \begin{cases} 1 + \sum_{j=1}^{M} (2j+1) m_j y_n^{2j}, & \text{if } n = l, \\ 0 & \text{otherwise,} \end{cases} \tag{7.7}$$

leading to a diagonal Jacobian matrix, whose determinant is given by

$$\prod_{n=P+1}^{N} \left[1 + \sum_{j=1}^{M} (2j+1) m_j y_n^{2j} \right]. \tag{7.8}$$

If we define vectors

$$\mathbf{h}_n = [3y_n^2 \quad 5y_n^4 \quad \dots \quad (2M+1)y_n^{2M}]^T \tag{7.9}$$

for $n = P+1, \dots, N$, we can write the Jacobian determinant in a more compact form as

$$\prod_{n=P+1}^{N} \left[1 + \mathbf{h}_n^T \mathbf{m} \right]. \tag{7.10}$$

By using Theorem 2.4.1 in Equation 7.5, we can deduce the likelihood of the desired parameters given vector $\mathbf{y}$:

$$p(\mathbf{y}|\mathbf{m}, \mathbf{a}, \sigma_e^2) = \prod_{n=P+1}^{N} \left| 1 + \mathbf{h}_n^T \mathbf{m} \right| \times \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left[-\frac{1}{2\sigma_e^2} (\mathbf{y} + \mathbf{Y}\mathbf{m})^T \mathbf{A}^T \mathbf{A} (\mathbf{y} + \mathbf{Y}\mathbf{m}) \right]. \tag{7.11}$$

A method for estimating the parameters is maximizing the likelihood with respect to $\mathbf{m}$, $\mathbf{a}$ and σ_e^2, leading to an estimate of the parameters for which the distorted signal is the most probable one. This approach was first proposed in [45] and published with some modifications in [67], where the Gauss-Newton method [18] was used to maximize the likelihood. We explain in details only the Bayesian approach, since the contribution of the thesis to the topic was in this context.

7.2.3 Bayesian estimation of the nonlinear distortion

Some drawbacks of likelihood maximization by means of a deterministic algorithm are that it could be trapped close to a local maximum, and that it is not straightforward to deal with the constraints imposed to the coefficients of the polynomial $g(.)$, since they define a quite complicated region of $\mathbb{R}^M$. In the Bayesian context these issues are simpler to address, and we also have the possibility of treating more

than one defect in the audio signal at a time, by incorporating it on the PDF of the signal. It is important to note that this last point can be misleading, since it seems that it is straightforward to bundle all the defects that an audio signal can possess, estimate its respective parameters and restore the signal, but even slight modifications in the problem can lead to very complicated PDFs, as we will see in the case of nonlinear distortion with memory. Anyway, despite such practical difficulties, Bayesian methods provide us with this great flexibility.

By using the Bayes' Theorem, we obtain

$$p(\mathbf{m}, \mathbf{a}, \sigma_e^2|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{m}, \mathbf{a}, \sigma_e^2)p(\mathbf{m})p(\mathbf{a})p(\sigma_e^2), \tag{7.12}$$

where $p(\mathbf{m})$, $p(\mathbf{a})$ and $p(\sigma_e^2)$ are prior distributions for the respective parameters, chosen as:

- $p(\mathbf{m}) \propto \exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}^T\mathbf{m}\right)\Omega(\mathbf{m})$, where function $\Omega(\mathbf{m})$ returns 1 if g satisfies the required constraints[3] and 0 otherwise;
- $p(\mathbf{a}) \propto \exp\left(-\frac{1}{2\sigma_a^2}\mathbf{a}^T\mathbf{a}\right)\Phi(\mathbf{a})$, where function $\Phi(\mathbf{a})$ returns 1 if $\mathbf{a}$ contains the coefficients of a stable all-pole filter and 0 otherwise;
- $\sigma_e^2 \sim \text{IG}(\alpha, \beta)$, where IG denotes the Inverse Gamma distribution.

The particular choice of a truncated Gaussian and an Inverse Gamma are for mathematical convenience since they lead to simpler conditional distributions, as will be seen later. Hyper-parameters σ_m^2, σ_a^2, α and β are chosen to make the priors vague (large variance).

Now, we can write the joint posterior distribution of the parameters:

$$\begin{aligned} p(\mathbf{m}, \mathbf{a}, \sigma_e^2|\mathbf{y}) \propto & \left[\prod_{n=P+1}^{N} \left|1 + \mathbf{h}_n^T\mathbf{m}\right| \times \right. \\ & \left. \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left\{-\frac{1}{2\sigma_e^2}(\mathbf{y} + \mathbf{Y}\mathbf{m})^T\mathbf{A}^T\mathbf{A}(\mathbf{y} + \mathbf{Y}\mathbf{m})\right\}\right] \times \\ & \left[\exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}^T\mathbf{m}\right)\Omega(\mathbf{m})\right] \times \left[\exp\left(-\frac{1}{2\sigma_a^2}\mathbf{a}^T\mathbf{a}\right)\Phi(\mathbf{a})\right] \times \\ & \left[\sigma_e^{2^{-(\alpha+1)}} \exp\left(-\frac{\beta}{\sigma_e^2}\right)\right]. \end{aligned} \tag{7.13}$$

A possible estimation of the parameters based on the posterior distribution is the *maximum a posteriori*, which yields the parameters that maximize this distribution. This estimate can be interpreted as the most probable parameters that caused the distortion. If the introduction of Bayesian methods was done, among

[3]How this is done will be addressed later when we show some results of the method.

other things, in order to avoid maximizing the likelihood in Equation 7.11, it makes no sense to maximize the even more complicated posterior distribution. An alternative approach is to take samples of this distribution, and numerically estimate any quantity of interest based on those samples. In this work, we estimate the mean of the posterior distribution by simply averaging its samples, leading to an estimate of the parameters that can be used to obtain a restored signal that is as close as possible from the original one in the ℓ_2 norm.

In order to obtain samples from the posterior distributions we employ the Gibbs sampler, which generates samples from the conditional distributions in an iterative fashion:

a) Initialize values $\mathbf{m}^{(0)}$, $\mathbf{a}^{(0)}$ and $\sigma_e^{2^{(0)}}$

b) For k from 1 to N_{iter}:

   i) Sample $\sigma_e^{2^{(k)}}$ from distribution $p(\sigma_e^2|\mathbf{m}^{(k-1)}, \mathbf{a}^{(k-1)}, \mathbf{y})$

   ii) Sample $\mathbf{a}^{(k)}$ from distribution $p(\mathbf{a}|\mathbf{m}^{(k-1)}, \sigma_e^{2^{(k)}}, \mathbf{y})$

   iii) Sample $\mathbf{m}^{(k)}$ from distribution $p(\mathbf{m}|\mathbf{a}^{(k)}, \sigma_e^{2^{(k)}}, \mathbf{y})$

Note that it is not strictly necessary to initiate the variable σ_e^2 since it is the first variable to be sampled in the iterative scheme presented above. Note also that the order in which the sampling is performed does not matter [22].

We then estimate the posterior mean by the formulas:

$$\widehat{\sigma_e^2} = \frac{1}{N_{\text{iter}} - N_{\text{burn-in}}} \sum_{k=N_{\text{burn-in}}+1}^{N_{\text{iter}}} \sigma_e^{2^{(k)}}, \tag{7.14}$$

$$\widehat{\mathbf{a}} = \frac{1}{N_{\text{iter}} - N_{\text{burn-in}}} \sum_{k=N_{\text{burn-in}}+1}^{N_{\text{iter}}} \mathbf{a}^{(k)}, \tag{7.15}$$

$$\widehat{\mathbf{m}} = \frac{1}{N_{\text{iter}} - N_{\text{burn-in}}} \sum_{k=N_{\text{burn-in}}+1}^{N_{\text{iter}}} \mathbf{m}^{(k)}, \tag{7.16}$$

where $N_{\text{burn-in}}$ is the *burn-in time* of the Markov chain, as explained in Chapter 3.

To use the Gibbs sampler we must first compute the conditional distributions, which can be easily obtained by simply considering the posterior distribution in Equation 7.13 as a function of only the variable in question, ignoring the others. We now perform this procedure.

7.2.3.1 Conditional distribution of σ_e^2

Recall the formula for the joint posterior distribution of the parameters in Equation 7.13. We need to compute the quantity $p(\sigma_e^2|\mathbf{m}, \mathbf{a}, \mathbf{y})$. This conditional distribution

means that the values of $\mathbf{m}$, $\mathbf{a}$ and $\mathbf{y}$ are fixed, so the desired distribution depends only on σ_e^2, being the other terms only normalizing constants to ensure that this function in fact integrates to one. So, if we consider only the terms dependent on σ_e^2 in Equation 7.13 we obtain:

$$\begin{aligned} p(\sigma_e^2|\mathbf{m},\mathbf{a},\mathbf{y}) &\propto \left[\frac{1}{(\sigma_e^2)^{\frac{N-P}{2}}}\exp\left\{-\frac{1}{2\sigma_e^2}\mathbf{e}^T\mathbf{e}\right\}\right]\times\left[(\sigma_e^2)^{-(\alpha+1)}\exp\left(-\frac{\beta}{\sigma_e^2}\right)\right] \\ &\propto (\sigma_e^2)^{-\left(\frac{N-P}{2}+\alpha+1\right)}\exp\left(-\frac{(\mathbf{e}^T\mathbf{e}+\beta)/2}{\sigma_e^2}\right). \end{aligned} \tag{7.17}$$

Note that this last expression is the kernel of an Inverse Gamma distribution with parameters $\frac{N-P}{2}+\alpha$ and $\frac{\mathbf{e}^T\mathbf{e}+\beta}{2}$, i.e. this is the distribution of $p(\sigma_e^2|\mathbf{m},\mathbf{a},\mathbf{y})$, from which is easy to sample from via built-in functions of numerical softwares.

This procedure will be repeated along the text to obtain all conditional distributions needed to perform the restoration.

7.2.3.2 Conditional distribution of a

To compute $p(\mathbf{a}|\mathbf{m},\sigma_e^2,\mathbf{y})$ note first that the likelihood in Equation 7.11 depends on $\mathbf{a}$ in a complicated way. But recall from Chapter 4 and Equation 7.5 that this dependence can be rewritten in order to be quadratic in $\mathbf{a}$. Using this in the posterior distribution we obtain that:

$$\begin{aligned} p(\mathbf{a}|\mathbf{m},\sigma_e^2,\mathbf{y}) &\propto \left[\exp\left\{-\frac{1}{2\sigma_e^2}(\mathbf{x}_1-\mathbf{X}\mathbf{a})^T(\mathbf{x}_1-\mathbf{X}\mathbf{a})\right\}\right]\times\left[\exp\left(-\frac{1}{2\sigma_a^2}\mathbf{a}^T\mathbf{a}\right)\Phi(\mathbf{a})\right] \\ &= \exp\left\{-\frac{1}{2\sigma_e^2}(\mathbf{x}_1-\mathbf{X}\mathbf{a})^T(\mathbf{x}_1-\mathbf{X}\mathbf{a})-\frac{1}{2\sigma_a^2}\mathbf{a}^T\mathbf{a}\right\}\Phi(\mathbf{a}), \end{aligned} \tag{7.18}$$

which can be rearranged to be the kernel of a Gaussian distribution, restricted to the support of the function $\Phi(.)$. Its covariance matrix is $\mathbf{C}_a = \left(\frac{\mathbf{X}^T\mathbf{X}}{\sigma_e^2}-\frac{\mathbf{I}_P}{\sigma_a^2}\right)^{-1}$ and its mean is $\boldsymbol{\mu}_a = \frac{1}{\sigma_e^2}\mathbf{C}_a\mathbf{X}^T\mathbf{x}_1 = \frac{1}{\sigma_e^2}\left(\frac{\mathbf{X}^T\mathbf{X}}{\sigma_e^2}-\frac{\mathbf{I}_P}{\sigma_a^2}\right)\mathbf{X}^T\mathbf{x}_1$.

7.2.3.3 Conditional distribution of m

Finally, to compute the conditional of $\mathbf{m}$, note that the Jacobian determinant in the posteriori distribution in Equation 7.13 makes things quite complicated, since the

distribution will not belong to any well known family:

$$p(\mathbf{m}|\mathbf{a},\sigma_e^2,\mathbf{y}) \propto \left[\prod_{n=P+1}^{N} \left|1+\mathbf{h}_n^T\mathbf{m}\right| \times \exp\left\{-\frac{1}{2\sigma_e^2}(\mathbf{y}+\mathbf{Y}\mathbf{m})^T\mathbf{A}^T\mathbf{A}(\mathbf{y}+\mathbf{Y}\mathbf{m})\right\}\right] \times \left[\exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}^T\mathbf{m}\right)\Omega(\mathbf{m})\right]. \tag{7.19}$$

Note that the terms inside the exponentials are quadratic in $\mathbf{m}$, suggesting that the conditional distribution is "almost" a truncated Gaussian, only modified by the Jacobian determinant.

In order to sample from this distribution, we perform a Metropolis-Hastings step inside the Gibbs sampler, leading to the Metropolis within Gibbs algorithm previously discussed in Chapter 3: at each step k of the algorithm, instead of directly sampling $\mathbf{m}^{(k)}$ from $p(\mathbf{m}|\mathbf{a},\sigma_e^2,\mathbf{y})$, we sample $\mathbf{m}^*$ from a proposal distribution $q(\mathbf{m}|\mathbf{m}^{(k-1)})$, that may in principle be dependent on the previous sample, and decide if it is "good" enough to be considered a genuine sample from $p(\mathbf{m}|\mathbf{a},\sigma_e^2,\mathbf{y})$ by tossing a coin that comes up head with probability

$$\alpha(\mathbf{m}^{(k-1)},\mathbf{m}^*) = \min\left(1, \frac{p(\mathbf{m}^*|\mathbf{a}^{(k-1)},\sigma_e^{2(k-1)},\mathbf{y})q(\mathbf{m}^{(k-1)}|\mathbf{m}^*)}{p(\mathbf{m}^{(k-1)}|\mathbf{a}^{(k-1)},\sigma_e^{2(k-1)},\mathbf{y})q(\mathbf{m}^*|\mathbf{m}^{(k-1)})}\right), \tag{7.20}$$

this sample being accepted if head is obtained and rejected otherwise.

The proposal distribution $q(\mathbf{m}|\mathbf{m}^{(k-1)})$ will be obtained by means of the *Laplace approximation*, first proposed by Laplace in 1774, in a work whose translation can be found in [68] by S. M. Stigler[4], a notable researcher of the History of Statistics [9].

7.2.3.3.1 Interlude: Laplace approximation

The brief explanation we give here is closely related to the one found in [69], and we transcribe it here for completeness reasons.

Consider a random vector in $\mathbb{R}^N$ for which the PDF is given by $p(\mathbf{z}) = h(\mathbf{z})/Z$, where Z is the (perhaps unknown) normalizing constant of the PDF. If $\mathbf{z}_0$ is a maximum point of $h(.)$, we can write down the following approximation:

$$\begin{aligned}\ln h(\mathbf{z}) &\approx \ln h(\mathbf{z}_0) + \frac{1}{h(\mathbf{z}_0)}\nabla h(\mathbf{z}_0) - \frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^T\mathbf{H}(\mathbf{z}-\mathbf{z}_0) \\ &= \ln h(\mathbf{z}_0) - \frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^T\mathbf{H}(\mathbf{z}-\mathbf{z}_0),\end{aligned} \tag{7.21}$$

[4]Stephen Mack Stigler, Minneapolis, August 10, 1941.

where $\mathbf{H}$ is (minus) the Hessian matrix of $\ln h(.)$ evaluated at $\mathbf{z}_0$:

$$\mathbf{H} = -\nabla\nabla \ln h(\mathbf{z}_0). \tag{7.22}$$

Note that since $\mathbf{z}_0$ is a maximum point of $h(.)$, this matrix is non-negative.

By exponentiating both sides of Equation 7.21, we obtain:

$$h(\mathbf{z}) \approx h(\mathbf{z}_0) \exp\left\{-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^T \mathbf{H}(\mathbf{z}-\mathbf{z}_0)\right\}, \tag{7.23}$$

and finally

$$p(\mathbf{z}) \approx \frac{1}{Z} h(\mathbf{z}_0) \exp\left\{-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^T \mathbf{H}(\mathbf{z}-\mathbf{z}_0)\right\}, \tag{7.24}$$

at least for $\mathbf{z}$ sufficiently close to $\mathbf{z}_0$. One can find the constant Z if it is necessary, but we will not need this for now. The important conclusion here is that close to a maximum point, the PDF $p(.)$ can be well approximated by a Gaussian with mean $\mathbf{z}_0$ and covariance matrix $\mathbf{H}^{-1}$.

7.2.3.4 Back to the conditional distribution of m

If we consider the natural logarithm of $p(\mathbf{m}|\mathbf{a}, \sigma_e^2, \mathbf{y})$ in Equation 7.19, we have that:

$$\begin{aligned}\ln p(\mathbf{m}|\mathbf{a}, \sigma_e^2, \mathbf{y}) = &\sum_{n=P+1}^{N} \ln\left|1 + \mathbf{h}_n^T \mathbf{m}\right| \\ &- \frac{1}{2\sigma_e^2}(\mathbf{y}+\mathbf{Y}\mathbf{m})^T \mathbf{A}^T \mathbf{A}(\mathbf{y}+\mathbf{Y}\mathbf{m}) + \\ &- \frac{1}{\sigma_m^2}\mathbf{m}^T\mathbf{m} + C,\end{aligned} \tag{7.25}$$

which is only defined for $\mathbf{m}$ in the support of the function $\Omega(.)$ and the additive constant C arises from the proportionality constant in Equation 7.19.

In order to apply the Laplace approximation to this distribution, it is necessary to find its maxima. This function is not straightforwardly maximized by analytical means, so we compute the maxima by employing the Newton method [18], initialized at the last sampled and accepted value of $\mathbf{m}$. If we denote the obtained maximum point of $\ln p(\mathbf{m}|\mathbf{a}, \sigma_e^2, \mathbf{y})$ by $\mathbf{m}_{\max}$ and by $\mathbf{H}$ the negative-Hessian matrix computed at this maximum, the proposal distribution in the Metropolis step will be a Gaussian with mean $\mathbf{m}_{\max}$ and covariance matrix $\mathbf{H}^{-1}$. Both the Hessian matrix and the gradient vector of $\ln p(\mathbf{m}|\mathbf{a}, \sigma_e^2, \mathbf{y})$ have been previously computed by hand to be used in the Newton method.

7.3 Piecewise linear approximation to the nonlinear distortion

Now we present another Bayesian solution, resulted from approximating the inverse of the nonlinear distortion by a piece-wise linear function. Again this solution was first proposed in [45], and my contributions were: reformulation of the model to analyze several blocks of signal simultaneously; and further tests with real signals corrupted by piece-wise linear and more general distortions that resulted in realizing the method suffers from some limitations not previously encountered when dealing with artificial signals [45]. This solution was presented in EUSIPCO'2015 [6], and the exposition here follows the paper very closely, including more details in some critical parts.

7.3.1 Description of the approximation

In order to treat several blocks of signal, we split the original and distorted signals $\mathbf{x}$ and $\mathbf{y}$, respectively, each one consisting of N time samples, in B sub-signals of length L, denoted by $\mathbf{x}_j$ and $\mathbf{y}_j$, for $j = 1, \ldots, B$, respectively, corresponding to contiguous sections of each one.

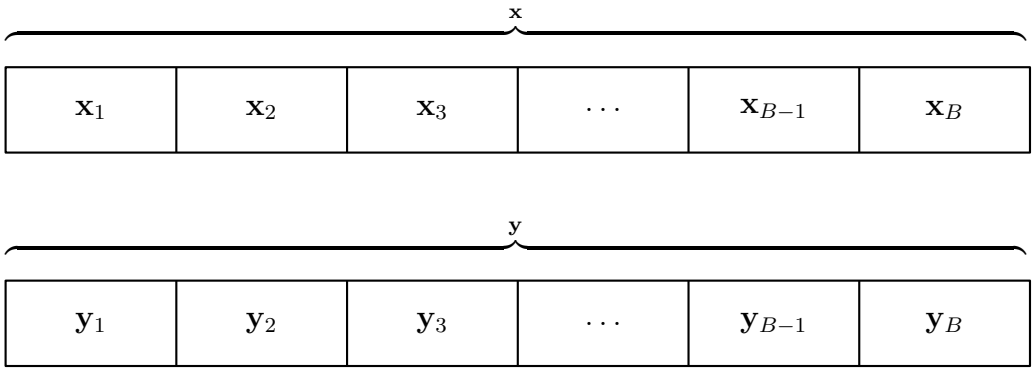

Figure 7.3: Signals $\mathbf{x}$ and $\mathbf{y}$ split in blocks.

Each block $\mathbf{x}_j$ is supposed to follow an AR model of order P, with parameters $\mathbf{a}_j$ and $\sigma^2_{e_j}$. We also suppose, for simplicity, that the blocks are independent[5]. For notational reasons, we will denote by $\mathbf{a}$ and $\boldsymbol{\sigma}^2_e$ the vectors containing the concatenations of each $\mathbf{a}_j$ and $\sigma^2_{e_j}$, respectively. For each block, there is a corresponding excitation signal, denoted by $\mathbf{e}_j$, of length $(L - P)$, and consisting of iid samples of a Gaussian distribution with zero mean and variance $\sigma^2_{e_j}$ for $j = 1, \ldots, B$, whose

[5]This hypothesis is clearly false, since the AR coefficients of one block are related with its adjacent ones because of the continuity within the signal. However, giving up of the dependency is not harmless to the method, as will be seen in the results in Chapter 9, in addition to simplifying all computations that follow.

concatenation we will denote as $\mathbf{e}$. Finally, we denote the first P time samples of each block of signals $\mathbf{x}$ and $\mathbf{y}$ by $\mathbf{x}_j^0$ and $\mathbf{y}_j^0$, and the remaining $(L-P)$ time samples by $\mathbf{x}_j^1$ and $\mathbf{y}_j^1$, respectively, for $j = 1, \ldots, B$.

As in the previous case, the distortion is supposed to be anti-symmetric, invertible and with unitary derivative at the origin; and again we describe an approximation for the inverse of the nonlinear distortion, also denoted by $g(.)$, illustrated in Figure 7.4 below. This illustration is important to better understand the following definitions.

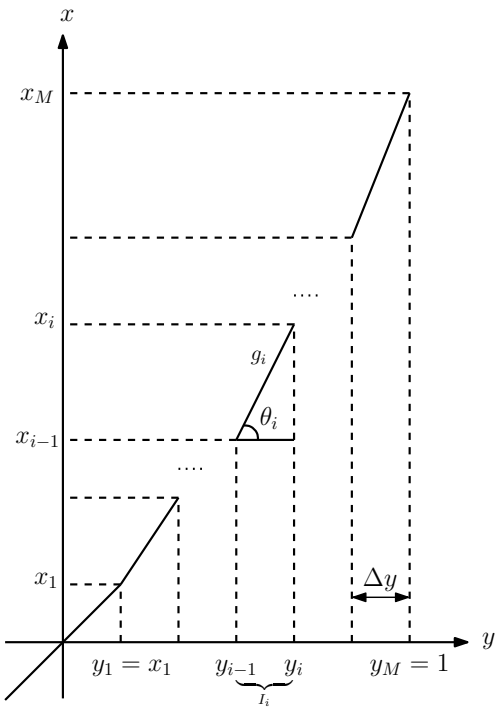

Figure 7.4: Piecewise linear approximation for $f^{-1}(.)$.

Without loss of generality, assume that all time samples of the corrupted signal $\mathbf{y}$ are in the interval $[-1, 1]$. Taking into account the hypothesis of anti-symmetry in function $g(.)$, it is necessary to specify it only for positive arguments. So we split the interval $[0, 1]$ into M contiguous intervals[6] of the same length $\Delta y = 1/M$, and denote each of these sub-intervals by $I_i = [y_{i-1}, y_i]$, for $i = 1, \ldots, M$. Note that $y_M = 1$ and define $y_0 = x_0 = 0$. Over each interval I_i, the inverse of the nonlinear distortion is approximated by an affine function with slope $m_i = \tan(\theta_i)$, denoted by $g_i(.)$. The whole approximation, computed at some point by calculating the corresponding piecewise linear approximation, is denoted by $g(.)$. To impose the required restrictions to this function, we assume that each m_i is positive in order to

[6]The same letter M was used in Section 7.2 to denote the number of polynomial coefficients, and is now used to denote the number of linear segments approximating the non-linear function. This should cause no confusion to the reader, since the two models under consideration are different.

ensure the invertibility, and $m_1 = 1$, to enforce unitary derivative around the origin.

Obviously the function $g(.)$ must also be continuous, so the intercept of each $g_i(.)$ must be chosen in order to ensure that there will be no "jumps" in the transition from one segment to another. In other words, the function $g_i(.)$ has slope $m_i = \tan(\theta_i)$ and passes through the point (y_{i-1}, x_{i-1}). This implies that it must be given by

$$x = g_i(y) = x_{i-1} + m_i(y - y_{i-1}), \tag{7.26}$$

where y is a generic point in the interval I_i and x is its image via function $g(.)$. In order to compute this, note that it is possible to obtain the value of each x_i, for $i = 1, \ldots, M$, as a function of known terms, given explicitly by the formula

$$x_i = \Delta y \sum_{j=1}^{i} m_j = \Delta y \sum_{j=1}^{i} \tan(\theta_j). \tag{7.27}$$

To prove this relation, consider $i \in \{1, \ldots, M\}$ fixed and note that

$$\begin{aligned}
\sum_{j=1}^{i} \tan(\theta_j) =& \tan(\theta_1) + \tan(\theta_2) + \cdots + \tan(\theta_{i-1}) + \tan(\theta_i) \\
=& \frac{x_1}{\Delta y} + \frac{x_2 - x_1}{\Delta y} + \cdots + \frac{x_{i-1} - x_{i-2}}{\Delta y} + \frac{x_i - x_{i-1}}{\Delta y} \\
=& \frac{x_i}{\Delta y},
\end{aligned} \tag{7.28}$$

where in the third equality all the terms in the numerator cancels out, except the last one. Passing Δy to the other side gives the required result.

Therefore, the function $g_i(.)$ can be computed in the following way:

$$x = g_i(y) = \Delta y \sum_{j=1}^{i-1} m_j + m_i(y - y_{i-1}), \tag{7.29}$$

for $y \in I_i$. Now, if $-y < 0$, there is a value of $i \in \{1, \ldots, M\}$ such that $y \in I_i$, and we can compute $g(-y)$ using its anti-symmetry property:

$$g(-y) = -g(y) = -g_i(y) = -\left[\Delta y \sum_{j=1}^{i-1} m_j + m_i(y - y_{i-1})\right]. \tag{7.30}$$

Finally, by assembling the results above, for any value of $y \in [-1, 1]$, the value of $x = g(y)$ can be computed by the formula

$$x = g(y) = \text{sign}(y) g_i(|y|) = \text{sign}(y) \left[\Delta y \sum_{j=1}^{i-1} m_j + m_i(|y| - y_{i-1})\right], \tag{7.31}$$

where sign(y) is the sign of the number y and the index i is such that $|y| \in I_i$. Note that if $i = 1$, the above formula reduces to

$$x = g(y) = \text{sign}(y)(|y| - y_0) = \text{sign}(y)|y|. \tag{7.32}$$

Since both Equations 7.31 and 7.32 are linear in each m_j, we can write it in a vector form. Define the vector $\mathbf{m}$ as

$$\mathbf{m} = [m_2 \quad m_3 \quad \dots \quad m_M]^T, \tag{7.33}$$

and suppose that $|y| \in I_i$, with $i > 1$. Then, Equation 7.31 can be rewritten as

$$x = g(y) = \text{sign}(y)\Delta y + \mathbf{r}_y^T \mathbf{m}, \tag{7.34}$$

where

$$\mathbf{r}_y = \text{sign}(y)[\Delta y \quad \dots \quad \Delta y \quad (|y| - y_{i-1}) \quad 0 \quad \dots \quad 0]^T, \tag{7.35}$$

and the element $(|y| - y_{i-1})$ occupies the $(i-1)$-th entry of vector $\mathbf{r}_y$, in order to multiply m_i in vector $\mathbf{m}$.

Note that if $i = 1$,

$$x = g(y) = \text{sign}(y)|y| + \mathbf{r}_y^T \mathbf{m}, \tag{7.36}$$

where now $\mathbf{r}_y = \mathbf{0}$.

By assembling the above results, we conclude that the original signal $\mathbf{x}$ can be written in function of the distorted one $\mathbf{y}$ and the vector $\mathbf{m}$ as

$$\mathbf{x} = \mathbf{u} + \mathbf{R}\mathbf{m}, \tag{7.37}$$

where the vector $\mathbf{u}$ contains the terms that do not multiply $\mathbf{m}$ in Equations 7.34 and 7.36 and the lines of the matrix $\mathbf{R}$ are the vectors $\mathbf{r}_{y_n}^T$ as defined above, for each value of y_n.

Since this relation between $\mathbf{x}$ and $\mathbf{m}$ is linear in $\mathbf{m}$, the estimation procedure will be very similar to the one employed in the polynomial approximation, described in Section 7.2.

7.3.2 Computation of the likelihood

With the notation and hypotheses introduced in the beginning of Section 7.3.1, the exact likelihood of the parameters $\mathbf{a}$ and $\boldsymbol{\sigma}_e^2$ are approximated as

$$\begin{aligned} p(\mathbf{x}|\mathbf{a}, \boldsymbol{\sigma}_e^2) &= \prod_{j=1}^{B} p(\mathbf{x}_j|\mathbf{a}_j, \sigma_{e_j}^2) \approx \prod_{j=1}^{B} p(\mathbf{x}_j^1|\mathbf{x}_j^0, \mathbf{a}_j, \sigma_{e_j}^2) \\ &= \prod_{j=1}^{B} \frac{1}{(2\pi\sigma_{e_j}^2)^{\frac{L-P}{2}}} \exp\left(-\frac{1}{2\sigma_{e_j}^2}\mathbf{e}_j^T\mathbf{e}_j\right) \\ &= \prod_{j=1}^{B} \frac{1}{(2\pi\sigma_{e_j}^2)^{\frac{L-P}{2}}} \exp\left(-\frac{1}{2\sigma_{e_j}^2}\mathbf{x}_j^T\mathbf{A}_j^T\mathbf{A}_j\mathbf{x}_j\right) \\ &= \prod_{j=1}^{B} \frac{1}{(2\pi\sigma_{e_j}^2)^{\frac{L-P}{2}}} \exp\left(-\frac{1}{2\sigma_{e_j}^2}(\mathbf{x}_j^1 - \mathbf{X}_j\mathbf{a}_j)^T(\mathbf{x}_j^1 - \mathbf{X}_j\mathbf{a}_j)\right), \end{aligned} \tag{7.38}$$

where matrices $\mathbf{A}_j$ and $\mathbf{X}_j$ were defined in Chapter 4, and the indexes denote the corresponding block of signal. Again we will abuse the notation and write an equality sign instead of the approximation above.

The change of variables from $\mathbf{x}$ to $\mathbf{y}$ does not have unitary Jacobian, and again we must resort to the change of variables given in Theorem 2.4.1. Following the notation introduced in the Theorem, function $g(.)$ already defines functions s_n, for $n = P+1, \ldots, N$:

$$x_n = s_n(y_1, \ldots, y_N) = g_{i_n}(y_n), \tag{7.39}$$

where i_n is the index such that $y_n \in I_{i_n}$, for $n = P+1, \ldots, N$. Their partial derivatives are given by

$$\frac{\partial s_n}{\partial y_l} = \begin{cases} m_{i_n}, & \text{if } n = l \\ 0 & \text{otherwise,} \end{cases} \tag{7.40}$$

leading again to a diagonal Jacobian matrix, whose determinant is given by

$$\prod_{i=1}^{M} m_i^{N_i}, \tag{7.41}$$

where N_i is the number of time samples over the entire signal that are affected by m_i.

Therefore, the likelihood of the parameters given vector $\mathbf{y}$ is

$$p(\mathbf{y}|\mathbf{m},\mathbf{a},\boldsymbol{\sigma}_e^2) = \prod_{i=1}^{M} m_i^{N_i} \times \prod_{j=1}^{B} \frac{1}{(2\pi\sigma_{e_j}^2)^{\frac{L-P}{2}}} \exp\left(-\frac{1}{2\sigma_{e_j}^2}(\mathbf{u}_j + \mathbf{R}_j\mathbf{m})^T\mathbf{A}_j^T\mathbf{A}_j(\mathbf{u}_j + \mathbf{R}_j\mathbf{m})\right), \quad (7.42)$$

where vector $\mathbf{u}_j$ and matrix $\mathbf{R}_j$ are defined as in Section 7.3.1, for their respective blocks.

7.3.3 Bayesian estimation of the nonlinear distortion

In order to estimate the vector $\mathbf{m}$ containing the slopes of the piecewise linear approximation to the inverse of the nonlinear distortion, we appeal again to a Bayesian framework, motivated by the same reasons as in the previous case. By using Bayes' Theorem, we obtain

$$p(\mathbf{m}, \mathbf{a}, \boldsymbol{\sigma}_e^2|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{m}, \mathbf{a}, \boldsymbol{\sigma}_e^2)p(\mathbf{m})p(\mathbf{a})p(\boldsymbol{\sigma}_e^2), \quad (7.43)$$

where $p(\mathbf{m})$, $p(\mathbf{a})$ and $p(\boldsymbol{\sigma}_e^2)$ are prior distributions for the respective parameters, which we choose as:

- $p(\mathbf{m}) \propto \exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}^T\mathbf{m}\right)\Psi(\mathbf{m})$, where function $\Psi(\mathbf{m})$ returns 1 if g satisfies the required constraints, that is $m_2, \ldots, m_M > 0$, and 0 otherwise;
- $p(\mathbf{a}) \propto \prod_{j=1}^{B} \exp\left(-\frac{1}{2\sigma_a^2}\mathbf{a}_j^T\mathbf{a}_j\right)\Phi(\mathbf{a}_j)$, where function $\Phi(\mathbf{a}_j)$ returns 1 if $\mathbf{a}_j$ contains the coefficients of a stable all-pole filter and 0 otherwise;
- $\boldsymbol{\sigma}_e^2 \sim \prod_{j=1}^{B} \mathrm{IG}(\alpha, \beta)$, where IG denotes the Inverse Gamma distribution.

The hyper-parameters σ_m^2, σ_a^2, α and β are chosen to turn the priors vague.

The joint posterior distribution of the parameters is then given by:

$$
\begin{aligned}
p(\mathbf{m},\mathbf{a},\boldsymbol{\sigma}_e^2|\mathbf{y}) \propto & \left[\prod_{i=1}^{M} m_i^{N_i} \times \right. \\
& \left. \prod_{j=1}^{B} \frac{1}{(2\pi\sigma_{e_j}^2)^{\frac{L-P}{2}}} \exp\left(-\frac{1}{2\sigma_{e_j}^2}(\mathbf{u}_j + \mathbf{R}_j\mathbf{m})^T\mathbf{A}_j^T\mathbf{A}_j(\mathbf{u}_j + \mathbf{R}_j\mathbf{m})\right)\right] \times \\
& \left[\exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}^T\mathbf{m}\right)\Psi(\mathbf{m})\right] \times \left[\prod_{j=1}^{B}\exp\left(-\frac{1}{2\sigma_a^2}\mathbf{a}_j^T\mathbf{a}_j\right)\Phi(\mathbf{a}_j)\right] \times \\
& \left[\prod_{j=1}^{B}\sigma_{e_j}^{2^{-(\alpha+1)}}\exp\left(-\frac{\beta}{\sigma_{e_j}^2}\right)\right]
\end{aligned}
\tag{7.44}
$$

To obtain samples from this distribution, we use the Gibbs sampler, implemented in the following way:

a) Initialize values $\mathbf{m}^{(0)}$, $\mathbf{a}_1^{(0)},\ldots,\mathbf{a}_B^{(0)}$ and $\sigma_{e_1}^{2^{(0)}},\ldots,\sigma_{e_B}^{2^{(0)}}$

b) For k from 1 to N_{iter}:

   i) For j from 1 to B:
   Sample $\sigma_{e_j}^{2^{(k)}}$ from distribution $p(\sigma_{e_j}^2|\mathbf{m}^{(k-1)},\mathbf{a}^{(k-1)},\boldsymbol{\sigma}_{e_{(-j)}}^{2^{(k-1)}},\mathbf{y})$

   ii) For j from 1 to B:
   Sample $\mathbf{a}_j^{(k)}$ from distribution $p(\mathbf{a}_j|\mathbf{m}^{(k-1)},\mathbf{a}_{(-j)}^{(k-1)},\boldsymbol{\sigma}_e^{2^{(k)}},\mathbf{y})$

   iii) Sample $\mathbf{m}^{(k)}$ from distribution $p(\mathbf{m}|\mathbf{a}^{(k)},\boldsymbol{\sigma}_e^{2^{(k)}},\mathbf{y})$,

where $\boldsymbol{\sigma}_{e_{(-j)}}^2$ and $\mathbf{a}_{(-j)}$ denote the vectors $\boldsymbol{\sigma}_e^2$ and $\mathbf{a}$ without the respective $j-$th components.

We then estimate the posteriori mean by the formulas:

$$
\widehat{\sigma_{e_j}^2} = \frac{1}{N_{\text{iter}} - N_{\text{burn-in}}} \sum_{k=N_{\text{burn-in}}+1}^{N_{\text{iter}}} \sigma_{e_j}^{2^{(k)}}, \text{ for } j = 1,\ldots,B, \tag{7.45}
$$

$$
\widehat{\mathbf{a}_j} = \frac{1}{N_{\text{iter}} - N_{\text{burn-in}}} \sum_{k=N_{\text{burn-in}}+1}^{N_{\text{iter}}} \mathbf{a}_j^{(k)}, \text{ for } j = 1,\ldots,B, \tag{7.46}
$$

$$
\widehat{\mathbf{m}} = \frac{1}{N_{\text{iter}} - N_{\text{burn-in}}} \sum_{k=N_{\text{burn-in}}+1}^{N_{\text{iter}}} \mathbf{m}^{(k)}, \tag{7.47}
$$

where $N_{\text{burn-in}}$ is the *burn-in time* of the Markov chain, as explained in Chapter 3.

The derivation of the conditional distributions is very similar to those detailed in Section 7.2.3, and therefore we will skip most of the already explained details.

7.3.3.1 Conditional distribution of $\sigma_{e_j}^2$

Since we are supposing independent blocks, this conditional can be obtained by assuming that it is the only block being analyzed, leading to $p(\sigma_{e_j}^2|\mathbf{m},\mathbf{a},\mathbf{y}) \sim \text{IG}\left(\frac{L-P}{2}+\alpha, \frac{\mathbf{e}_j^T\mathbf{e}_j+\beta}{2}\right)$, for $j=1,\ldots,B$.

7.3.3.2 Conditional distribution of $\mathbf{a}_j$

Similar considerations hold in this case, and we obtain that $p(\mathbf{a}_j|\mathbf{m},\boldsymbol{\sigma}_e^2,\mathbf{y})$ is a truncated Gaussian, restricted to the support of the function $\Phi(.)$, with covariance matrix $\mathbf{C}_{a_j} = \left(\frac{\mathbf{X}_j^T\mathbf{X}_j}{\sigma_{e_j}^2} - \frac{\mathbf{I}_P}{\sigma_a^2}\right)^{-1}$ and mean $\boldsymbol{\mu}_{a_j} = \frac{1}{\sigma_{e_j}^2}\mathbf{C}_{a_j}\mathbf{X}_j^T\mathbf{x}_j^1 = \frac{1}{\sigma_{e_j}^2}\left(\frac{\mathbf{X}_j^T\mathbf{X}_j}{\sigma_{e_j}^2} - \frac{\mathbf{I}_P}{\sigma_a^2}\right)\mathbf{X}_j^T\mathbf{x}_j^1$, for $j=1,\ldots,B$.

7.3.3.3 Conditional distribution of $\mathbf{m}$

Due to the geometric restrictions on $\mathbf{m}$ and the Jacobian of the transformation from $\mathbf{x}$ to $\mathbf{y}$, this conditional is not of any known family of distributions:

$$\begin{aligned} p(\mathbf{m}|\mathbf{a},\boldsymbol{\sigma}_e^2,\mathbf{y}) \propto & \left[\prod_{i=1}^{M} m_i^{N_i} \times \right. \\ & \left. \prod_{j=1}^{B} \exp\left(-\frac{1}{2\sigma_{e_j}^2}(\mathbf{u}_j+\mathbf{R}_j\mathbf{m})^T\mathbf{A}_j^T\mathbf{A}_j(\mathbf{u}_j+\mathbf{R}_j\mathbf{m})\right)\right] \times \\ & \left[\exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}^T\mathbf{m}\right)\Psi(\mathbf{m})\right]. \end{aligned} \tag{7.48}$$

In order to sample from this distribution, we employ again the Metropolis within Gibbs technique, sampling at each step k of the algorithm a vector $\mathbf{m}^*$ from a proposal distribution $q(\mathbf{m}|\mathbf{m}^{(k-1)})$, also obtained by means of the Laplace approximation of the target distribution $p(\mathbf{m}|\mathbf{a},\boldsymbol{\sigma}_e^2,\mathbf{y})$, and accepting $\mathbf{m}^*$ as a genuine sample from this distribution with probability

$$\alpha(\mathbf{m}^{(k-1)},\mathbf{m}^*) = \min\left(1, \frac{p(\mathbf{m}^*|\mathbf{a}^{(k-1)},\boldsymbol{\sigma}_e^{2(k-1)},\mathbf{y})q(\mathbf{m}^{(k-1)}|\mathbf{m}^*)}{p(\mathbf{m}^{(k-1)}|\mathbf{a}^{(k-1)},\boldsymbol{\sigma}_e^{2(k-1)},\mathbf{y})q(\mathbf{m}^*|\mathbf{m}^{(k-1)})}\right). \tag{7.49}$$

The proposal distribution is again a Gaussian, centered at the maximum value of $\ln p(\mathbf{m}|\mathbf{a},\boldsymbol{\sigma}_e^2,\mathbf{y})$, denoted by $\mathbf{m}_{\max}$ and obtained by the Newton method with initial point $\mathbf{m}^{(k-1)}$, and with covariance matrix given by the negative-Hessian of $\ln p(\mathbf{m}|\mathbf{a},\boldsymbol{\sigma}_e^2,\mathbf{y})$ computed at $\mathbf{m}_{\max}$. The gradient vector and Hessian matrix required to use the method were previously computed by hand.

7.4 A note about the estimation of the undistorted signal

During this chapter we presented strategies to estimate the vector $\mathbf{m}$, containing parameters that describe the nonlinear distortion. Denote this estimate by $\widehat{\mathbf{m}}$. In order to estimate the original signal, we compute then

$$\widehat{\mathbf{x}} = \mathbf{y} + \mathbf{Y}\widehat{\mathbf{m}} \tag{7.50}$$

when the inverse of the nonlinearity is approximated by a polynomial (Section 7.2) or

$$\widehat{\mathbf{x}} = \mathbf{u} + \mathbf{R}\widehat{\mathbf{m}} \tag{7.51}$$

when it is approximated by a piecewise linear function (Section 7.3), where matrices and vectors $\mathbf{Y}$, $\mathbf{R}$, $\mathbf{y}$, and $\mathbf{u}$ were all previously defined above. We are then computing a function of the expected value of the conditional distribution of $\mathbf{m}$.

To be more coherent with the Bayesian approach used in this chapter, it would be better to sample from the conditional distribution of $\mathbf{x}$, which when conditioned in a particular value of $\mathbf{m}$, denoted here by $\mathbf{m}_0$, possesses all the probability concentrated in a single point given by $\mathbf{y}+\mathbf{Y}\mathbf{m}_0$ or $\mathbf{u}+\mathbf{R}\mathbf{m}_0$, depending on the hypothesis over the nonlinearity. After the burn-in time, the estimate for $\mathbf{x}$ would be then the average of these samples, and each particular sample is given as a function of $\mathbf{m}$. Therefore, we will be computing the expected value of a function of the conditional distribution of $\mathbf{m}$.

Up to this point we have no evidence that these two different estimates for $\mathbf{x}$ are equal. But note that the functions of $\mathbf{m}$ considered above are linear, implying in equality in Jensen's[7] inequality ($\varphi(\mathbb{E}[X]) \leq \mathbb{E}[\varphi(X)]$ if φ is convex, with equality if and only if φ is linear), and thus we can conclude that both estimates are equivalent. Since it is convenient to compute only once the estimate for $\mathbf{x}$, we implement the first procedure.

We thank Professor Ralph Silva for warning me of this little issue in the text.

[7] Johan Ludwig William Valdemar Jensen, Nakskov, May 8, 1859 – Copehnagen, March 5, 1925.

Chapter 8

Invertible nonlinear distortions with memory

This chapter proposes a method for treating *nonlinear distortions with memory*, a defect quite more complicated than the one discussed in Chapter 7. It is organized as follows: Section 8.1 motivates the introduction of memory in the nonlinear system analyzed, with examples from real cases; next, in Section 8.2 we present a way to introduce the memory in the system via the Hammerstein model, a particular case of the Volterra series model presented in Chapter 6 and recalled in Section 8.3; in Section 8.4 we compute the likelihood of the desired parameters of the nonlinear distortion, and finally in Section 8.5 we present an algorithm based on Bayesian techniques to estimate such parameters. This solution was published in [7].

Remarkable previous works in this field include [61] and [70]. In the first one, a Volterra series model coupled with an AR model is proposed to restore nonlinearly distorted audio signals. This structure is capable to account for the memory, but as stated by the authors the large number of coefficients to estimate makes the procedure very difficult. In the second reference, the authors report that the use of a Hammerstein model is sufficiently accurate to model distortions caused by amplifiers and magnetic recorders, and also propose a method to identify such system. However, their approach assumes that the shape of the spectrum of the undistorted signal is constant over time, which is not an accurate hypothesis for audio.

The method proposed here does not require strong hypothesis on the signal spectrum, and since it is based on the Hammerstein system rather than Volterra series, it does not need to estimate a large number of coefficients.

The contributions of the thesis to the topic were some modifications in the proposal distributions and reinitialization procedure for the estimated parameters, besides suggestions about the tests performed.

8.1 Why memory?

The models presented in previous chapters were said to be *memoryless*, meaning that the distortion applied to one particular time sample of the original signal does not depend on the time samples around it. Although important to give insight about the problem and approximate some distortions created by real devices, this hypothesis is too simple for many practical applications and must therefore be improved. For example, the attack and release times in dynamic range compressors determine how fast the compressor reacts to changes in the input signal level, meaning that at a particular time sample the effect depends not only on it but also on their neighbors[1]; analog guitar pedal effects are built using circuits composed by resistors, inductors and capacitors, whose output is described by a differential equation involving the input signal, and thus depends on its integral; finally, the process of recording on a magnetic tape is inherently nonlinear and not memoryless, because of the differential equations relating the input signal and the magnetic field applied to the tape. If one is interested in accurately treating a nonlinear distortion caused by any of these examples, it is necessary to model this dependence of a single time sample of the distorted signal on more than one time sample of the original one. This dependence will be called the *memory* of the system.

8.2 How to introduce it?

Even though the signals could be analog, we will be interested on their digital counterparts, since the restoration procedure is performed on digital computers. Recall from Chapter 6 that the discrete-time Volterra series model is a very general and physically reasonable way of describing nonlinear systems. Considering its more general form given in Equations 6.14, 6.15 and 6.16, the memory is also modeled, since the $k-$th time sample of the system's output $y[k]$ could depend on the entire input signal $u[.]$. But as we noted later in that chapter, even the finite-dimensional versions of the discrete-time Volterra series model, given by Equation 6.17, are very complicated, since they require a very large number of parameters to be uniquely determined. We introduced in Section 6.3 the block-oriented models, a simplification of the general case that balances complexity and flexibility. We then presented three examples: the Hammerstein, Wiener and Wiener-Hammerstein models. Any of these could in principle be used to model devices causing nonlinear distortions in audio signals. We chose the Hammerstein model here, because of its compromise between simplicity and accuracy, and also because it was the first block-oriented

[1] If the compressor and the input signal are analog, we can say that the effect depends on the derivative of the input signal at each point.

model used in the restoration of nonlinearly distorted audio [70], with good results being reported. In that work, based on experiments performed with tube amplifier and magnetic recorder, the authors argue that the model is reasonably accurate to model real devices[2].

Intuitively, adopting the Hammerstein model is also a good approach to introduce memory in our system. Denote the original undistorted signal by $\mathbf{x}$ and the observed distorted signal by $\mathbf{z}$. As we can see in Figure 8.1, the original signal is processed by a memoryless nonlinear distortion $f(.)$, creating the intermediate signal $\mathbf{y}$. Now, a linear filter is applied to this intermediate signal, and this last block is responsible for the memory, since a linear filter is usually represented by a difference equation relating its input with its output. For example, if $B(z)$ is the transfer function of an FIR filter, each time sample z_n of the distorted signal is a linear combination of the time samples $y_n, y_{n-1}, \ldots, y_{n-Q}$ of the intermediate signal, which are in turn nonlinear functions of the time samples $x_n, x_{n-1}, \ldots, x_{n-Q}$.

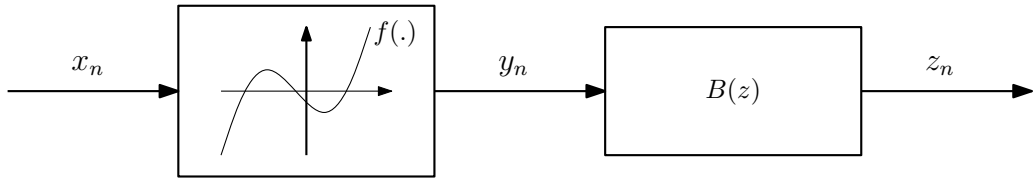

Figure 8.1: Hammerstein model.

8.3 Describing the model

As we convinced ourselves in the last section, the Hammerstein model is a good candidate to introduce memory in the nonlinear system. We now detail some additional assumptions made in the context of audio restoration.

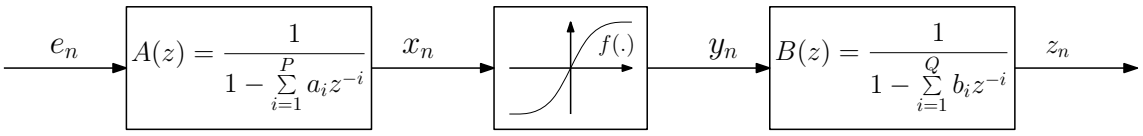

Figure 8.2: Complete model generating a nonlinearly distorted audio signal.

Figure 8.2 above represents the overall system, from the generation of the original and undistorted audio signal to its posterior nonlinear distortion. As in Chapter 7, short excerpts of the original signal are modeled by an AR process of order P. The

[2]This is half of a paragraph in Section 2 of the paper, and there is no reference or more descriptions of the experiments, for reasons of space, since the paper is a conference one.

first block represents an all-pole filter, excited by a Gaussian sequence $\mathbf{e}$ and giving rise to an excerpt of the original signal $\mathbf{x}$. This signal passes through the static memoryless nonlinearity $f(.)$, creating the intermediate signal $\mathbf{y}$. As in Section 7.2, we suppose that this function is anti-symmetric, invertible and possesses unitary derivative in zero, implying that its inverse $g(.) = f^{-1}(.)$ exists and satisfies:

- $g(y) = -g(-y)$, for all $y \in \mathbb{R}$;
- $g'(y) > 0$, for all $y \in \mathbb{R}$;
- $g'(0) = 1$.

We again as in Section 7.2 approximate $g(.)$ by its Taylor polynomial of order M centered in zero, in order to obtain a parametric form for the distortion. Finally, the linear filter whose transfer function is $B(z)$ is chosen to be an all-pole of order Q, since as we will see in the next sections, with this assumption it is easier to write the likelihood functions for the parameters. We note again that the orders P, M and Q are unknown beforehand and can be tuned manually by the user to obtain better results, an issue to be addressed in the future.

8.4 Computation of the likelihood

As in Chapter 7, we consider $\mathbf{x} = [x_1 \ \dots \ x_N]^T$ as an excerpt of the original signal of length N, $\mathbf{x}_0 = [x_0 \ \dots \ x_P]^T$ its first P time samples and $\mathbf{x}_1 = [x_{P+1} \ \dots x_N]^T$ its remaining $(N-P)$ time samples. Denote by $\mathbf{e} = [e_{P+1} \ \dots \ e_N]^T$ the corresponding excitation signal and by $\mathbf{y}$ the intermediate signal, split in $\mathbf{y}_0$ and $\mathbf{y}_1$ analogously to $\mathbf{x}$. Finally, denote by $\mathbf{z}$ the observed distorted signal and define $R = P + Q$. Split $\mathbf{z}$ in its first R time samples $\mathbf{z} = [z_1 \ \dots \ z_R]^T$ and its remaining $(N-R)$ time samples $\mathbf{z} = [z_{R+1} \ \dots \ z_N]^T$.

We computed in Section 7.2.2 the likelihood of parameters $\mathbf{m}$, $\mathbf{a}$ and σ_e^2, which is given by

$$p(\mathbf{y}|\mathbf{m},\mathbf{a},\sigma_e^2) = \prod_{n=P+1}^{N} \left|1 + \mathbf{h}_n^T\mathbf{m}\right| \times \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left[-\frac{1}{2\sigma_e^2}(\mathbf{y}+\mathbf{Y}\mathbf{m})^T\mathbf{A}^T\mathbf{A}(\mathbf{y}+\mathbf{Y}\mathbf{m})\right]. \quad (8.1)$$

Since we will need to perform another change of variables, it will be useful to change a little bit the notation:

- Recall firstly that this likelihood is not exact, but an approximation, since the term on the right side is the likelihood conditioned to $\mathbf{y}_0$, denoted by $p(\mathbf{y}_1|\mathbf{y}_0, \mathbf{m}, \mathbf{a}, \sigma_e^2)$.

- Each term of the Jacobian determinant above can be rewritten as $|g'(y_n; \mathbf{m})|$, where the derivative is taken with respect to the variable of the polynomial and we now make explicit the dependence of $g(.)$ on $\mathbf{m}$.

- The term $(\mathbf{y} + \mathbf{Ym})^T \mathbf{A}^T \mathbf{A}(\mathbf{y} + \mathbf{Ym})$ inside the exponential can be rewritten as $\mathbf{e}^T\mathbf{e} = \sum_{n=P+1}^{N} e_n^2$. In turn, each e_n is given by

$$e_n = x_n - \sum_{i=1}^{P} a_i x_{n-i} = g(y_n; \mathbf{m}) - \sum_{i=1}^{P} a_i g(y_{n-i}; \mathbf{m}), \tag{8.2}$$

  which we can denote for the moment as $e_n(\mathbf{m}, \mathbf{a})$.

Therefore, the likelihood in Equation 8.1 can be rewritten as:

$$p(\mathbf{y}|\mathbf{m}, \mathbf{a}, \sigma_e^2) \approx p(\mathbf{y}_1|\mathbf{y}_0, \mathbf{m}, \mathbf{a}, \sigma_e^2) = \prod_{n=P+1}^{N} |g'(y_n; \mathbf{m})| \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left[-\frac{1}{2\sigma_e^2} \sum_{n=P+1}^{N} e_n(\mathbf{m}, \mathbf{a})^2\right]. \tag{8.3}$$

We now consider the all-pole filter whose transfer function is $B(z)$ acting on the signal $\mathbf{y}$. The relation between $\mathbf{y}$ and $\mathbf{z}$ is given by

$$z_n = y_n + \sum_{i=1}^{Q} b_i z_{n-i}. \tag{8.4}$$

Since this equation is recursive and our signal is finite, it is not valid for every time sample of signal $\mathbf{z}$. When we used the AR equation to describe the relation between $\mathbf{e}$ and $\mathbf{x}$, it was valid only for x_n with $n = P + 1, \ldots, N$. Now, to compute some z_n we need the previous Q time samples, implying that the relation in Equation 8.4 is valid only for $n = R + 1, \ldots, N$, where R was defined as $P + Q$. If we write these equations, we have:

$$\begin{cases} z_{P+Q+1} &= y_{P+Q+1} + b_1 z_{P+Q} + b_2 x_{P+Q-1} + \cdots + b_Q z_{P+1} \\ z_{P+Q+2} &= y_{P+Q+2} + b_1 z_{P+Q+1} + b_2 x_{P+Q} + \cdots + b_Q z_{P+2} \\ \quad \vdots & \\ z_N &= y_N + b_1 z_{N-1} + b_2 z_{N-2} + \cdots + b_Q x_{N-Q}. \end{cases} \tag{8.5}$$

As for the AR model, we must assume that the time samples $z_{P+1}, z_{P+2}, \ldots, z_{P+Q}$ are known beforehand, implying that the above transformation from $\mathbf{y}$ to $\mathbf{z}$ is of unitary Jacobian.

Then, the likelihood in Equation 8.3 is updated to:

$$p(\mathbf{z}|\mathbf{b},\mathbf{m},\mathbf{a},\sigma_e^2) \approx p(\mathbf{z}_1|\mathbf{z}_0,\mathbf{b},\mathbf{m},\mathbf{a},\sigma_e^2) = \prod_{n=R+1}^{N} \left| \frac{\partial}{\partial z_n} h(z_n,\ldots,z_{n-i};\mathbf{b},\mathbf{m}) \right| \frac{1}{(2\pi\sigma_e^2)^{\frac{N-R}{2}}} \exp\left[-\frac{1}{2\sigma_e^2} \sum_{n=R+1}^{N} e_n(\mathbf{b},\mathbf{m},\mathbf{a})^2 \right], \quad (8.6)$$

where the function $h(\cdot;\mathbf{b},\mathbf{m})$ is defined by

$$h(z_n;\mathbf{b},\mathbf{m}) = g\left(\underbrace{z_n - \sum_{i=1}^{Q} b_i z_{n-i}}_{y_n}; \mathbf{m} \right), \quad (8.7)$$

and $e_n(\mathbf{b},\mathbf{m},\mathbf{a})$ is given by

$$e_n = g\left(\underbrace{z_n - \sum_{j=1}^{Q} b_j z_{n-j}}_{y_n}; \mathbf{m} \right) - \sum_{i=1}^{P} a_i g\left(\underbrace{z_{n-i} - \sum_{j=1}^{Q} b_j z_{n-i-j}}_{y_{n-i}}; \mathbf{m} \right). \quad (8.8)$$

Note that now we must make explicit in the Jacobian determinant the variable in which each term is derived, since it does not depend on a single time sample of signal $\mathbf{z}$ anymore.

As before, we want to find the best values of $\mathbf{b}$ and $\mathbf{m}$ based only on the knowledge of the distorted signal $\mathbf{z}$. If maximizing the likelihood by means of a deterministic method was reasonable in the memoryless case, here this strategy seems to be inadequate, since the likelihood in Equation 8.6 is a very complicated function, potentially with several local maxima. In fact, tests performed previously in [45] show that the Newton method is not very effective here, since its convergence to the global maximum is conditioned on a good starting point, which is not readily available. In order to estimate the desired parameters we employ again Bayesian methods, described in more detail in the next section.

8.5 Bayesian estimation of the distortion parameters

By using the Bayes' Theorem, we have

$$p(\mathbf{b},\mathbf{m},\mathbf{a},\sigma_e^2|\mathbf{z}) \propto p(\mathbf{z}|\mathbf{b},\mathbf{m},\mathbf{a},\sigma_e^2)p(\mathbf{b})p(\mathbf{m})p(\mathbf{a})p(\sigma_e^2), \quad (8.9)$$

where $p(\mathbf{b}), p(\mathbf{m}), p(\mathbf{a})$ and $p(\sigma_e^2)$ are the prior distributions for the parameters, given, similarly as in the memoryless case, by:

- $p(\mathbf{b}) \propto \exp\left(-\frac{1}{2\sigma_b^2}\mathbf{b}^T\mathbf{b}\right)\Phi(\mathbf{b})$, where function $\Phi(\mathbf{b})$ returns 1 if $\mathbf{b}$ contains the coefficients of a stable all-pole filter and 0 otherwise;
- $p(\mathbf{m}) \propto \exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}^T\mathbf{m}\right)\Omega(\mathbf{m})$, where function $\Omega(\mathbf{m})$ returns 1 if g satisfies the required constraints presented in Section 7.2.1 and 0 otherwise;
- $p(\mathbf{a}) \propto \exp\left(-\frac{1}{2\sigma_a^2}\mathbf{a}^T\mathbf{a}\right)\Phi(\mathbf{a})$, where function $\Phi(\mathbf{a})$ is the same defined in $p(\mathbf{b})$;
- $\sigma_e^2 \sim \text{IG}(\alpha, \beta)$, where IG denotes the Inverse Gamma distribution.

The hyper-parameters σ_m^2, σ_a^2, σ_b^2, α and β are chosen to turn the priors vague.

The posterior distribution is then given by:

$$
\begin{aligned}
&p(\mathbf{b}, \mathbf{m}, \mathbf{a}, \sigma_e^2|\mathbf{z}) \propto \\
&\prod_{n=R+1}^{N} \left|\frac{\partial}{\partial z_n} h(z_n, \ldots, z_{n-i}; \mathbf{b}, \mathbf{m})\right| \frac{1}{(2\pi\sigma_e^2)^{\frac{N-R}{2}}} \exp\left[-\frac{1}{2\sigma_e^2}\sum_{n=R+1}^{N} e_n(\mathbf{b}, \mathbf{m}, \mathbf{a})^2\right] \times \\
&\left[\exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}^T\mathbf{m}\right)\Omega(\mathbf{m})\right] \times \left[\exp\left(-\frac{1}{2\sigma_a^2}\mathbf{a}^T\mathbf{a}\right)\Phi(\mathbf{a})\right] \times \\
&\left[\exp\left(-\frac{1}{2\sigma_b^2}\mathbf{b}^T\mathbf{b}\right)\Phi(\mathbf{b})\right] \times \left[\sigma_e^{2^{-(\alpha+1)}} \exp\left(-\frac{\beta}{\sigma_e^2}\right)\right].
\end{aligned} \tag{8.10}
$$

As in the memoryless case, we obtain samples from this distribution and compute the posterior mean as an estimate of the parameters $\mathbf{b}$, $\mathbf{m}$, $\mathbf{a}$, σ_e^2.

We sample from this distribution using the Gibbs sampler, implemented in the following way:

a) Initialize values $\mathbf{m}^{(0)}$, $\mathbf{a}^{(0)}$, $\mathbf{b}^{(0)}$ and $\sigma_e^{2^{(0)}}$

b) For k from 1 to N_{iter}:

   i) Sample $\sigma_e^{2^{(k)}}$ from distribution $p(\sigma_e^2|\mathbf{b}^{(k-1)}, \mathbf{m}^{(k-1)}, \mathbf{a}^{(k-1)}, \mathbf{z})$

   ii) Sample $\mathbf{a}^{(k)}$ from distribution $p(\mathbf{a}|\mathbf{b}^{(k-1)}, \mathbf{m}^{(k-1)}, \sigma_e^{2^{(k)}}, \mathbf{z})$

   iii) Sample $\mathbf{b}^{(k)}$ from distribution $p(\mathbf{b}|\mathbf{m}^{(k-1)}, \mathbf{a}^{(k)}, \sigma_e^{2^{(k)}}, \mathbf{z})$

   iv) Sample $\mathbf{m}^{(k)}$ from distribution $p(\mathbf{m}|\mathbf{b}^{(k)}, \mathbf{a}^{(k)}, \sigma_e^{2^{(k)}}, \mathbf{z})$

The posterior mean is estimated by the formulas:

$$\widehat{\sigma_e^2} = \frac{1}{N_{\text{iter}} - N_{\text{burn-in}}} \sum_{k=N_{\text{burn-in}}+1}^{N_{\text{iter}}} \sigma_e^{2^{(k)}}, \tag{8.11}$$

$$\widehat{\mathbf{a}} = \frac{1}{N_{\text{iter}} - N_{\text{burn-in}}} \sum_{k=N_{\text{burn-in}}+1}^{N_{\text{iter}}} \mathbf{a}^{(k)}, \tag{8.12}$$

$$\widehat{\mathbf{b}} = \frac{1}{N_{\text{iter}} - N_{\text{burn-in}}} \sum_{k=N_{\text{burn-in}}+1}^{N_{\text{iter}}} \mathbf{b}^{(k)}, \tag{8.13}$$

$$\widehat{\mathbf{m}} = \frac{1}{N_{\text{iter}} - N_{\text{burn-in}}} \sum_{k=N_{\text{burn-in}}+1}^{N_{\text{iter}}} \mathbf{m}^{(k)}, \tag{8.14}$$

where N_b is the *burn-in time* of the Markov chain, as explained in Chapter 2.

To compute the conditional distributions required by the Gibbs sampler, we consider the posterior distribution in Equation 8.10 as a function of only the variable of interest, ignoring the others, resulting in the distributions described below.

8.5.1 Conditional distribution of σ_e^2

$$\begin{aligned} p(\sigma_e^2|\mathbf{b},\mathbf{m},\mathbf{a},\mathbf{z}) &\propto \left[\frac{1}{(\sigma_e^2)^{\frac{N-R}{2}}} \exp\left\{-\frac{1}{2\sigma_e^2}\sum_{n=R+1}^{N} e_n(\mathbf{b},\mathbf{m},\mathbf{a})^2\right\}\right] \\ &\quad \times \left[(\sigma_e^2)^{-(\alpha+1)} \exp\left(-\frac{\beta}{\sigma_e^2}\right)\right] \\ &\propto (\sigma_e^2)^{-\left(\frac{N-R}{2}+\alpha+1\right)} \exp\left(-\frac{(\sum_{n=R+1}^{N} e_n(\mathbf{b},\mathbf{m},\mathbf{a})^2 + \beta)/2}{\sigma_e^2}\right), \end{aligned} \tag{8.15}$$

which implies that $p(\sigma_e^2|\mathbf{b},\mathbf{m},\mathbf{a},\mathbf{z}) \sim \text{IG}\left(\frac{N-R}{2}+\alpha, \frac{\sum_{n=R+1}^{N} e_n(\mathbf{b},\mathbf{m},\mathbf{a})^2+\beta}{2}\right)$ a distribution that is easy to sample from, by using built-in functions of numerical softwares.

8.5.2 Conditional distribution of a

Recall that we can write the posterior distribution in Equation 8.10 in such a way that the dependence in $\mathbf{a}$ is quadratic, implying that its conditional distribution is given by:

$$\begin{aligned} p(\mathbf{a}|\mathbf{b},\mathbf{m},\sigma_e^2,\mathbf{z}) &\propto \left[\exp\left\{-\frac{1}{2\sigma_e^2}(\mathbf{x}_1 - \mathbf{X}\mathbf{a})^T(\mathbf{x}_1 - \mathbf{X}\mathbf{a})\right\}\right] \times \left[\exp\left(-\frac{1}{2\sigma_a^2}\mathbf{a}^T\mathbf{a}\right)\Phi(\mathbf{a})\right] \\ &= \exp\left\{-\frac{1}{2\sigma_e^2}(\mathbf{x}_1 - \mathbf{X}\mathbf{a})^T(\mathbf{x}_1 - \mathbf{X}\mathbf{a}) - \frac{1}{2\sigma_a^2}\mathbf{a}^T\mathbf{a}\right\}\Phi(\mathbf{a}), \end{aligned} \tag{8.16}$$

where matrix $\mathbf{X}$ was defined in Chapter 4. Each entry of vector $\mathbf{x}_1$ necessary to compute the formula above is given by

$$x_n = g\left(z_n - \sum_{j=1}^{Q} b_j z_{n-j}; \mathbf{m}\right), \tag{8.17}$$

implying that it can be computed only from $\mathbf{z}$ and from the variables on which the distribution is conditioned.

Therefore, the conditional distribution for $\mathbf{a}$ is a truncated Gaussian, constrained to the support of the function $\Phi(.)$, with covariance matrix $\mathbf{C}_a = \left(\frac{\mathbf{X}^T\mathbf{X}}{\sigma_e^2} - \frac{\mathbf{I}_P}{\sigma_a^2}\right)^{-1}$ and mean $\boldsymbol{\mu}_a = \frac{1}{\sigma_e^2}\mathbf{C}_a\mathbf{X}^T\mathbf{x}_1 = \frac{1}{\sigma_e^2}\left(\frac{\mathbf{X}^T\mathbf{X}}{\sigma_e^2} - \frac{\mathbf{I}_P}{\sigma_a^2}\right)\mathbf{X}^T\mathbf{x}_1$.

8.5.3 Conditional distribution of m

As in the memoryless case, the conditional distribution of $\mathbf{m}$ does not belong to a well known family that is easy to sample from, because of the Jacobian determinant in Equation 8.6:

$$\begin{aligned} &p(\mathbf{m}|\mathbf{b}, \mathbf{a}, \sigma_e^2, \mathbf{z}) \propto \\ &\left[\prod_{n=R+1}^{N} \left|\frac{\partial}{\partial z_n} h(z_n, \ldots, z_{n-i}; \mathbf{b}, \mathbf{m})\right| \times \exp\left\{-\frac{1}{2\sigma_e^2}\sum_{n=R+1}^{N} e_n(\mathbf{b}, \mathbf{m}, \mathbf{a})^2\right\}\right] \times \\ &\left[\exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}^T\mathbf{m}\right)\Omega(\mathbf{m})\right]. \end{aligned} \tag{8.18}$$

We can rewrite the argument of the first exponential in order to make explicit the quadratic dependence in $\mathbf{m}$:

$$\begin{aligned} p(\mathbf{m}|\mathbf{b}, \mathbf{a}, \sigma_e^2, \mathbf{z}) \propto &\left[\prod_{n=R+1}^{N} \left|\frac{\partial}{\partial z_n} h(z_n, \ldots, z_{n-i}; \mathbf{b}, \mathbf{m})\right| \times \right. \\ &\left. \exp\left\{-\frac{1}{2\sigma_e^2}(\mathbf{y} + \mathbf{Y}\mathbf{m})^T\mathbf{A}^T\mathbf{A}(\mathbf{y} + \mathbf{Y}\mathbf{m})\right\}\right] \times \\ &\left[\exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}^T\mathbf{m}\right)\Omega(\mathbf{m})\right], \end{aligned} \tag{8.19}$$

where matrix $\mathbf{Y}$ is given by

$$\mathbf{Y} = \begin{bmatrix} y_1^3 & y_1^5 & \cdots & y_1^{2M+1} \\ y_2^3 & y_2^5 & \cdots & y_2^{2M+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_N^3 & y_N^5 & \cdots & y_N^{2M+1} \end{bmatrix}, \tag{8.20}$$

each y_n is computed as

$$y_n = z_n - \sum_{i=1}^{Q} b_i z_{n-i}, \tag{8.21}$$

and matrix $\mathbf{A}$ was defined in Chapter 4. Therefore, all the quantities required can also be computed from $\mathbf{z}$ and from the other variables on which the distribution is conditioned.

To sample from this distribution we employ a Metropolis within Gibbs step, where the proposal distribution at each step is computed by means of the Laplace approximation, as in the memoryless case. More specifically, the proposal distribution is Gaussian with mean equal to some mode of $\ln p(\mathbf{m}|\mathbf{b}, \mathbf{a}, \sigma_e^2, \mathbf{z})$ and covariance matrix given by the negative-Hessian of this function computed at this mode. This maximum point is obtained by the Newton method with starting point equal to the current sample $\mathbf{m}^{(k-1)}$. The gradient vector and the Hessian matrix were computed by hand in a very tedious computation that was omitted from the text for conciseness sake.

8.5.4 Conditional distribution of b

Analogously to the conditional distribution of $\mathbf{m}$, this distribution is not a member of a well known family:

$$\begin{aligned} &p(\mathbf{b}|\mathbf{m}, \mathbf{a}, \sigma_e^2, \mathbf{z}) \propto \\ &\prod_{n=R+1}^{N} \left| \frac{\partial}{\partial z_n} h(z_n, \ldots, z_{n-i}; \mathbf{b}, \mathbf{m}) \right| \exp\left[-\frac{1}{2\sigma_e^2} \sum_{n=R+1}^{N} e_n(\mathbf{b}, \mathbf{m}, \mathbf{a})^2 \right] \times \\ &\left[\exp\left(-\frac{1}{2\sigma_b^2} \mathbf{b}^T \mathbf{b} \right) \Phi(\mathbf{b}) \right]. \end{aligned} \tag{8.22}$$

Now the situation is quite more complicated, since the argument inside the first exponential cannot be rewritten in order to be quadratic in $\mathbf{b}$, since higher powers of the coefficients b_i appear in each e_n:

$$e_n(\mathbf{b}, \mathbf{m}, \mathbf{a}) = g\left(z_n - \sum_{j=1}^{Q} b_j z_{n-j}; \mathbf{m} \right) - \sum_{i=1}^{P} a_i g\left(z_{n-i} - \sum_{j=1}^{Q} b_j z_{n-i-j}; \mathbf{m} \right). \tag{8.23}$$

Despite this difficulty, we can still use the Laplace approximation and Metropolis within Gibbs, as in the sampling of the conditional distribution of $\mathbf{m}$. To find a maximum point of $\ln p(\mathbf{b}|\mathbf{m}, \mathbf{a}, \sigma_e^2, \mathbf{z})$ we use the Newton method initialized in the current sample $\mathbf{b}^{(k-1)}$ and use the obtained point as the mean of the proposal Gaussian distribution. Its covariance matrix is given by the negative-Hessian of $\ln p(\mathbf{b}|\mathbf{m}, \mathbf{a}, \sigma_e^2, \mathbf{z})$ computed at the obtained maximum. Again the gradient vector

and the Hessian matrix were computed by hand, but now the computation is even more tedious and longer than for the conditional distribution of **m**. It will also be omitted from the text for conciseness reasons.

8.5.5 Reinitialization procedure

Preliminary tests implementing the procedure described above with real signals distorted with artificial distortions showed that there is a great chance of the chain being trapped for a very long time around a local maximum, if the starting point is not close enough to the global maximum. This was also an issue in using the Newton method for maximizing the likelihood in Equation 8.6. Some theorems of the MCMC theory guarantee that the chain will eventually explore the region where the global maximum is located, but this can take a very long time, since some sample from the proposal distribution must be in this region, and this sample must be accepted. In order to overcome this problem by increasing the odds of the chain quickly reaching the region around the global maximum, the reinitialization procedure described below was proposed:

i) Randomly choose parameters **a**, **b** and **m** respecting the constraints imposed by functions $\Phi(.)$ and $\Omega(.)$;

ii) Find a local maximum of their respective conditional distributions, by using the Newton method initialized at the previously chosen values;

iii) Approximate the conditional distributions around the local maximum found in the previous step via Laplace approximation;

iv) Sample **a**, **b** and **m** from this approximate distribution;

v) Accept the samples generated in the previous step with probability given by the acceptance probability of the Metropolis-Hastings algorithm (Equation 3.3), with $q(\cdot\,;\cdot) = 1$.

The last step departs from the standard Metropolis-Hastings algorithm, since the proposal distribution is not taken into account in the calculation of the acceptance probability. The precise computation of this quantity is very complicated, since the sampled distribution is not specified by a single formula, but is constructed in steps i), ii) and iii) above. However, this simplification does not impact the convergence of the chain, since this procedure is only performed during the burn-in. How often the reinitialization procedure is performed is a parameter set by the user.

Chapter 9

Results and Future works

In Chapters 7 and 8 we presented methods for restoring audio signals degraded by nonlinear distortion without and with memory, respectively. We now report some results obtained with these methods. Tests were performed with real and artificial signals, artificially distorted by nonlinear systems following or not their respective models. The organization of the chapter is as follows: in Sections 9.1 and 9.2 we present the results for the memoryless case, where the distortion possesses polynomial inverse or is piecewise linear, respectively, and in Section 9.3 the effects of some parameters are tested using some statistical tools and discussed in details; in Section 9.4 the results for the case with memory are shown, and finally in Section 9.5 some conclusions are summarized and future works are indicated.

Algorithms were implemented and executed in my personal computer[1] ASUS^{TM}[2] K45VM with processor Intel Core i7 3610QM^{TM}[3] at 2.3 GHz clock and possessing 8 GB of RAM, in MATLAB^{TM}[4] version R2014a.

9.1 Memoryless nonlinear distortions: Polynomial approximation

The polynomial approximation to the inverse of the nonlinear distortion was presented in Section 7.2. Two main types of tests were performed: artificial signals modified with artificial distortions following the model; and real signals modified with artificial distortions both following the model or more general ones. The first type of test is performed in order to assess the method's accuracy and convergence, while the second one aims at verifying the method's generality and capability of

[1]Except those presented in Section 9.4, for the restoration of signals degraded by nonlinear distortions with memory

[2]ASUSTeK Computer Inc., `http://www.asus.com/`

[3]Intel Corporation, `http://www.intel.com/`

[4]The MathWorks, Inc., `http://www.mathworks.com/`

treating more realistic distortions.

9.1.1 Artificial signals and distortion following the model

An artificial signal with 1,000 time samples that follows an AR model was generated. The filter representing the AR model had poles at frequencies $\pi/16$, $\pi/8$ and $\pi/4$ radians per time sample and its respective conjugated frequencies, each one with module 0.99. The variance of the excitation signal was chosen to be $\sigma_e^2 = 5 \times 10^{-6}$, a value of an order of magnitude typically seen in real digitized audio signals, whose amplitude is at most 1. The signal was distorted by the inverse function of the polynomial

$$g(y) = y + 5y^3 + 30y^5, \tag{9.1}$$

and this inverse was computed for each value of the original signal $\mathbf{x}$ via the bisection method. Note that $g(.)$ is in fact invertible, since its derivative is always positive.

We wish then to recover the variables $\mathbf{m} = [5 \;\; 30]^T$ containing the coefficients of $g(.)$ except for the first one, $\mathbf{a} = [5.1713 \;\; -11.7727 \;\; 15.1104 \;\; -11.5384 \;\; 4.9675 \;\; -0.9415]^T$ containing the coefficients of the AR model above and the variance of the excitation signal $\sigma_e^2 = 5 \times 10^{-6}$.

The single block of $N = 1{,}000$ time samples was analyzed, with constants P and M fixed at their correct values, 6 and 2, respectively. The parameters of the prior distributions were $\sigma_m^2 = 10^{10}$, $\sigma_a^2 = 10^{10}$ and $\alpha = \beta = 10^{-10}$.

A Gibbs sampler together with a Metropolis step for sampling $\mathbf{m}$, as described in Section 7.2, was run for 100 iterations, each one lasting approximately 0.412 s. The burn-in period was 50 iterations. Note that from the statistical viewpoint this is too few iterations, but for our purposes here, obtaining good audible results, this amount of iterations suffice.

Figure 9.1 shows the gradual convergence along the iterations to the polynomial coefficients, the most important parameters to be estimated. Although the chain output appears to be constant after the first ones, this is not the case. Zooming on the last 50 iterations, after the burn-in period, illustrates this point, as can be seen in Figure 9.2. We can also compare the original, distorted and restored signals by plotting them simultaneously or in a scatter plot, as in Figures 9.3 and 9.4, respectively. Figure 9.5 illustrates the convergence of $\mathbf{a}$ and finally in Tables 9.1 and 9.2 we can compare the real and estimated values, as the mean after the burn-in time, for parameters $\mathbf{a}$ and $\mathbf{m}$, respectively. On the convergence plots, the red square and the green circle always denote the real and estimated values, respectively.

We can then conclude that the method is capable of correctly identifying the desired parameters in this simple scenario, and assess its accuracy and convergence.

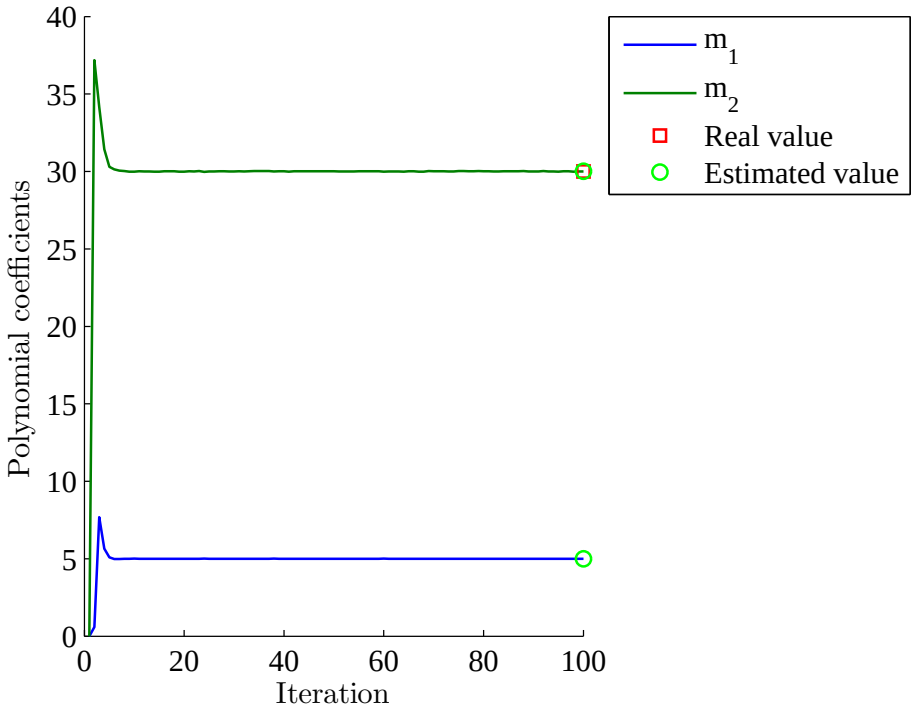

Figure 9.1: Artificial signal and distortion with polynomial inverse: Convergence of polynomial coefficients.

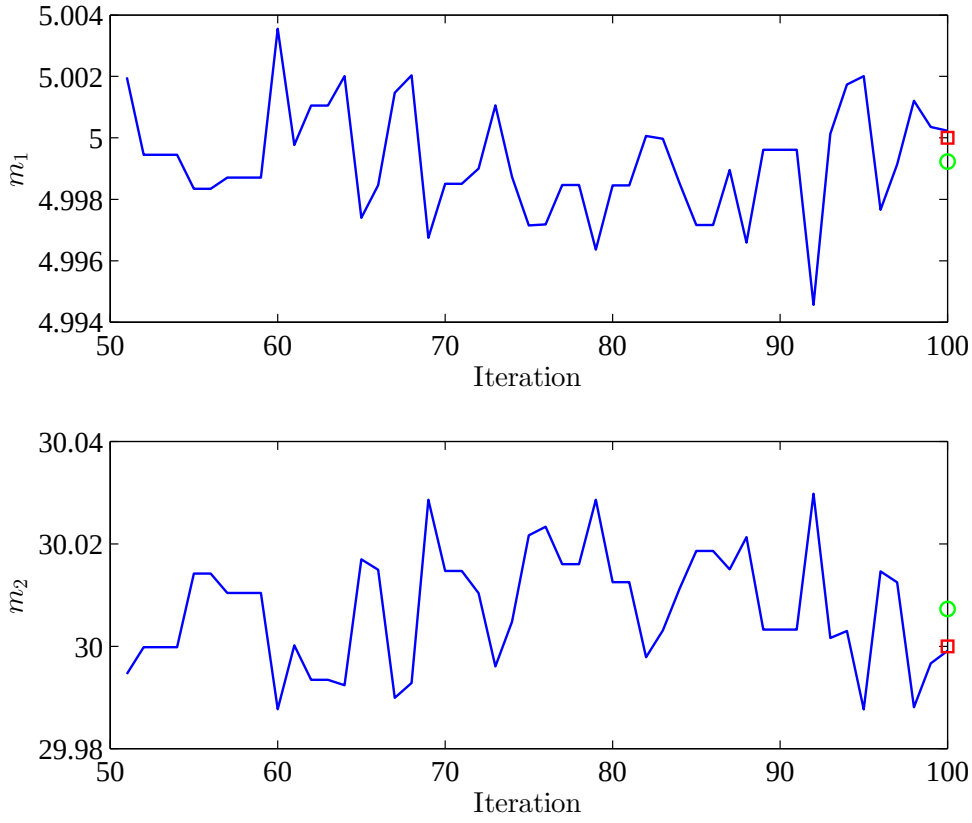

Figure 9.2: Artificial signal and distortion with polynomial inverse: Convergence of polynomial coefficients after the burn-in time.

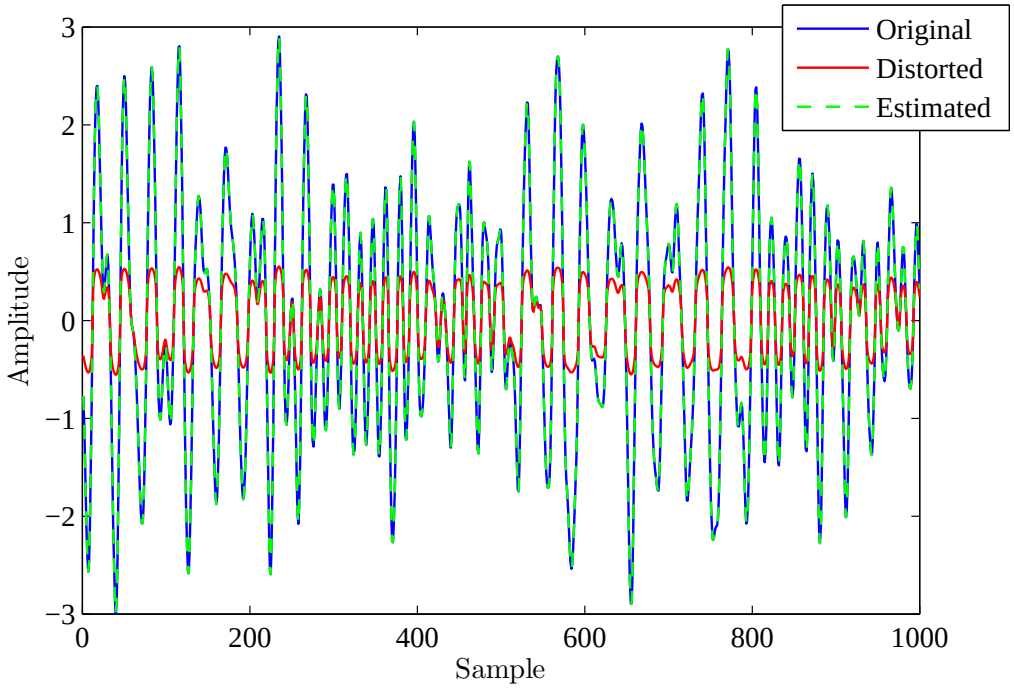

Figure 9.3: Artificial signal and distortion with polynomial inverse: Comparison of original, distorted and restored signals.

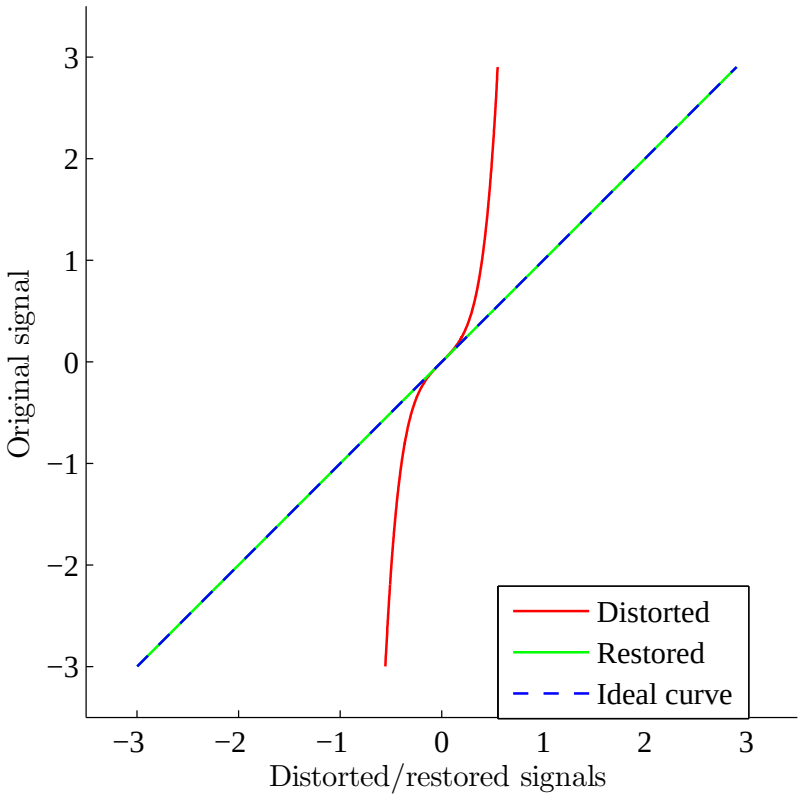

Figure 9.4: Artificial signal and distortion with polynomial inverse: Distorted and restored signals plotted against the original one.

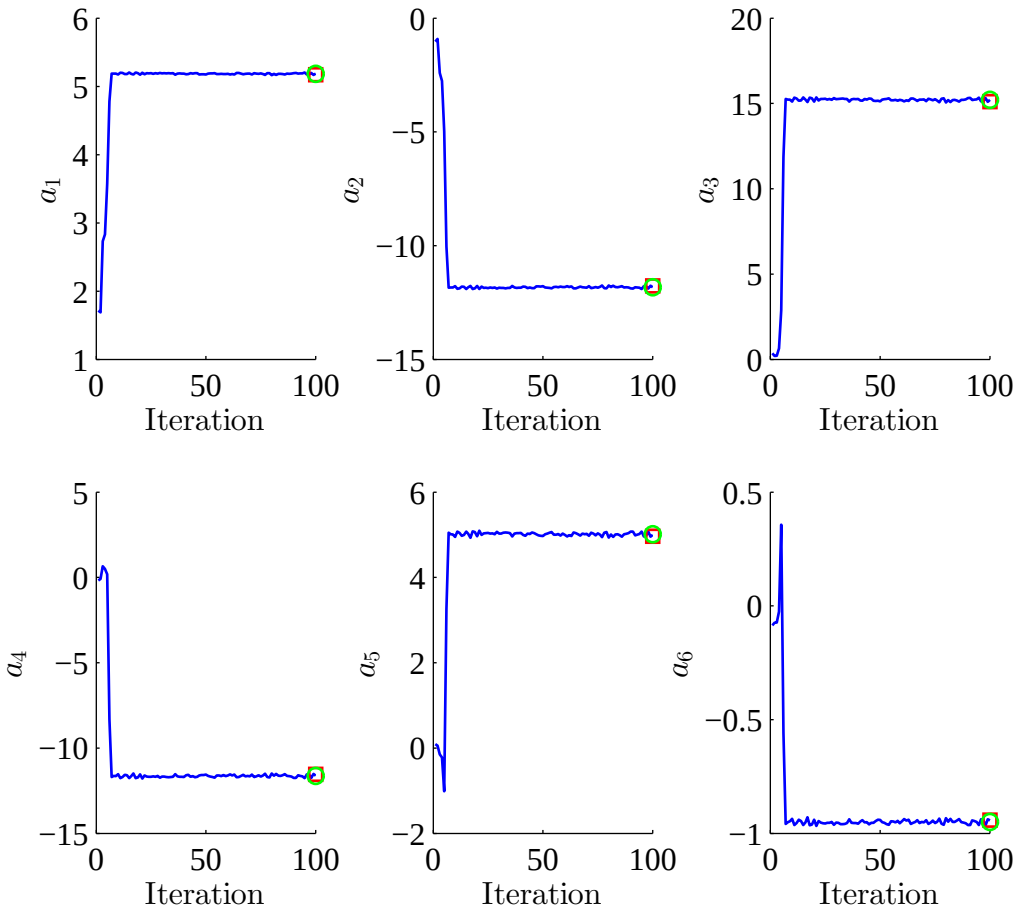

Figure 9.5: Artificial signal and distortion with polynomial inverse: Convergence of the AR model coefficients.

Table 9.1: Artificial signal and distortion with polynomial inverse: Comparison of real and estimated values of the coefficients of the AR model.

	Real	Estimated
a_1	5.1713	5.1846
a_2	-11.7727	-11.8278
a_3	15.1104	15.2079
a_4	-11.5384	-11.6288
a_5	4.9675	5.0102
a_6	-0.9415	-0.9493

Table 9.2: Artificial signal and distortion with polynomial inverse: Comparison of real and estimated values of the polynomial coefficients.

	Real	Estimated
m_1	5	4.9992
m_2	30	30.0073

9.1.2 Real signals and distortion following the model

The next step is to consider real signals distorted by a function whose inverse is polynomial. We consider three signals:

- `flute.wav`, a 3-s single note played on the flute, with some vibrato;
- `orchestra.wav`, a 9-s long excerpt of orchestral music, with long notes being played;
- `voice.wav`, a 2-s long speech signal in Portuguese, emitted by a female voice.

All the signals above are monophonic, in PCM format and sampled at 44,100 Hz, except `voice.wav`, sampled at 48,000 Hz. Each one was distorted by the inverse of the polynomial $g(y) = y + 5y^3 + 30y^5$, as in the previous case. A block of 1,000 time samples around the maximum amplitude time sample of each signal was analyzed, since this region was expected to provide more information about the high-order terms in $g(.)$.

The value of P that better describes the considered block of signal is unknown and fixed at 40 in all cases. To verify the accuracy of the estimation of $\mathbf{a}$, we compare its estimated values via the Gibbs sampler with the values estimated directly from the original signal using the *covariance method* [71], implemented in the `arcov` function of MATLAB™.

Since we are dealing with real signals, we can compare them by hearing, and not only by looking at plots that can possibly fool our senses. The tested signals are available in website [72] under the link *Companion website to my D.Sc. Thesis*[5]. Moreover, we compare the restored and original signals via the *Rnonlin* metric [47], a perceptually-based objective evaluation tool that mimics the way our ears and brain interpret nonlinearity in audio signals. It compares the tested signal against the original one using a series of filters, called *gammatone filters*, and returns a grade from 0 to 1. The closer the grade is to 1, better the tested signal should sound. It is important to note that the scale is not linear: a signal that scores 0.7 could sound **very** bad, and for some signals there is a great difference if its distorted version scores 0.95 and the restored one scores 0.99, despite the numbers being quite close. Grades below 0.7 were not observed in any of our tests. It is worth noting that the Rnonlin metric is invariant under scaling of the signals: multiply the tested or the reference signal by a positive constant does not affect the obtained grade. Therefore, any eventual scaling done to avoid clipping when saving some signal does not affect its respective Rnonlin grade.

For each of the three signals tested, three figures are shown: the first one displays the convergence of $\mathbf{m}$ (Figures 9.6, 9.9 and 9.12), the second one the convergence of the first 6 coefficients of the respective AR models (Figures 9.7, 9.10 and 9.13) and the third one the distorted and restored signals against the original ones. (Figures 9.8, 9.11 and 9.14).

Table 9.3: Real signals and distortion with polynomial inverse: Rnonlin grades for distorted and restored signals.

	`flute.wav`	`classical.wav`	`voice.wav`
Distorted	0.8775	0.9615	0.9165
Restored	0.9954	0.9985	0.9955

[5]Note that the dynamic range of these signals is different from the dynamic range presented in the figures below. This occurs because in order to save an audio signal in MATLAB^{TM} it is necessary to normalize it by keeping its maximum amplitude time sample equal to one; otherwise the signal will be clipped.

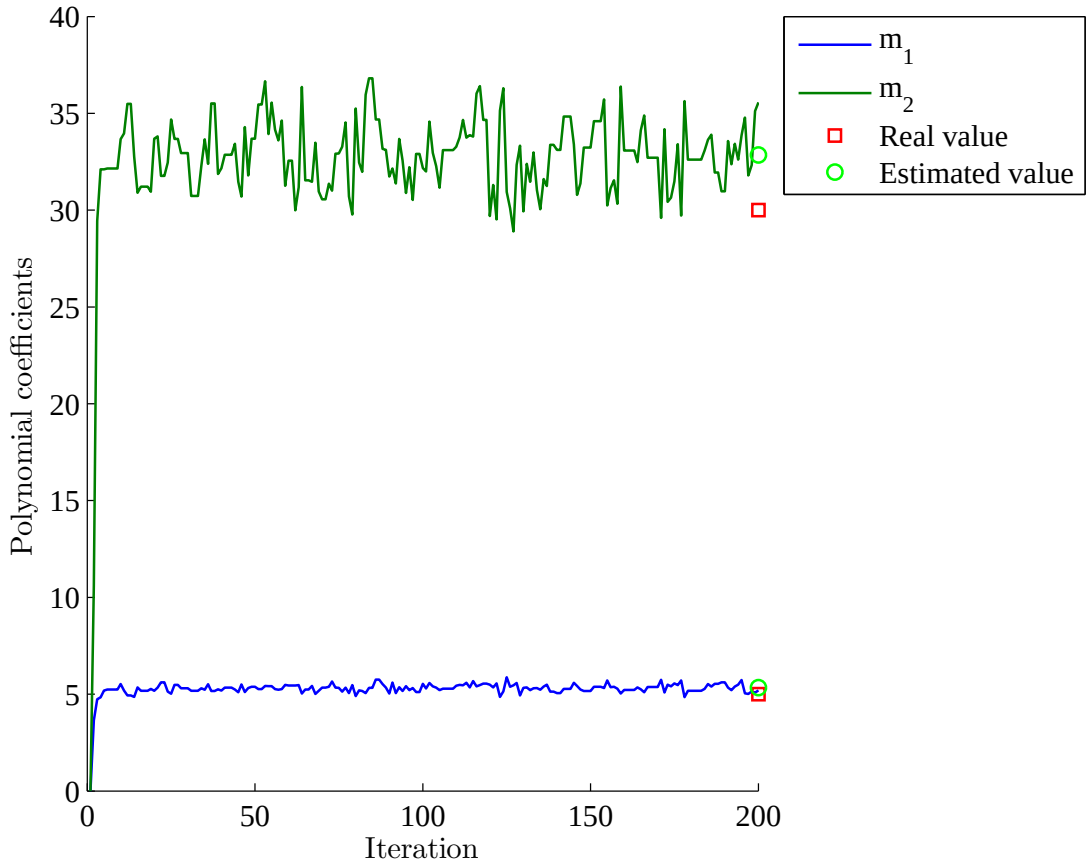

Figure 9.6: Signal `flute.wav` and distortion with polynomial inverse: Convergence of polynomial coefficients.

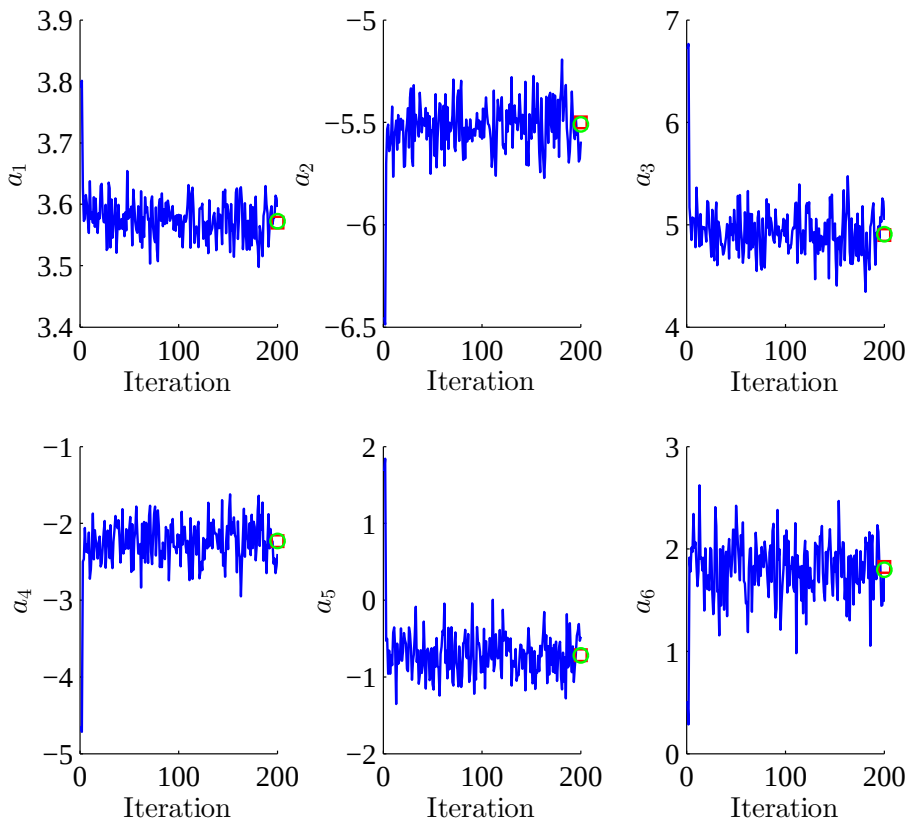

Figure 9.7: Signal `flute.wav` and distortion with polynomial inverse: Convergence of the first 6 AR model coefficients.

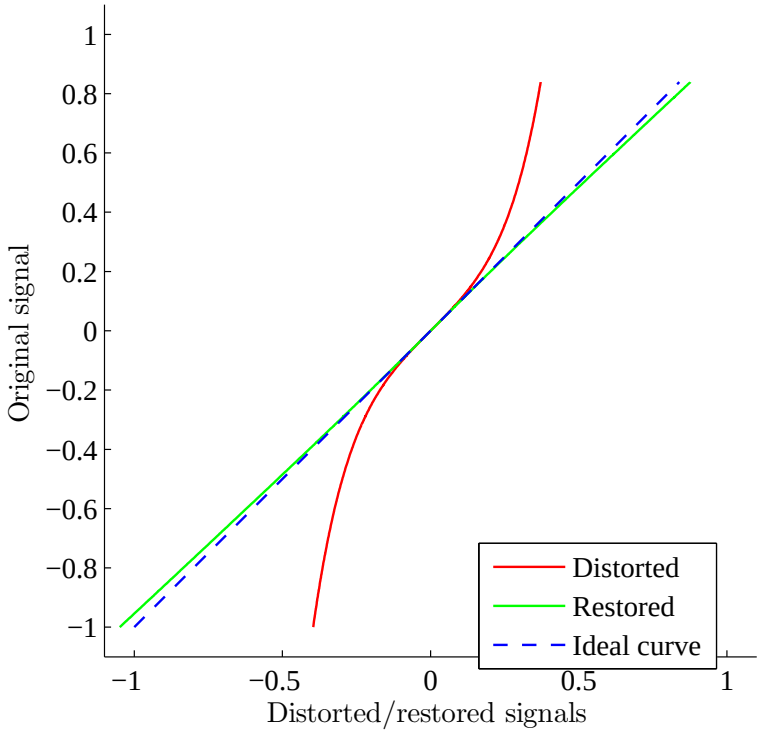

Figure 9.8: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one.

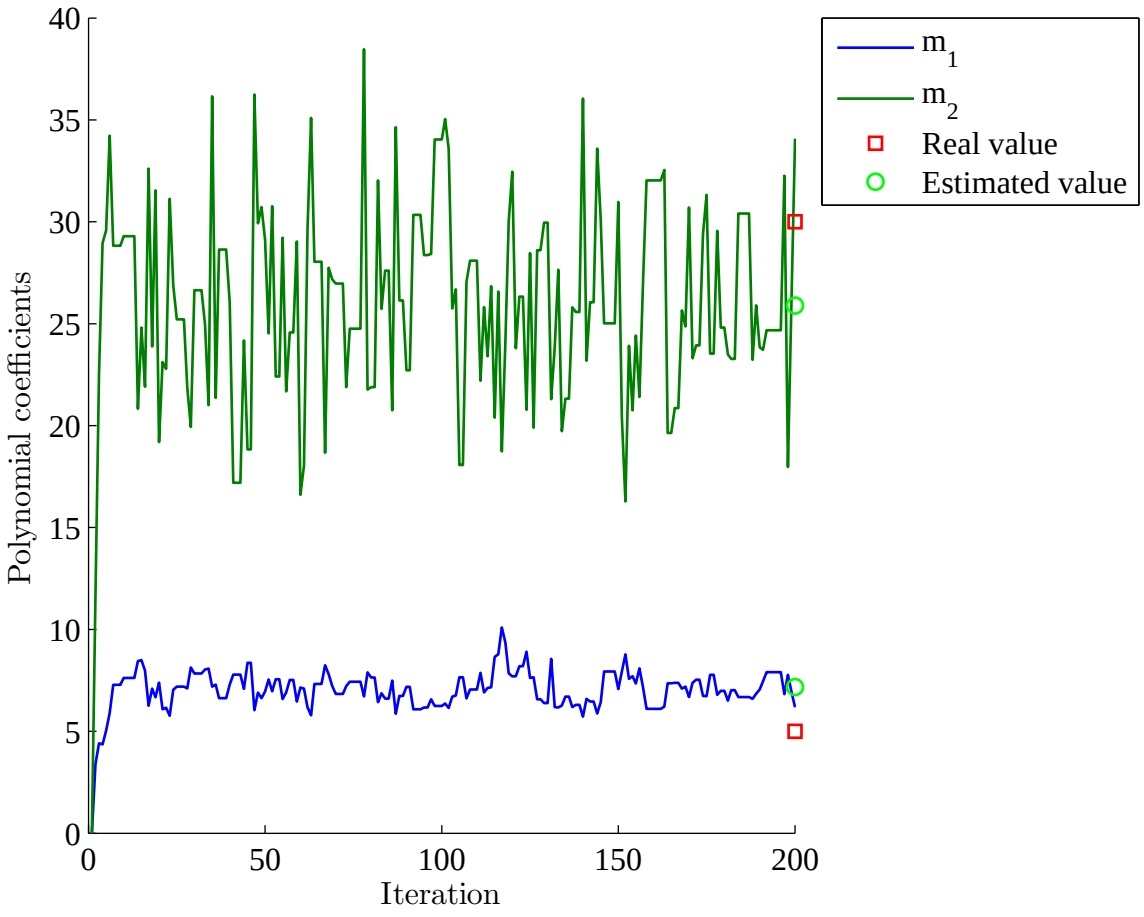

Figure 9.9: Signal `classical.wav` and distortion with polynomial inverse: Convergence of polynomial coefficients.

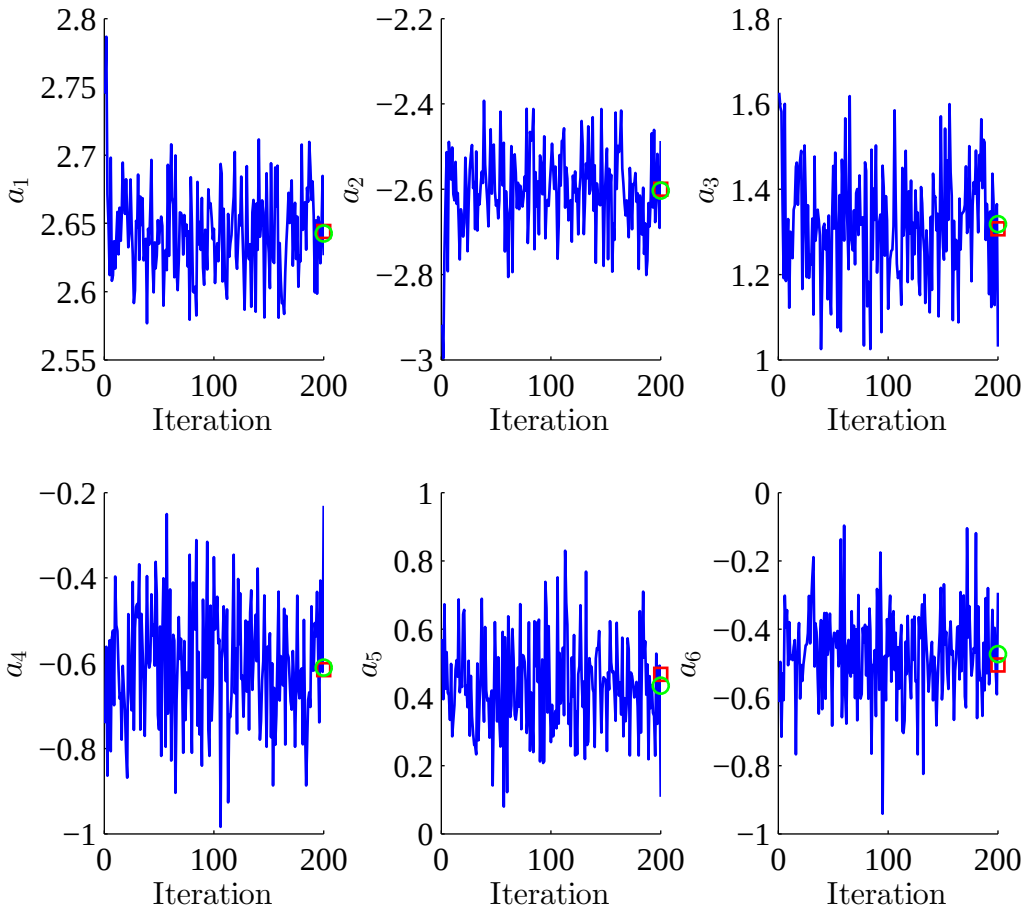

Figure 9.10: Signal `classical.wav` and distortion with polynomial inverse: Convergence of the first 6 AR model coefficients.

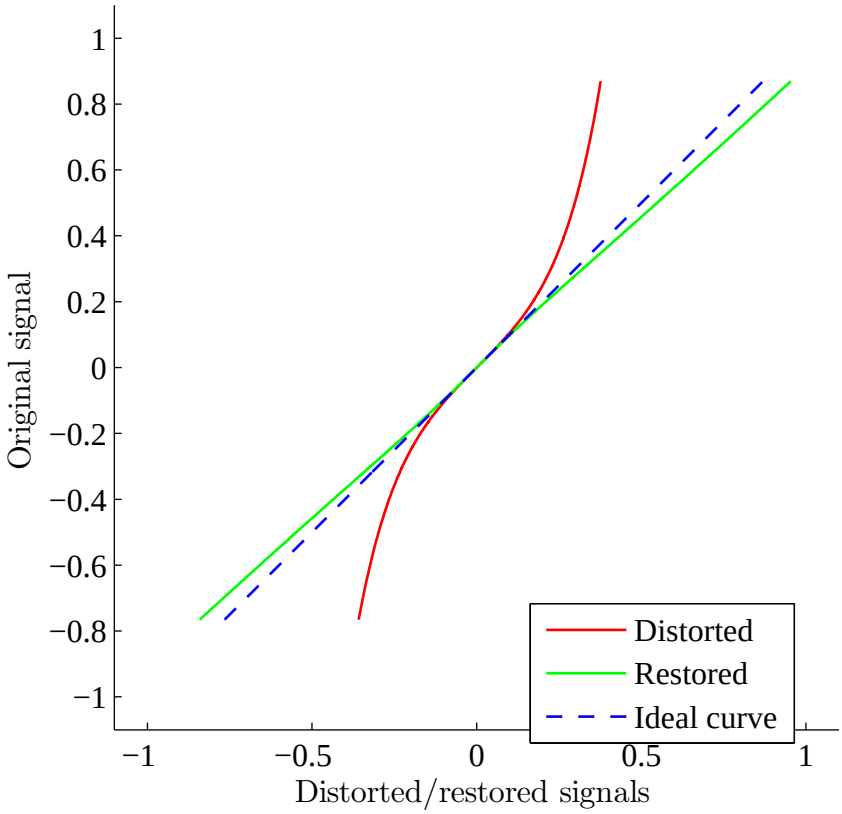

Figure 9.11: Signal `classical.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one.

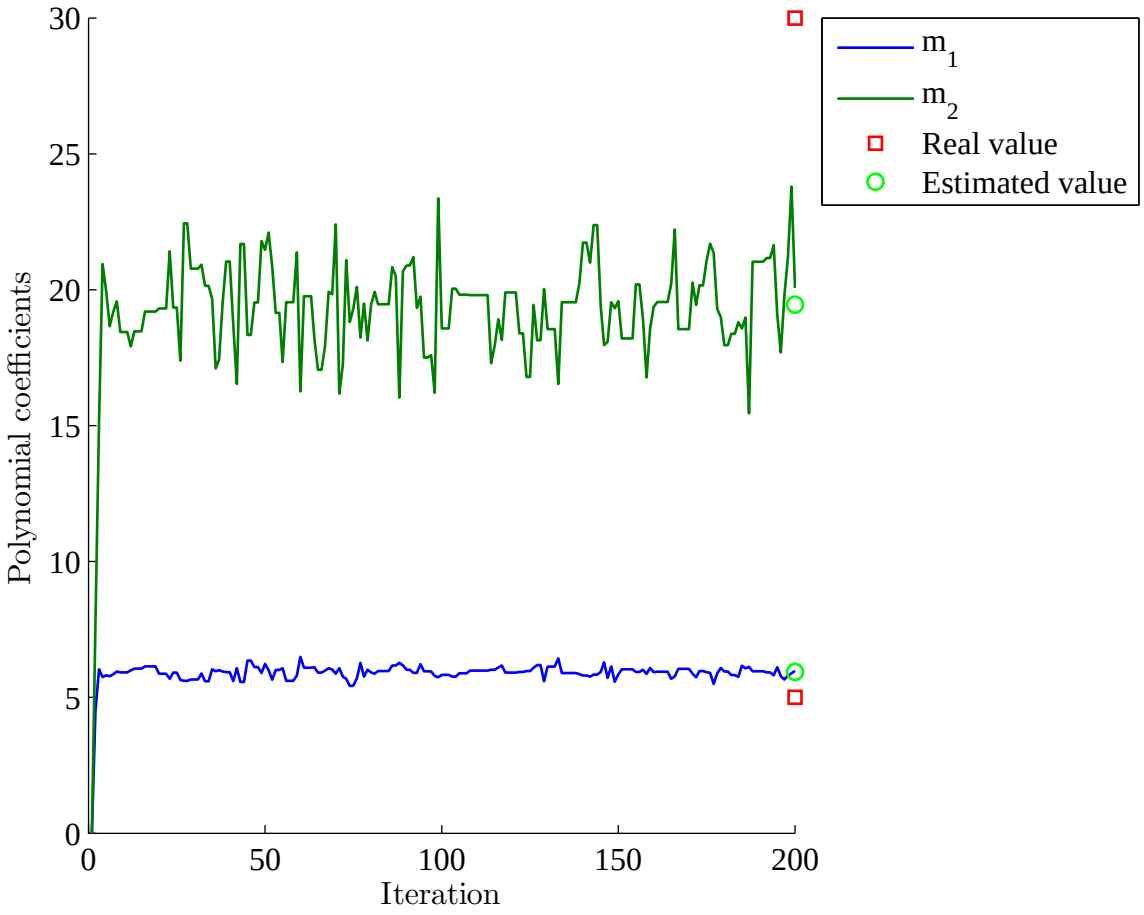

Figure 9.12: Signal `voice.wav` and distortion with polynomial inverse: Convergence of polynomial coefficients.

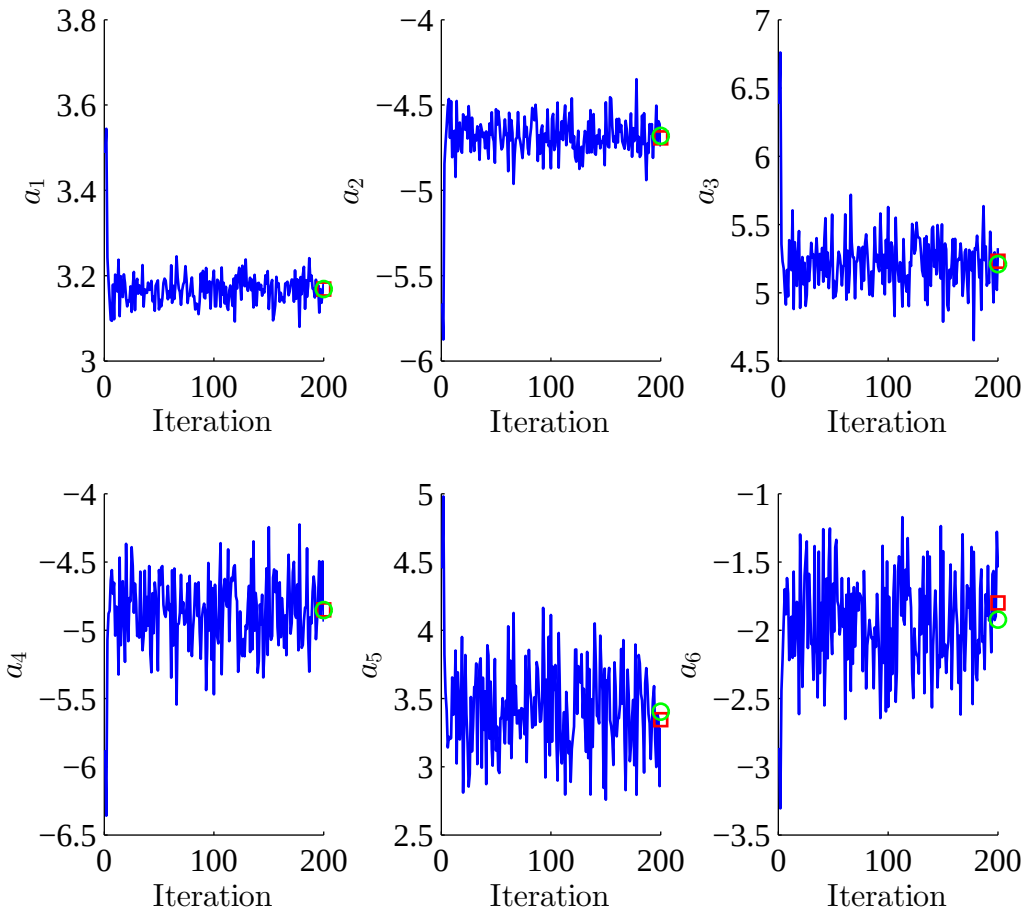

Figure 9.13: Signal `voice.wav` and distortion with polynomial inverse: Convergence of the first 6 AR model coefficients.

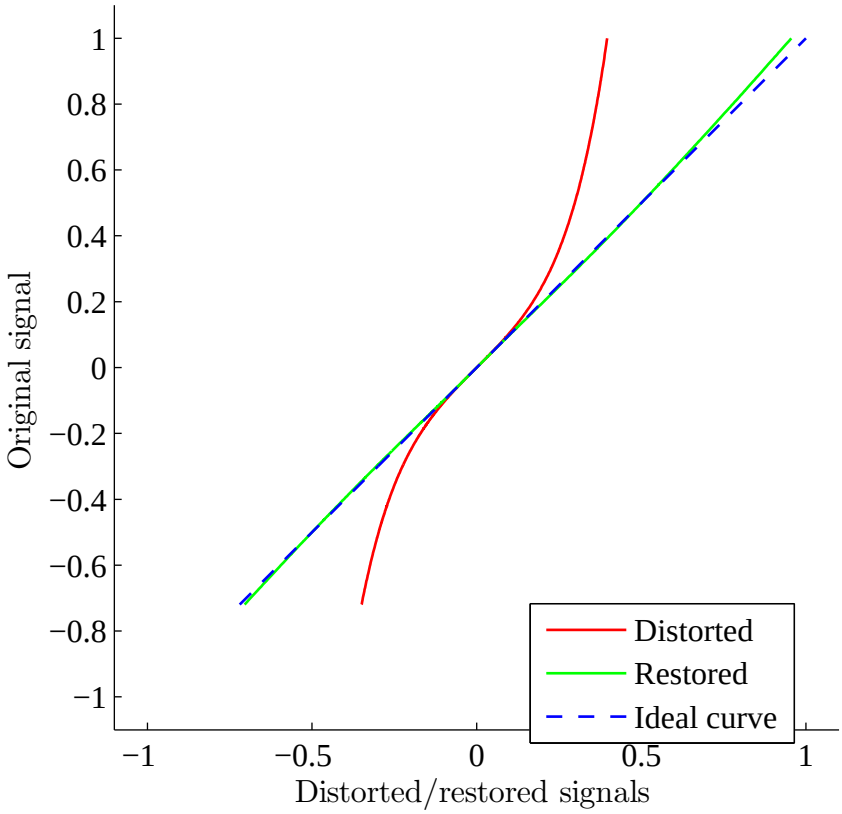

Figure 9.14: Signal `voice.wav` with piecewise linear distortion: Distorted and restored signals plotted against the original one.

By hearing the signals and comparing their Rnonlin grades we conclude that the method is also effective in this more realistic scenario, since no audible difference between the restored and original signals are noted. However, the estimation of parameters $\mathbf{a}$ and $\mathbf{m}$ is not as accurate as in the previous case. This was an expected behavior, since the underlying signals do not follow exactly an AR model.

An important point to be discussed is about the convergence of $\mathbf{m}$, in particular m_2. By looking at the convergence plots in Figures 9.6, 9.9 and 9.12, it seems that the posterior mean of m_2 was estimated very far from its real value, mainly in signal `voice.wav`. In fact the estimation was not so accurate, and a possible reason is discussed. In the previous case, when analyzing artificial signals, the largest entry of the original signal in absolute value was far greater than one, and now when dealing with real signals they have been normalized to limit the magnitude of its largest entry in absolute value to one. After applying the inverse of the polynomial $f(.)$ to the original signal, its time samples are now much smaller than one. This new distorted signal provides us only a little amount of useful information to estimate the high-order coefficients of $g(.)$, since when a small number is raised to a positive power, it becomes even smaller. Another possibility is that the AR model is not adequate to model the considered excerpt of the speech signal.

However, as we can note from hearing the signals or comparing their Rnonlin grades, this issue almost does not impact the perceived quality of the restored signals. As the distorted signals provide little information about the high-order coefficients of $g(.)$, the sensitivity of the signal to these coefficients is low, since they multiply very small numbers. This is a very important reason to not look only at the plots when dealing with real signals! Our ears and eyes perceive the information in very different ways, and our senses may be easily fooled.

9.1.3 Real signals and more general distortions

Finally, in this section we report the more realistic case of real signals distorted by more general distortions. The test signals are the same as the previous ones, but distorted by the function $f(x) = \arctan(\lambda x)/\lambda$, where the value of λ allows one to control the distortion level applied to the original signal while keeping the constraint of unitary derivative in the origin. The evaluated signals are the same used in the previous sections, and we considered $\lambda \in \{1, \ldots, 5\}$.

For each signal we considered a single block of length 1,000 centered at the respective maximum amplitude time sample. The values of M and P were fixed at 4 and 40, respectively, for all signals, since this choice of parameters showed good results in all cases. The parameters of the prior distributions were again $\sigma_m^2 = 10^{10}$, $\sigma_a^2 = 10^{10}$, $\alpha = \beta = 10^{-10}$. In Table 9.4, the Rnonlin grades for the distorted and

restored signals for each value of λ are shown. All signals are available in webpage [72].

Table 9.4: Real signals with more general distortions: Rnonlin grades for distorted and restored signals.

		`flute.wav`	`classical.wav`	`voice.wav`
$\lambda = 1$	Distorted	0.9730	0.9989	0.9852
	Restored	0.9999	1.0000	1.0000
$\lambda = 2$	Distorted	0.9288	0.9903	0.9554
	Restored	0.9999	1.0000	1.0000
$\lambda = 3$	Distorted	0.8948	0.9736	0.9281
	Restored	0.9999	1.0000	1.0000
$\lambda = 4$	Distorted	0.8694	0.9524	0.9058
	Restored	0.9999	1.0000	1.0000
$\lambda = 5$	Distorted	0.8486	0.9295	0.8880
	Restored	0.9999	1.0000	1.0000

We can conclude that the method is also efficient for distortions that do not follow the model, being then capable of removing distortions caused by real devices that deviate from the memoryless assumption. In informal hearings evaluation, no audible difference between restored and original signal was noted, in accordance to the Rnonlin grades.

9.2 Memoryless nonlinear distortions: Piecewise linear approximation

The piecewise linear approximation to the inverse of the nonlinear distortion was presented in Section 7.3. The results illustrated here expand and detail those shown in [6]. The performed tests are similar to those in Section 9.1.

9.2.1 Artificial signals and distortion following the model

A 1,000-time sample artificial signal following an AR model with poles of module 0.99 at frequencies $\pi/16$, $\pi/8$, and $\pi/4$ radians per time sample and its corresponding conjugate frequencies was generated. The variance of the excitation signal was chosen to be $\sigma_e^2 = 5 \times 10^{-6}$. The original signal was distorted by a piecewise linear function where each segment had slopes given by 1, 1/2, 1/4, 1/8, and 1/16, respectively. Therefore, we wish to recover vector $\mathbf{m} = [2 \quad 4 \quad 8 \quad 16]^T$, vector $\mathbf{a} = [5.1713 \quad -11.7727 \quad 15.1104 \quad -11.5384 \quad 4.9675 \quad -0.9415]^T$, and the variance of the excitation signal $\sigma_e^2 = 5 \times 10^{-6}$.

The single block of 1,000 time samples was analyzed at once. Therefore the values of N, L and B were, respectively, 1,000, 1,000, and 1. The parameters of the prior distributions were $\sigma_m^2 = 10^{10}$, $\sigma_a^2 = 10^{10}$, $\alpha = \beta = 10^{-10}$. The values of M and P were the correct ones, that is, $M = 4$ and $P = 6$.

The Gibbs sampler described in Section 7.3 was run for 100 iterations, lasting 0.099 s each, in average. The burn-in time was 50 iterations long.

Figures 9.15, 9.16, 9.17, 9.18 and 9.19 show, respectively, the convergence of the angular coefficients, a "zoom" on its last 50 iterations, the convergence of **a** and two plots comparing the original, distorted and restored signals. Tables 9.5 and 9.6 also compare the real and estimated values for parameters **a** and **m**, respectively. On the convergence plots, the red square and the green circle always denote the real and estimated values, respectively.

Note that the typical amplitude of the restored signals here are quite different from the ones presented in the previous section, and a reason is provided now. Recall that when developing the theory of piecewise linear approximation in Section 7.3 we considered that the distorted signal lies in $[-1, 1]$, and in order to fulfill this hypothesis some scaling on the distorted signal was necessary. Because of the considered hypothesis on the distorting nonlinearity, the distorted signal tends to have a smaller dynamic range in relation to the undistorted one. Therefore, when applying the piecewise linear function to the distorted signal its dynamic range is amplified, yielding the large amplitude seen here. However, as explained before, this scaling does not impact the Rnonlin grade when dealing with real signals.

From these tests, we can then conclude that the algorithm is capable of identifying correctly the desired parameters in this simple scenario, and assess aspects of its accuracy and convergence.

Table 9.5: Artificial signal and piecewise linear distortion: Comparison of real and estimated values of the AR model coefficients.

	Real	Estimated
a_1	5.1713	5.1510
a_2	-11.7727	-11.6790
a_3	15.1104	14.9289
a_4	-11.5384	-11.3530
a_5	4.9675	4.8679
a_6	-0.9415	-0.9192

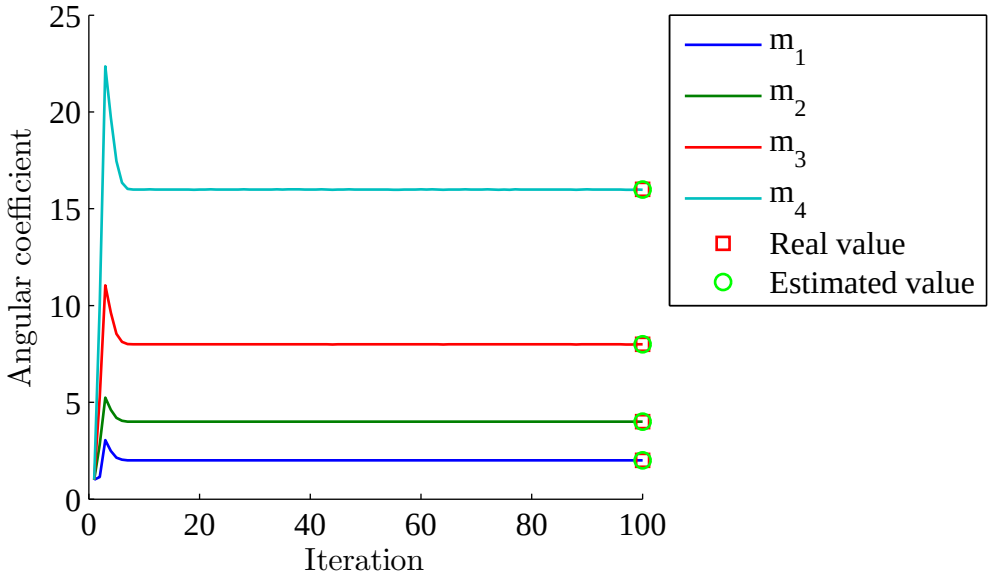

Figure 9.15: Artificial signal and piecewise linear distortion: Convergence of angular coefficients.

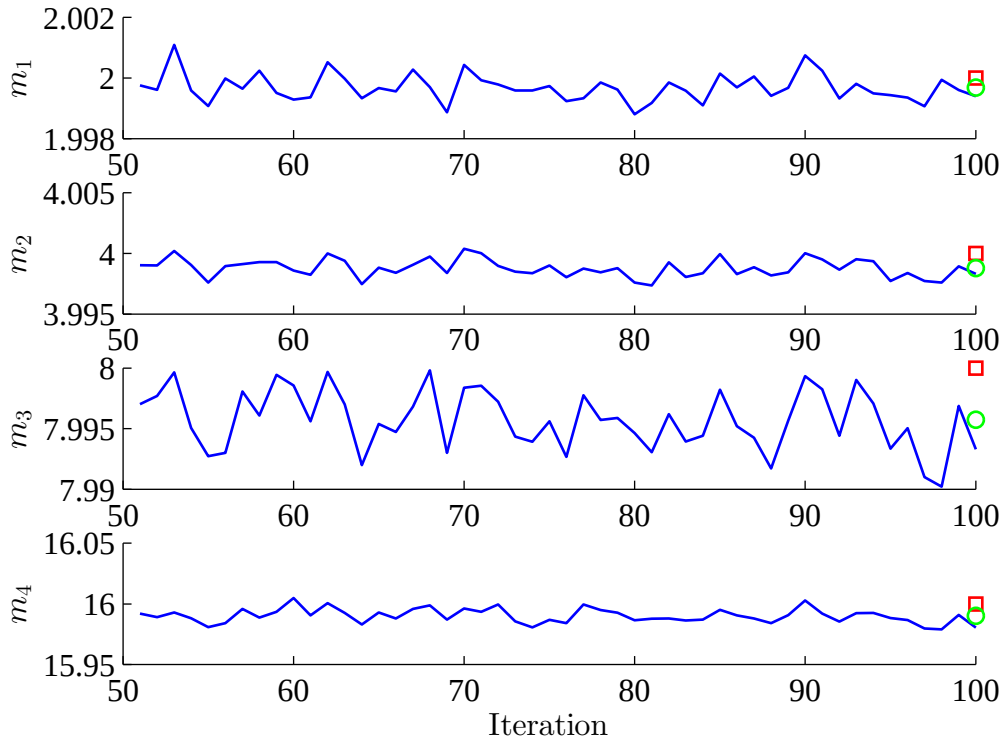

Figure 9.16: Artificial signal and piecewise linear distortion: Convergence of angular coefficients after the burn-in time.

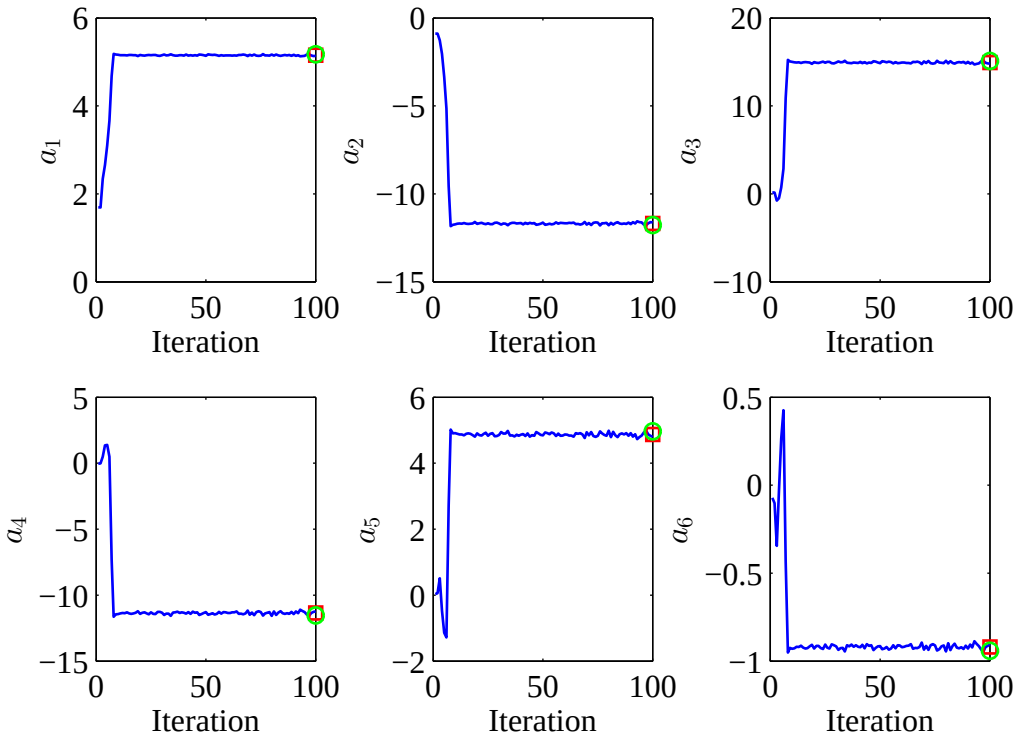

Figure 9.17: Artificial signal and piecewise linear distortion: Convergence of the AR model coefficients.

Table 9.6: Artificial signal and piecewise linear distortion: Comparison of real and estimated values of the angular coefficients.

	Real	Estimated
m_2	2	1.9997
m_3	4	3.9988
m_4	8	7.9957
m_5	16	15.9902

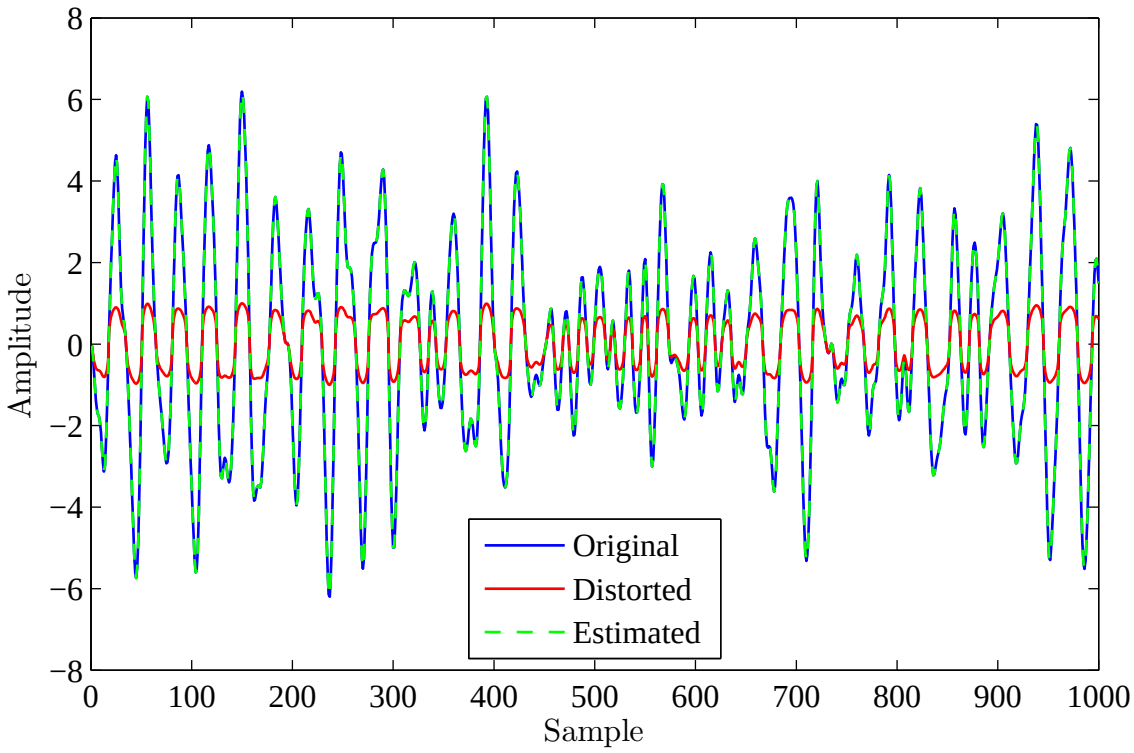

Figure 9.18: Artificial signal and piecewise linear distortion: Comparison of original, distorted and restored signals.

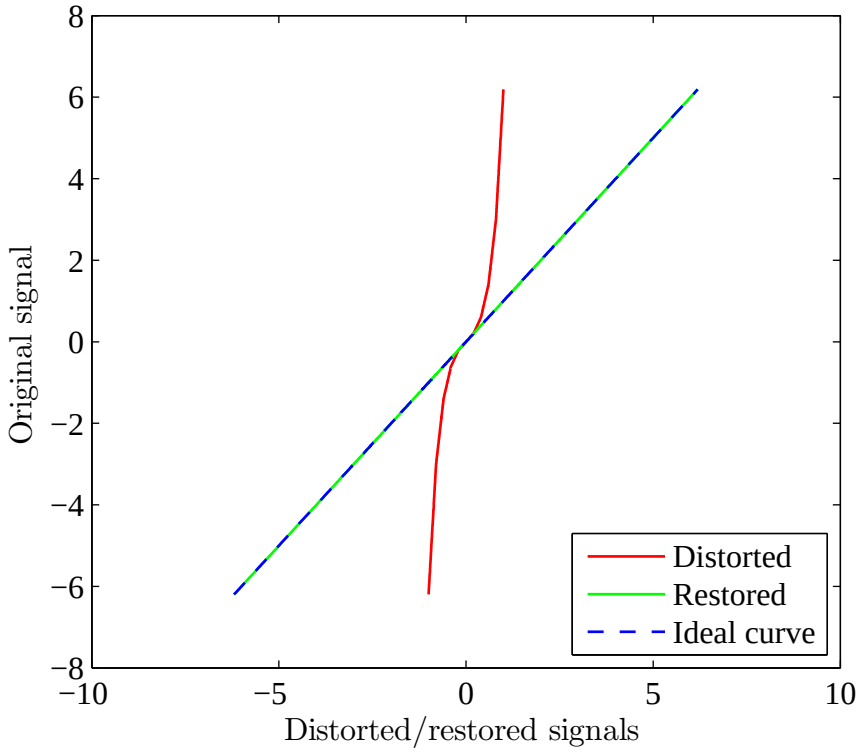

Figure 9.19: Artificial signal and piecewise linear distortion: Distorted and restored signals plotted against the original one.

9.2.2 Real signals and distortion following the model

The next step is to consider real signals, distorted by functions following the piecewise linear model. The tested signals were the same as in Section 9.1, distorted by a piecewise linear function where each segment has slopes given by 1, 1/2, 1/4, 1/8, and 1/16, respectively, as in the previous case, implying that the desired vector $\mathbf{m}$ is given by

$$\mathbf{m} = [2 \quad 4 \quad 8 \quad 16]^T. \tag{9.2}$$

For each of the three signals tested, three figures are shown: the first one displays the convergence of $\mathbf{m}$ (Figures 9.20, 9.23 and 9.26), the second one the convergence of the first 6 coefficients of the respective AR model (Figures 9.21, 9.24 and 9.27) and the third one the distorted and restored signals against the original ones (Figures 9.22, 9.25 and 9.28).

The value of P was chosen to be 40 for signals `flute.wav` and `voice.wav`, and 30 for the signal `classical.wav`, since this choice presented better results, and we compare the estimated value of $\mathbf{a}$ returned by the Gibbs sampler with those estimated from the original signal via the covariance method. We can see that the estimate of $\mathbf{m}$ and $\mathbf{a}$ is not as accurate as in the artificial scenario, but for all three signals tested the result is satisfactory.

The restored signals are compared with the original one using the Rnonlin tool, and the respective grades are displayed in Table 9.7.

We conclude that the method is capable of dealing with real signals, where the AR model is an approximation for the underlying signal. No audible differences between restored and original signals are noticed, in accordance to their respective Rnonlin grades.

Table 9.7: Real signals with piecewise linear distortion: Rnonlin grades for distorted and restored signals.

	`flute.wav`	`classical.wav`	`voice.wav`
Distorted	0.7404	0.7930	0.8069
Restored	0.9999	0.9999	0.9995

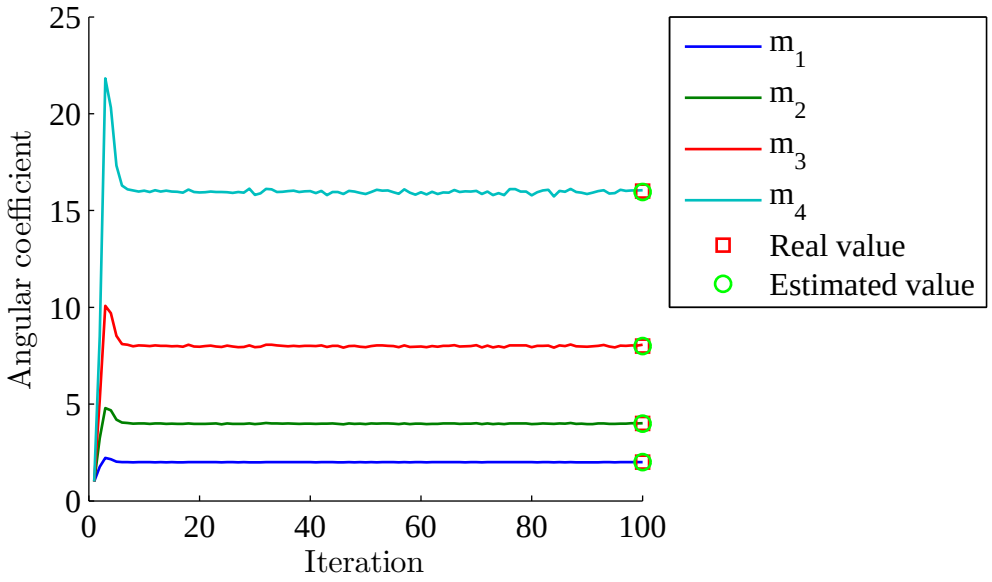

Figure 9.20: Signal `flute.wav` with piecewise linear distortion: Convergence of angular coefficients.

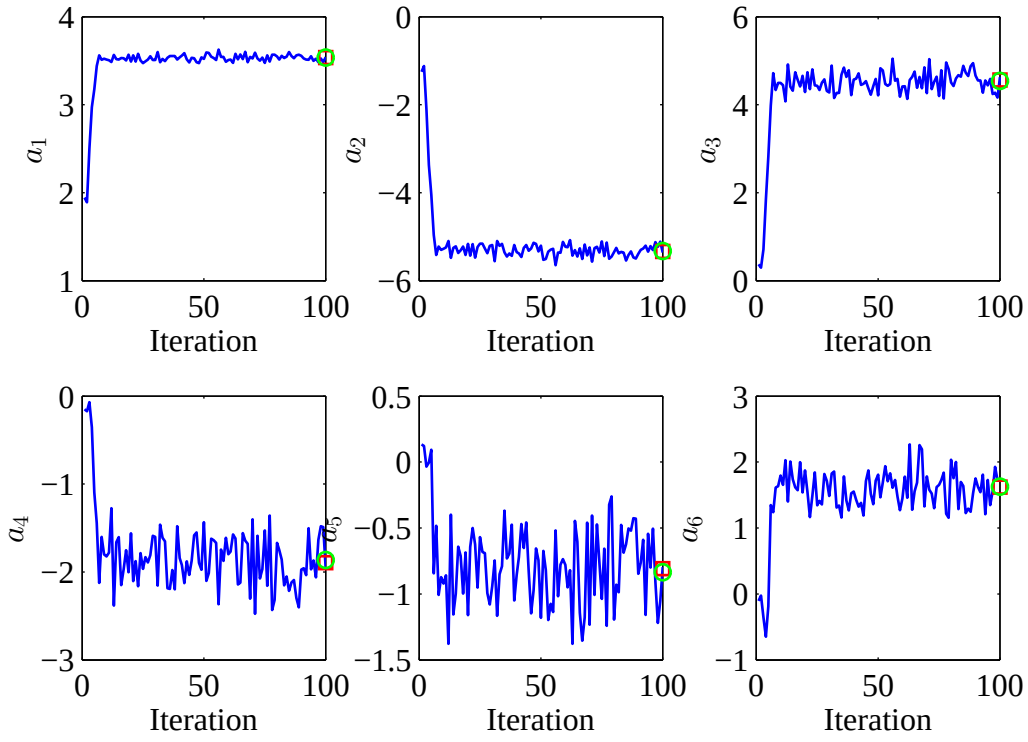

Figure 9.21: Signal `flute.wav` with piecewise linear distortion: Convergence of the first 6 AR model coefficients.

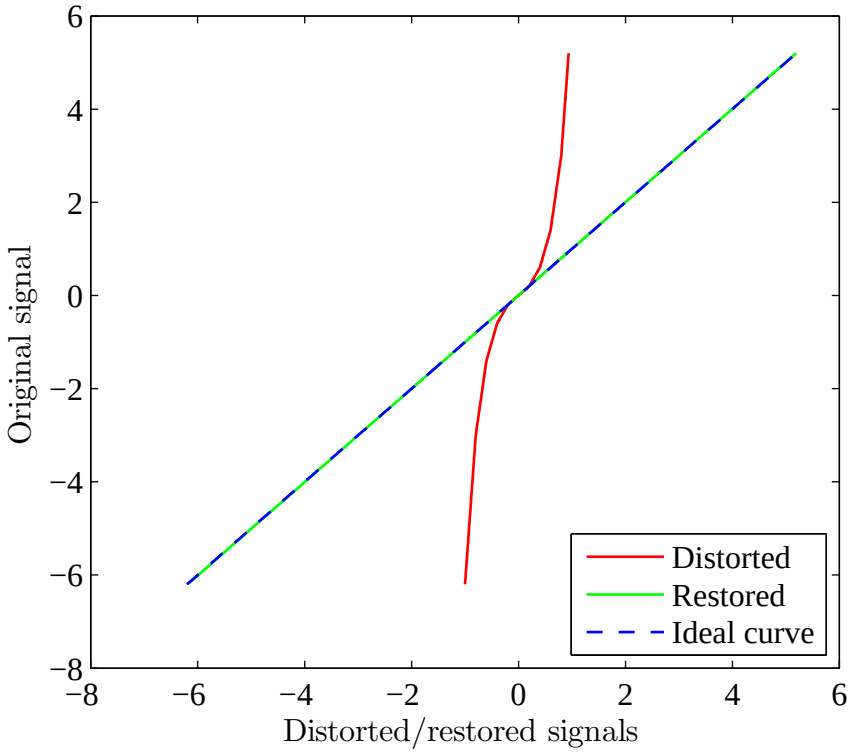

Figure 9.22: Signal `flute.wav` with piecewise linear distortion: Distorted and restored signals plotted against the original one.

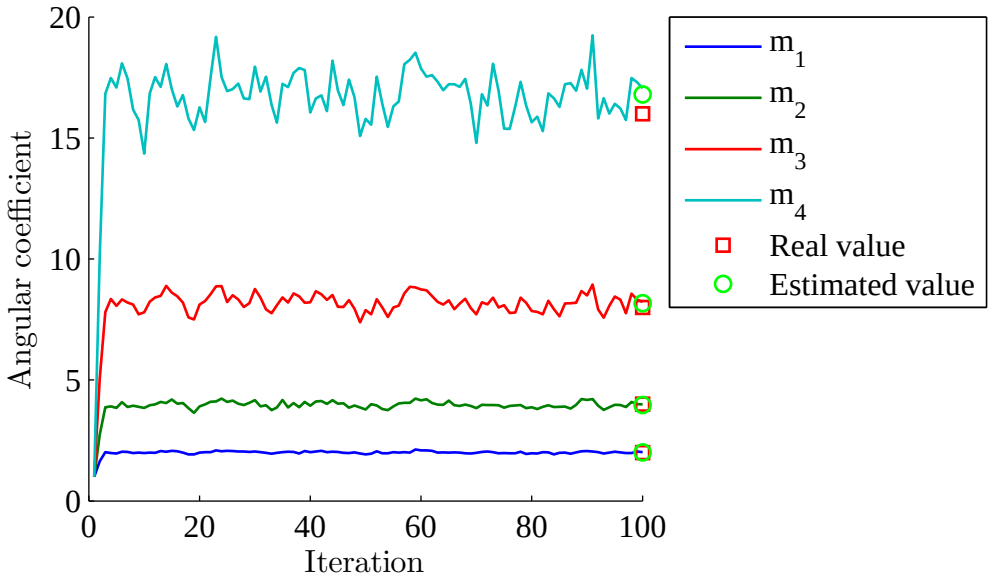

Figure 9.23: Signal `classical.wav` with piecewise linear distortion: Convergence of angular coefficients.

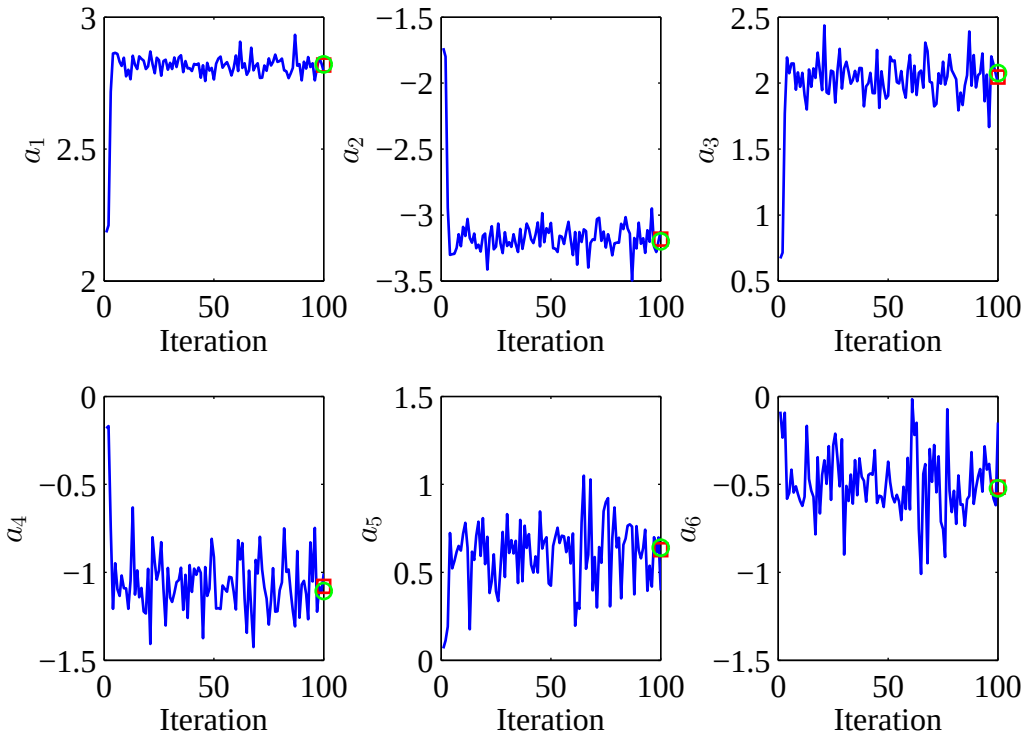

Figure 9.24: Signal `classical.wav` with piecewise linear distortion: Convergence of the first 6 AR model coefficients.

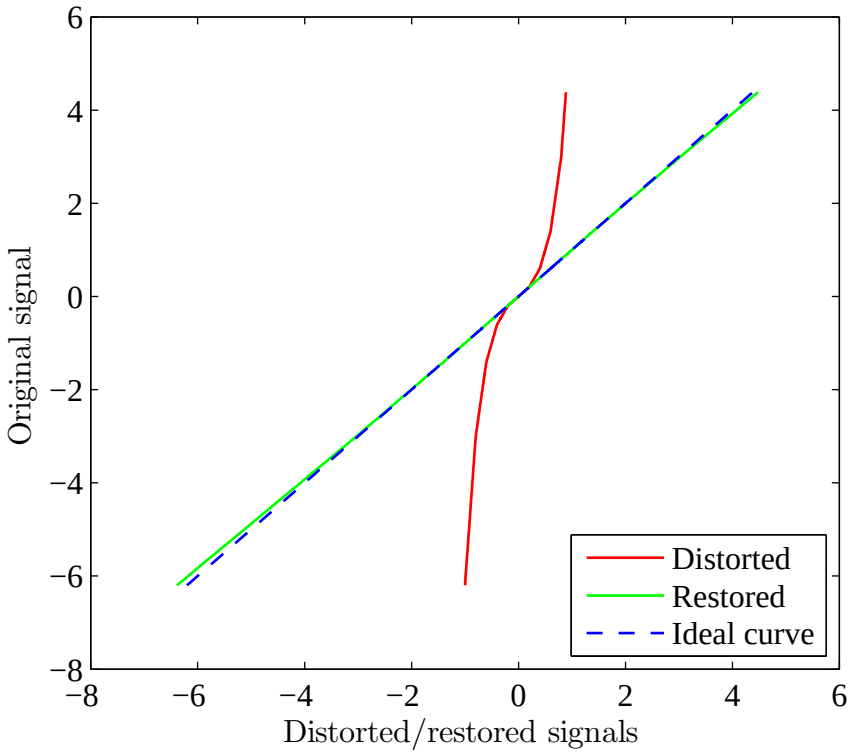

Figure 9.25: Signal `classical.wav` with piecewise linear distortion: Distorted and restored signals plotted against the original one.

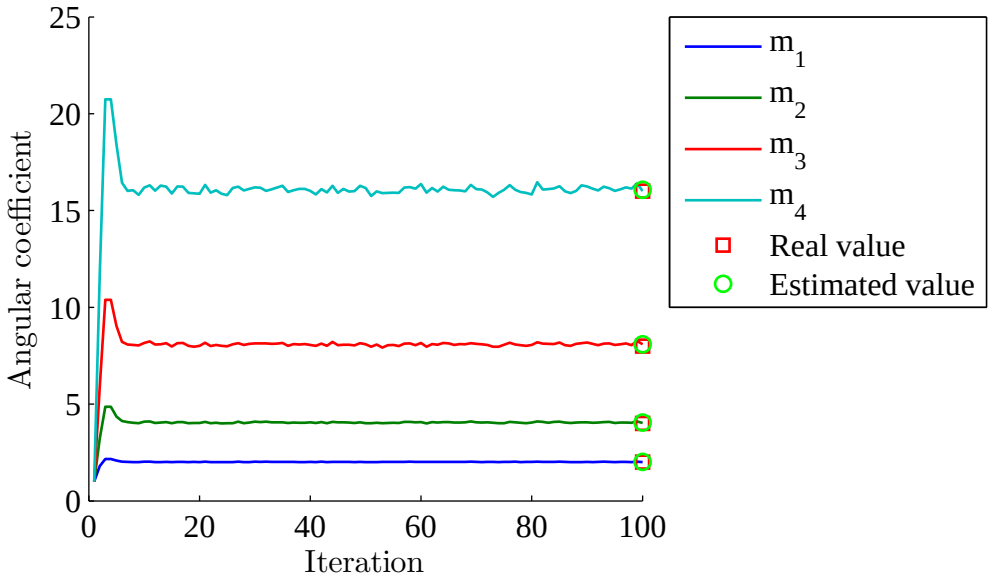

Figure 9.26: Signal `voice.wav` with piecewise linear distortion: Convergence of angular coefficients.

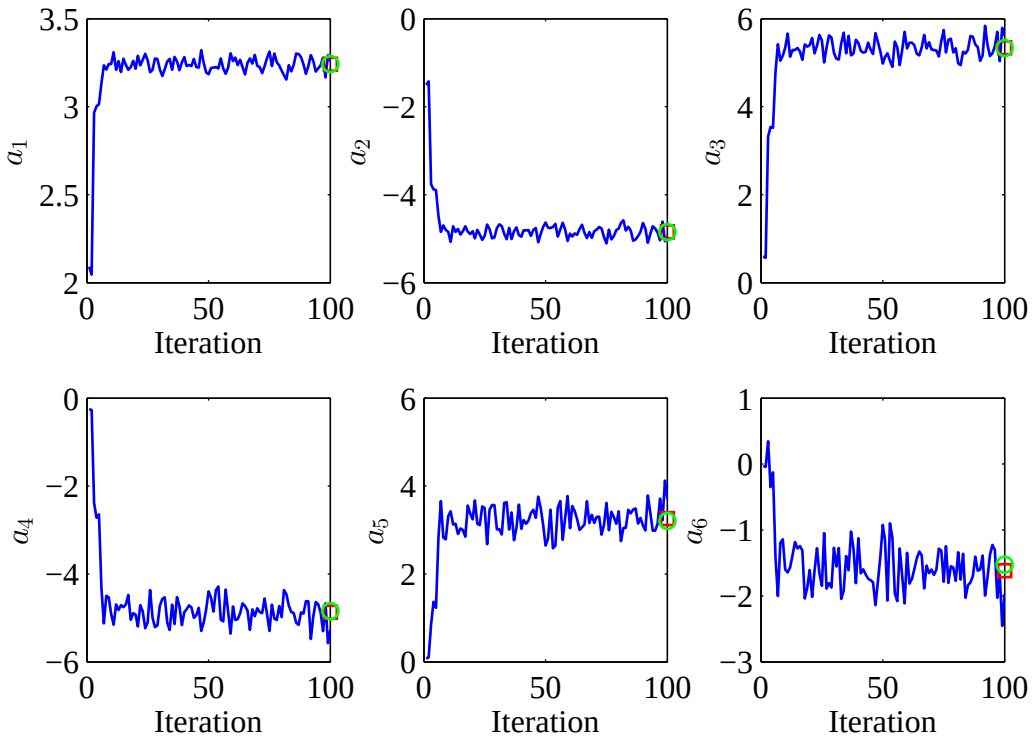

Figure 9.27: Signal `voice.wav` with piecewise linear distortion: Convergence of the first 6 AR model coefficients.

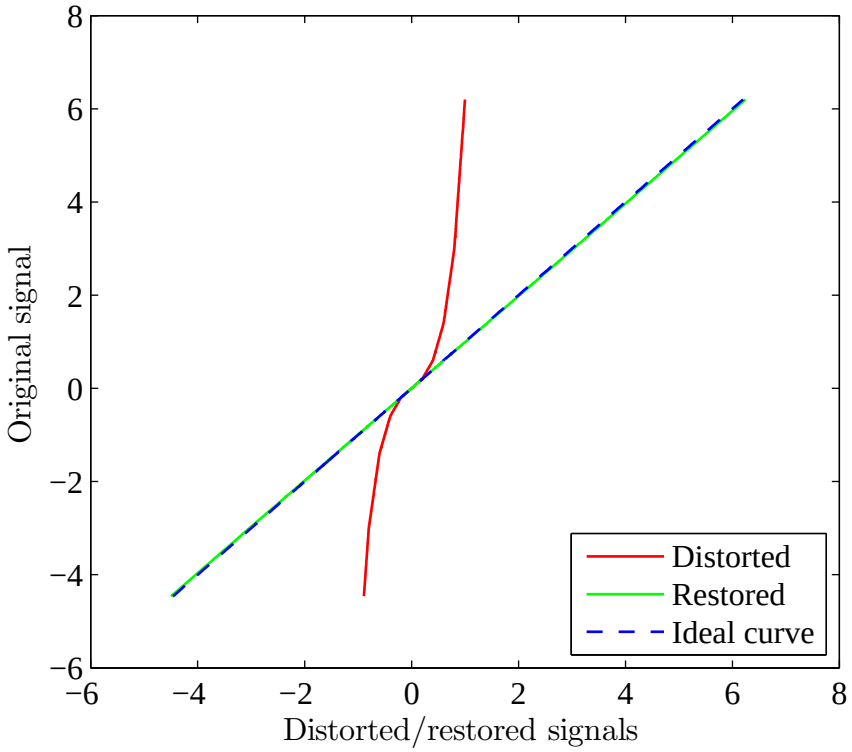

Figure 9.28: Signal `voice.wav` with piecewise linear distortion: Distorted and restored signals plotted against the original one.

9.2.3 Real signals and smooth distortions

Finally, in this section we report the more realistic case of real signals distorted by smooth distortions. The test signals are the same as before, but distorted by the function $f(x) = \arctan(\lambda x)/\lambda$, where the value of λ allows one to control the distortion level applied to the original signal while keeping the constraint of unitary derivative in the origin. We considered $\lambda \in \{1, \ldots, 5\}$.

The values of N, M and P, as well as the mean time per iteration, number of iterations and burn-in time for each analyzed signal are shown in Table 9.8. In each case, the value of N represent the size of the whole signal, split in contiguous blocks of 1,000 time samples. Note that it was necessary to analyze more than one block of signal and use very high M in order to obtain good results. A possible reason for these two facts will be presented soon. The parameters of the priors distributions were again $\sigma_m^2 = 10^{10}$, $\sigma_a^2 = 10^{10}$, $\alpha = \beta = 10^{-10}$. Since the values of M and P are quite high, I chose not to plot several figures with the convergence of the variables, but only display Table 9.9, where the Rnonlin grades for the distorted and restored signals for each value of λ are shown. All signals are available in webpage [72].

Table 9.8: Parameters of the algorithm for each signal.

	`flute.wav`	`classical.wav`	`voice.wav`
Mean time per iteration	1.40 s	6.58 s	1.47 s
Number of iterations	100	50	50
Burn-in time	50	30	30
N	10,000	30,000	10,000
M	90	100	90
P	40	10	40

Table 9.9: Rnonlin grades for distorted and restored signals.

		`flute.wav`	`classical.wav`	`voice.wav`
$\lambda = 1$	Distorted	0.9730	0.9989	0.9852
	Restored	0.9995	0.9996	0.9957
$\lambda = 2$	Distorted	0.9288	0.9903	0.9554
	Restored	0.9994	0.9998	0.9958
$\lambda = 3$	Distorted	0.8948	0.9736	0.9281
	Restored	0.9990	0.9998	0.9954
$\lambda = 4$	Distorted	0.8694	0.9524	0.9058
	Restored	0.9977	0.9999	0.9944
$\lambda = 5$	Distorted	0.8486	0.9295	0.8880
	Restored	0.9945	0.9999	0.9932

We conclude that the proposed method is capable of dealing with real signals and smooth distortions, although some manual tuning of parameters, specially the length of the considered signal (related to B) and the number of linear functions used in the approximation (given by M), must be made in order to reach a good of result. Since we have access to the original signal, we are able to compare the restored and original signals in order to judge the quality of the obtained result; in a more realistic scenario, this judgment should be performed perceptually by the user. Despite this, no audible differences between restored and original signals were noted, as confirmed by their respective Rnonlin grades.

We now return to the discussion about the requirement of a quite high M and several blocks in order to accurately estimate the smooth nonlinear distortion. When first evaluating this method with real signals, we noted that a small number of segments (e.g. $M \approx 7$) was not enough to approximate a smooth distortion, contradicting our initial intuition. We then tried larger values of M, but this yielded poorer estimates of $\mathbf{m}$. In order to provide more data to the algorithm, the method was extended to several contiguous blocks of signal (each one with its respective AR parameters, in order to fulfill the stationarity hypothesis in each block), and a combination of a high value of M and several blocks showed the good results reported

here.

After investigating this counterintuitive behavior of M, we noted that the approximation error due to the piecewise linear approximation to a smooth function induces something similar to additive white noise in the original signal, impairing then the estimation of all quantities of interest. When the value of M increases, the magnitude of this error gets smaller, implying a higher SNR, which above some value no longer impacts the estimation of the AR model parameters. It is interesting to note that the polynomial model also introduces an error, but it tends to be smoother and less harmful than the error in the piecewise linear approximation.

9.3 Further tests investigating the effects of model orders

In the previous sections we performed tests to evaluate the capability of the proposed methods to deal with real signals and distortions not following the respective models (polynomial or piecewise linear). However, when dealing with artificial signals and distortions following the respective models, the orders of both the AR model and the distortion were assumed known; moreover, when dealing with real signals a fixed value for these orders were considered, and no considerations were made about the sensitivity of the obtained results with respect to these variables. In this section we perform additional tests in order to study this behavior. We thank Professor Paulo Esquef for suggesting these tests[6], and Professor Ralph Silva for indicating me the adequate statistical tools to employ here.

More specifically, the following verifications are performed here:

- Test the significance of a particular polynomial coefficient via an approximate credence interval (also called *credible interval*), estimated from the obtained samples from the posterior distribution. For a 95% credence interval for some parameter, we assess that with probability 0.95 the correct value for this parameter lies within this interval[7]; if this interval contains the value zero, we have some statistical evidence that this parameter is irrelevant to the model. One expects that when the distortion possesses a polynomial inverse, the additional coefficients not present in the actual function will have low significance, which means that zero will be in their respective 95% approximate credence interval.

[6] And a lot of other improvements scattered along the text!

[7] Note that this probability is in the parameter space since we are in a Bayesian context, and can be understood as our degree of belief in the occurrence of this fact. For a more complete discussion about credence and confidence intervals, its counterpart in the frequentist context, see [10, 11].

Since the considered posterior distributions are unimodal and symmetric (via inspection of the histogram constructed from the MCMC samples), the 95% approximate credence interval was constructed by estimating the 0.025 and 0.975 quantiles of the posterior distribution.

- Another way to verify the significance of some parameter of the model is via the *deviance information criterion* (DIC) [73, 74]. This criterion is one possible way to evaluate which model is more adequate to a given dataset, specifically when samples from the variables within the model are obtained via a Gibbs sampler. For a more extensive discussion, see [75]. Basically, if $p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M})$ denotes the likelihood of the observed data with respect to the parameters $\boldsymbol{\theta}$ in the model $\mathcal{M}$, define the *deviance* as

$$D(\boldsymbol{\theta}) = -2 \log p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M}), \tag{9.3}$$

and define the DIC as

$$\mathrm{DIC}(\mathcal{M}) = \underbrace{\mathbb{E}[D(\boldsymbol{\theta})|\mathbf{y}, \mathcal{M}] - D(\mathbb{E}[\boldsymbol{\theta}|\mathbf{y}, \mathcal{M}])}_{=C_{\mathcal{M}}} + \underbrace{\mathbb{E}[D(\boldsymbol{\theta})|\mathbf{y}, \mathcal{M}]}_{=G_{\mathcal{M}}}. \tag{9.4}$$

The term $C_{\mathcal{M}}$ is responsible for the model complexity, and $G_{\mathcal{M}}$ evaluates the goodness of fit of the model, as argued in [75]. If $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L$ are samples of the posterior distribution for $\boldsymbol{\theta}$ (after the burn-in time of the corresponding Markov chain), the DIC can be approximated as

$$\mathrm{DIC}(\mathcal{M}) \approx \frac{2}{L} \sum_{i=1}^{L} D(\boldsymbol{\theta}_i) - D\left(\frac{1}{M} \sum_{i=1}^{L} \boldsymbol{\theta}_i \right). \tag{9.5}$$

Models with smaller DIC should be preferred to models with larger DIC. In our case, a model is simply a choice of orders to the polynomial and AR model.

The autocorrelation function of the chains generated via the Gibbs sampler were also estimated, in order to analyze its internal correlation. For all the cases we observed that the samples within each chain are very uncorrelated, supporting the fact that the posterior mean is well estimated by the sample mean of all the samples after the burn-in time.

Only the polynomial approximation of the inverse of the distorting function was considered in these tests, since it was already noticed that the piecewise linear approximation is more sensitive to the choice of both the AR model order and the number of linear coefficients.

9.3.1 Artificial signal

As before, we consider an artificial signal with 1,000 time samples following the AR model with poles at frequencies $\pi/16$, $\pi/8$ and $\pi/4$ radians per time sample and their respective conjugated frequencies, each one with module 0.99, variance of excitation signal equal to $\sigma_e^2 = 5 \times 10^{-6}$ and distorted by the inverse of the function

$$g(y) = y + 5y^3 + 30y^5. \tag{9.6}$$

In this case, the correct values of P and M are given by 6 and 2, respectively. Tests were performed by considering the combinations of the values below:

$$P_{\mathrm{w}} \in \{4 \;\; 5 \;\; 6 \;\; 7 \;\; 8\} \tag{9.7}$$

$$M_{\mathrm{w}} \in \{2 \;\; 3\}, \tag{9.8}$$

the subscript "w" being used to denote a possibly wrong value for the respective variable. For each combination of these parameters 1,000 iterations of the Gibbs sampler was run, with a burn-in period of 200 iterations, the parameters being then estimated as the mean after the burn-in time.

In Table 9.10 we can compare the DIC value for all possible combinations of P_{w} and M_{w}. The smallest value was obtained when $P_{\mathrm{w}} = 6$ and $M_{\mathrm{w}} = 3$, but the DIC value obtained by considering $M_{\mathrm{w}} = 2$, the correct value for M, is very close to the minimum. In Table 9.11 we can compare the estimated values for $\mathbf{m}$ for all possible values of P_{w} and by considering $M_{\mathrm{w}} = 2$. We conclude that the method is quite insensitive to the choice of the order of the AR model, estimating accurately the values of polynomial coefficient even when the order of the AR model used is not the correct one. And finally in Table 9.12 we show the estimated values for m_3 for all possible values of P_{w} and by considering $M_{\mathrm{w}} = 3$. The behavior of the first two coefficients was very similar to the one presented in Table 9.11 and was omitted to avoid redundancy. Note that all credible intervals contain the value zero, indicating that the additional coefficient is not significant.

Table 9.10: Values of DIC for every evaluated combination of P_{w} and M_{w}.

		P_{w}				
		4	5	6	7	8
M_{w}	2	37,402	-1587	-10,994	-10,985	-10,976
	3	37,689	-1556	-10,995	-10,985	-10,975

Table 9.11: Estimated values for $\mathbf{m}$ and its respective 95% credence interval, for all values of P_{w} and $M_{\mathrm{w}} = 2$.

		m_1^{EST}	95% credence interval	m_2^{EST}	95% credence interval
P_{w}	4	4.9861	(4.8841, 5.0791)	30.0986	(29.6090, 30.5911)
	5	4.9979	(4.9682, 5.0255)	30.0145	(29.8685, 30.1680)
	6	4.9994	(4.9946, 5.0043)	30.0074	(29.9808, 30.0356)
	7	4.9996	(4.9947, 5.0045)	30.0063	(29.9790, 30.0323)
	8	4.9996	(4.9949, 5.0048)	30.0062	(29.9782, 30.0345)

Table 9.12: Estimated values for m_3 and its respective 95% credence interval, for all values of P_{w} and $M_{\mathrm{w}} = 3$.

		m_3^{EST}	95% credence interval
P_{w}	4	1.4773	(-3.5764,6.7234)
	5	0.5853	(-1.3876, 2.6950)
	6	0.3493	(-0.0696, 0.7784)
	7	0.3531	(-0.0471, 0.7872)
	8	0.3366	(-0.0951, 0.7401)

9.3.2 Real signal and distortion following the model

The evaluated signal was `flute.wav`, distorted by the inverse of the function

$$g(y) = y + 5y^3 + 30y^5. \tag{9.9}$$

An excerpt of 1,000 time samples around the time sample with greatest amplitude was given as input to the algorithm.

A preliminary test was performed with the original undistorted signal, by estimating the coefficients of its AR model via the Gibbs sampler (a description of this procedure can be found in [37], and can be easily derived from the computations in Chapter 7 by not considering the nonlinear distortion). Recall that in Chapter 9 the considered value of P for this signal was $P = 40$, and now we estimated its AR model coefficients for all values of P between 1 and 100 and computed the respective DIC. The (quite surprising) result is shown in Figure 9.29. Note that the minimum value of the DIC is obtained when $P = 10$, a value much smaller than we typically use. From the statistical viewpoint, this result says that $P = 10$ is the order that best balances the goodness of fit of the model to the signal with the amount of coefficients being used. Motivated by this, the next performed test was to estimate the polynomial coefficients from the distorted signal, by varying its order and the order

of the AR model, considering the following possible values for these parameters:

$$P \in \{5 \;\; 10 \;\; 15 \;\; 35 \;\; 40 \;\; 45\} \tag{9.10}$$

$$M_{\mathrm{w}} \in \{2 \;\; 3 \;\; 4\}. \tag{9.11}$$

Note that the considered values for P were chosen to vary around the smallest DIC value and also around the previously used value of $P = 40$. Note also that the notation P_{w} is not being used now, since there is no correct value of P to compare with.

In Table 9.13 we can compare the value of the DIC for the possible combinations of P and M_{w}. We see that all the values in the table are quite similar, and the smallest one is obtained with $P = 15$ and $M_{\mathrm{w}} = 3$. One then expects that the restored signals in all the cases are similar, since the balance between goodness of fit and number of parameters is also similar.

This expectation is confirmed in Tables 9.14, 9.15, and 9.16, where we can compare the estimated coefficients and their respective credence intervals for all possible values of P and for $M_{\mathrm{w}} = 2, 3$, and 4, respectively. In all the cases the first two coefficients are accurately estimated, and in the majority of the cases, the credence interval for the additional coefficients contains the value zero, indicating a possible non-relevance of these coefficients to the model. A notable exception is the coefficient m_3 when considering $M_{\mathrm{w}} = 3$ and for $P = 35, 40$, and 45: note that in these cases, exactly the order of the AR model we usually employ, the credence intervals do not contain zero, despite being quite large.

Finally, Table 9.17 explores the obtained data from the perceptual viewpoint, showing the Rnonlin grades for the restored signals for all the evaluated combinations of P and M_{w}. Since all these grades are above 0.99, all the restored signals sound well and very similar to the original one.

From the facts here presented, there is a good indication that the method is robust when dealing with uncertainties in the order of the AR model and in the number of estimated polynomial coefficients.

Table 9.13: Values of DIC for the evaluated combinations of P and M_{w}, for signal `flute.wav`.

		P					
		5	10	15	35	40	45
M_{w}	2	-9761.7	-9929.8	-9946.7	-9817.3	-9793.1	-9749.8
	3	-9767.5	-9940.5	-9967.2	-9853.0	-9828.9	-9785.2
	4	-9768.2	-9939.3	-9968.8	-9849.9	-9826.0	-9782.7

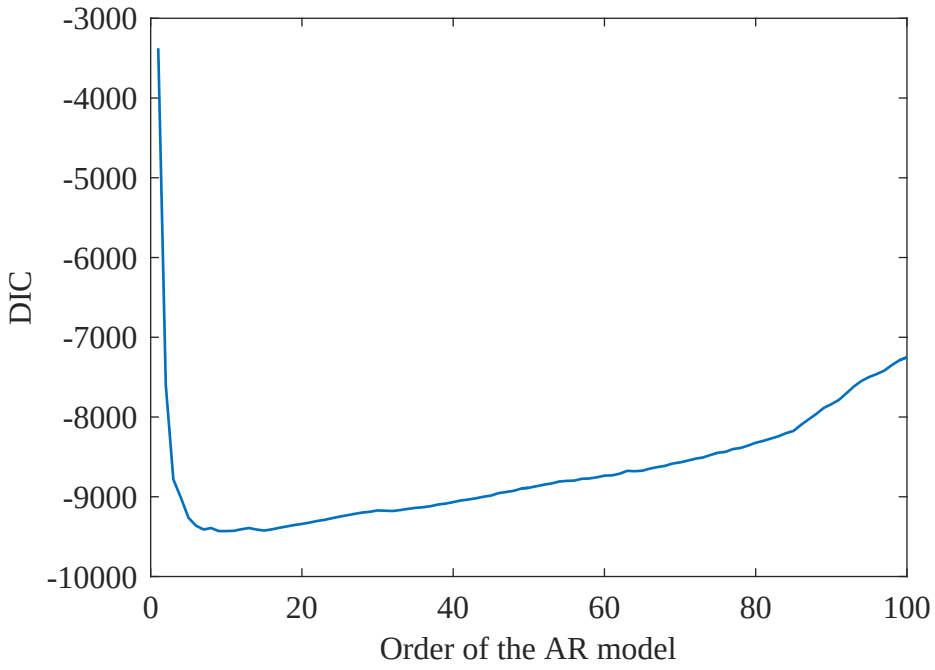

Figure 9.29: Values of DIC obtained by estimating the AR coefficients of signal `flute.wav` from the undistorted signal.

Table 9.14: Estimated values for $\mathbf{m}$ with $M_{\mathrm{w}} = 2$ and respective credence intervals for signal `flute.wav`.

P	m_1	m_2
5	4.9641 (4.4021, 5.5457)	21.8542 (17.1972, 26.1374)
10	4.7690 (4.3019, 5.2390)	26.1534 (21.4162, 30.7603)
15	4.8365 (4.4258, 5.2511)	27.3262 (22.8975, 31.7114)
35	4.9513 (4.5450, 5.3612)	25.7907 (22.0712, 30.0611)
40	4.9165 (4.5165, 5.2941)	26.4759 (22.4469, 30.6645)
45	4.9087 (4.5289, 5.2923)	26.5360 (22.7767, 30.7638)

Table 9.15: Estimated values for $\mathbf{m}$ with $M_{\mathrm{w}} = 3$ and respective credence intervals for signal `flute.wav`.

P	m_1	m_2	m_3
5	4.8179 (4.2234, 5.4139)	26.3430 (16.6037, 35.2139)	-26.5399 (-72.2440, 18.1854)
10	4.6576 (4.1690, 5.1213)	29.9602 (22.4057, 36.8454)	-27.8584 (-63.2952, 10.8280)
15	4.6882 (4.2833, 5.1597)	31.3323 (24.9481, 37.1822)	-33.0040 (-65.8360, 2.4175)
35	4.7261 (4.3170, 5.1231)	31.5246 (26.4135, 36.9292)	-46.5278 (-73.6289, -20.3166)
40	4.7301 (4.3650, 5.1077)	32.4659 (27.2651, 37.7625)	-48.5663 (-74.0608, -19.8508)
45	4.7427 (4.4092, 5.0880)	31.9574 (26.9208, 36.9537)	-45.6351 (-70.3939, -17.6893)

Table 9.16: Estimated values for $\mathbf{m}$ with $M_{\mathrm{w}} = 4$ and respective credence intervals for signal `flute.wav`.

P	m_1	m_2	m_3	m_4
5	4.9351 (4.3660, 5.5094)	17.2693 (5.1151, 29.4901)	120.6199 (-46.000, 277.30)	-582.8433 (-1172.0, 40.700)
10	4.7096 (4.1709, 5.2288)	22.7309 (12.5394, 32.8604)	102.4117 (-23.700, 231.10)	-520.8744 (-1000.4, -11.500)
15	4.7039 (4.3006, 5.1529)	24.0778 (14.7799, 32.3728)	89.2106 (-20.6723, 203.1274)	-476.4663 (-905.3743, -56.9789)
35	4.7096 (4.2835, 5.1158)	25.6183 (17.2817, 34.3839)	50.8882 (-54.3067, 157.2487)	-362.6989 (-741.5982, 32.0465)
40	4.7154 (4.3488, 5.1256)	28.7001 (20.9743, 36.7219)	18.2970 (-87.7011, 124.5855)	-259.0733 (-639.7903, 125.0064)
45	4.7213 (4.3181, 5.0867)	28.1296 (20.3072, 35.8469)	20.6746 (-88.1985, 129.4112)	-248.0669 (-652.7289, 159.8463)

Table 9.17: Rnonlin grades for restored signal for the possible combinations of P and M_{w}

M_{w} \ P	5	10	15	35	40	45
2	0.9904	0.9951	0.9976	0.9970	0.9975	0.9975
3	0.9920	0.9956	0.9971	0.9958	0.9967	0.9966
4	0.9931	0.9961	0.9967	0.9957	0.9969	0.9968

9.3.3 Real signal and more general distortions

The same signal `flute.wav` was considered, now being distorted by the function $f(x) = \arctan(\lambda x)/\lambda$, for $\lambda = 3$. This case is more interesting, since there is no "correct" values of P nor M to compare with. Since in Section 9.1.3 the number of estimated polynomial coefficients was $M = 4$ and motivated by the choice of P in the last section, here we considered the following possible values for M and P:

$$P \in \{5 \;\; 10 \;\; 15 \;\; 35 \;\; 40 \;\; 45\} \tag{9.12}$$

$$M \in \{3 \;\; 4 \;\; 5\}. \tag{9.13}$$

The DIC value for all the evaluated combinations of P and M was between -9,650 and -9,860, not indicating a clearly better choice of orders, at least from the statistical viewpoint. From the perceptual perspective the same behavior occurs: the Rnonlin grade for the signals restored with the polynomial coefficients estimated from all possible combinations of P and M were all above 0.99, indicating that these signals are essentially indistinguishable from the original undistorted signal.

Since the obtained results here presented no small variability, we chose to not display any table or picture detailing them.

9.3.4 Conclusion

From the set of tests performed in this section, we conclude that the proposed method to deal with memoryless nonlinear distortions by estimating a polynomial approximation to its inverse is quite robust to the choice of order of both AR model and number of polynomial coefficients, even when the considered distortion does not follow the model being used. Since the same behavior was not observed when considering the piecewise linear approximation to the inverse of the nonlinearity, only the polynomial approximation was employed in the model with memory, and the respective results are discussed in more detail in the next section.

9.4 Nonlinear distortions with memory

A very extensive set of tests were performed with this method, and their results are reported in [7]. In website [72] under the link *Companion website to paper submitted to IEEE SPL* the tested signals are available for listening, and a short description of these tests and respective results is presented here.

Three signals were tested, all in in 16-bit PCM format and sampled at 44,100 Hz:

- `chopin.wav`, a 9-s long excerpt of a piano piece composed by Chopin;

- `holst.wav`, a 16-s long excerpt of an orchestral piece composed by Holst;
- `voice.wav`, a 2-s long female speech signal in Portuguese.

Although the proposed algorithm was tailored to deal with a static memoryless nonlinearity with polynomial inverse followed by an IIR filter, tests were performed in more general scenarios: other nonlinearities were considered, like $g_1(y) = \arctan(\lambda y)/\lambda$, $g_2(y) = \text{erf}(\lambda y)/\lambda$ and $g_3(y) = \tanh(\lambda y)/\lambda$; and an FIR linear filter with transfer function $B(z) = 1 - 1.21z - 0.46z^2$. Besides that, we also considered distortions following the model $g(y) = (1/\mu)[(\mu y) + 5(\mu y)^3 + 10(\mu y)^5 + 30(\mu y)^7]$ followed by an IIR filter with transfer function $B(z) = 1/(1 - 1.68z + 2.112z^2 - 1.72z^3 + 0.95z^4 - 0.38z^5 + 0.079z^6)$. The parameters μ and λ are designed to produce varied distortion levels without changing the nature of the distortion. As can be seen in the Rnonlin grades reported in [72], the restoration procedure performs well even when dealing with more general cases, despite some restored signals not sounding exactly the same as the original one. This is a point to be further investigated, since apparently the nonlinearity is mitigated but a linear residual is left in the restored signal. From the perceptual viewpoint, this can be easily circumvented by an equalization of the restored signal, performed by the user, but we wish to seek for a more elegant and less hands-on solution.

9.5 Conclusion and future works

We presented two methods for restoring signals degraded with memoryless nonlinear distortion, one where the nonlinearity is modeled as the inverse function of a polynomial and other where it is supposed to be piecewise linear. An extension of the polynomial modeling was also presented, now coupled to a linear filter modeling the distorting system memory. The option for the polynomial instead of the piecewise linear model is due to the fact that the former showed to be more stable and less dependent of parameter tuning by the user.

In the memoryless case, no difference between restored and distorted signals were noted in any of the situations described above, whereas in the problem with memory apparently some residual of the linear filter was left on the restored signal when dealing with distortions more general than the model could describe, a point to be further investigated. It is important to note that both methods proposed to treat memoryless nonlinear distortions are robust to initialization.

Publications derived from this framework are: [6], which introduced the piecewise linear modeling at EUSIPCO 2015, and [7], which presents the algorithm for restoration of nonlinear distortions with memory. We discuss now some future works.

9.5.1 Future works

A drawback of all the three proposed methods is that a great feature of the Bayesian approach is being unused: the possibility of employing a prior distribution that reflects some previous knowledge about the underlying signal. For example, when modeling audio signals via the AR model, it is expected that the poles of the respective filter are close to the unit circumference, and that lower frequencies are more likely to appear than higher ones. Since we are modeling the underlying signal via the difference equation that defines the AR model, the only way we can control this information is via the coefficients of the polynomial in the denominator of its respective filter. It is not straightforward to translate this previous knowledge about the location of the roots of a polynomial into information about its coefficients. Moreover, as argued in [76], polynomial roots can be very sensitive to their coefficients, meaning that for some polynomials a small coefficient change greatly changes their roots. In [77] and [78], a framework for estimating AR models via their poles is presented, together with the possibility of modeling uncertainties in their orders. Implementing this method, first in the memoryless case to gain intuition, and then in the case with memory both for the AR model and for the linear filter, is a future work of this thesis. Despite needing a lot of technical effort to understand and implement the algorithm, we expect that this new framework will not present convergence problems and will require no fine tuning of parameters by the user nor the reinitialization procedure[8]. Another advantage of this approach is that with more significant prior information it will be possibly easier to estimate the parameters of the distortion when the signal contains some level of noise.

As mentioned above, it is not so easy to control the shape of a polynomial via its coefficients, and this also impacts the modeling of the static memoryless nonlinearity. Recall that in Chapters 7 and 8 we needed to perform a quite complicated procedure in order to ensure that the polynomial satisfied the required constraints, since they are not easily described via its coefficients. A possibility to circumvent this issue is to approximate the nonlinearity by splines instead of single polynomials[9]. Another possibility to model the static memoryless nonlinearity is via a Gaussian process [79]. This approach was reported to present good results in the context of blind identification of Wiener systems [80], and possesses the advantage of being nonparametric, allowing for more general nonlinearities.

Recall that we expect that, when dealing with nonlinear distortions with memory, the posterior distribution for the parameters of the linear filter was expected to be multi-modal. Sampling from distributions with this characteristic is quite com-

[8]We thank very much my friend Carlos Tadeu for suggesting me these two references and this extension of the work.

[9]We thank professors Thais Fonseca and Helio Migon for this and other suggestions.

plicated since there is a possibility to being trapped around a single mode for a long time, and a reinitialization procedure was proposed to avoid this issue and accelerate convergence. However, recently a method was proposed to sample from multi-modal distributions, inspired by problems in petroleum engineering [81]! Since in our case the dimension of the search space is quite small, this method is feasible to be used without increasing too much the time per iteration. Try to use this sampling scheme is a future work.

Departing somewhat from the Bayesian approach, another possibility to treat memoryless nonlinear distortions is an adaption of [82], where the authors propose an iterative algorithm to restore lost time samples in discrete-time signals that can be locally described by an AR process. The function to be minimized is the sum of squares of the residual errors, which is a function of the AR model parameters and the missing time samples, and is also quadratic with respect to both variables separately. In our case, this quantity is a function of the AR model parameters and the polynomial coefficients of the nonlinearity, but the property of being quadratic with respect to both variables separately still holds. Therefore, the same algorithm proposed in [82] could be applied here. Preliminary tests indicate that this approach presents good results, but more tests should be performed.

In [83] the authors present a Bayesian method for estimating time-frequency surfaces for signals corrupted with noise, with applications to music and speech signals. This framework might be more adequate to restore audio signals corrupted with background noise than by modeling them via the AR model and estimating its time samples, and coupling it with the estimation of nonlinear distortion is a possible future work. A crude implementation of the algorithm presented in [83] was done and not extensively tested. The coupling with the estimation of nonlinearity will be very challenging, both theoretically and computationally.

The maximum a posteriori estimator for the parameters of the distortion and for the underlying signal is not ideal from the perceptive viewpoint. As mentioned in Chapter 7, this estimator yields the signal that is closer to the original one in the ℓ_2 norm, and it is a well known fact that our ears do not perceive similarity in this norm[10]. Therefore, the introduction of psychoacoustical elements into the restoration procedure is a good idea to circumvent this problem. This approach showed good results when applied to the problem of enhancement of signals with broadband noise [84], but is not easily generalized to nonlinear distortions. However, how humans perceive nonlinearities in audio signals was a subject of study, and a series of experiments culminated in the Rnonlin metric we used to measure the quality of the restored signals (see [47] and references therein). More recently,

[10]For example consider MP3 files, which when plotted against their uncompressed versions are very different but may sound very similar.

the use of psychoacoustical elements was introduced in the nonlinear domain: in [85] the authors proposed a method for clipping signals with low perceived distortion with relation to the underlying signal; the "dual" problem, the declipping of an audio signal, was tackled in this framework in [86]; and in [87] a psychoacoustically motivated optimization criterion was proposed to compensate nonlinearities caused by loudspeakers. In this last work, the model used for the nonlinearity is the Hammerstein model, the same employed in Chapter 8 to model nonlinear distortions with memory.

Furthermore, comparisons of the proposals with previous works in this area will also be addressed as future works.

Finally, we recall that the main goal of this research line is to deal with nonlinear distortion with memory, more adequate to model nonlinearities present in audio signals. This will be the main focus of the future research in this direction.

Part III

RESTORATION OF AUDIO SIGNALS WITH NONLINEAR DISTORTIONS VIA SPARSITY-BASED MODELS

Chapter 10

Not so Bayesian: Memoryless nonlinear distortions from a sparse viewpoint

This chapter departs somewhat from the rest of the thesis, since Bayesian methods are not used here. It contains the first part of a collaboration with Leonardo Duarte from UNICAMP, and here an adaption of his work [4] to the context of audio signals is presented. Instead of modeling the original undistorted audio signal via the AR model, we now adopt another philosophy: it is reasonable to suppose that audio signals are approximately sparse in the frequency domain, and since nonlinearities "create new frequencies", as discussed in Chapter 5, nonlinearly distorted audio signals are less sparse in the frequency domain. This fact will be exploited to create an objective function, related to the sparsity degree of the restored signal, that when minimized will yield the coefficients of a polynomial approximation of the inverse of the nonlinear distortion.

The Chapter is organized as follows: in Section 10.1 we argue that it is reasonable to suppose that audio signals are approximately sparse in the frequency domain; next, in Section 10.2, the problem of restoring nonlinearly distorted audio signals is formulated in the sparsity context, and the corresponding proposed solution is presented; results are shown in Section 10.3, followed by a discussion of future works in Section 10.4.

10.1 Are audio signals really sparse in the frequency domain?

Let $\mathbf{x}$ be an excerpt of an audio signal of length N. We can compute its DFT, but since it is a complex transform, we opt for using its DCT. The DCT is a family

of real transforms (and thus of clearer interpretation) that projects a vector onto a basis composed of time-sampled cosines. The most widely used member of this family of transforms is usually called *the* DCT in the current literature, and the respective DCT coefficients of signal $\mathbf{x}$ are given by:

$$w_k = c_k \sum_{i=1}^{N} x_i \cos\left(\frac{\pi}{2N}(2i-1)(k-1)\right), \tag{10.1}$$

for $k = 1, \ldots, N$, where the weighting coefficients c_k are given by $c_1 = 1/\sqrt{N}$ and $c_k = \sqrt{2/N}$, for $k = 2, \ldots, N$. Alternatively we can write this more compactly in matrix form:

$$\mathbf{w} = \mathbf{\Psi}\mathbf{x}, \tag{10.2}$$

for an adequate matrix $\mathbf{\Psi}$. It can be shown that the DCT matrix is orthogonal, that is, the signal in time domain can be easily recovered via

$$\mathbf{x} = \mathbf{\Psi}^T\mathbf{w}. \tag{10.3}$$

The DCT was firstly introduced in [88], and its importance in Signal Processing, in particular for lossy image and audio compression, relies on its energy compaction property [89], which means that for typical audio and images signals most of its information tends to be concentrated in a few low-frequency components. More precisely, when the DCT of a signal is computed, one expects that there will be only a few coefficients substantially different from zero and that the large majority of them will be very close to zero. Signals with this property are called *compressible* in the DCT domain, since the smallest components can in principle be ignored with little impact on the overall structure of the signal. Mathematically it is easier to formulate the concept of *sparsity*, meaning that only a few coefficients are different from zero, and that all the others are exactly equal to zero.

The notion of sparsity is clearly just an approximation of the more realistic concept of compressibility, and the latter may be reasonably assumed in most audio signals, or at least in excerpts of them: one does not expect that a short excerpt of an audio signal (of approximately 20 ms, for example) contains a lot of frequencies with substantially large coefficients, unless the considered excerpt is very noisy or contains the attack of a note. With this in mind, we will assume that small excerpts of undistorted audio signals are in general compressible in the DCT domain.

10.2 Formulation of the problem and proposed solution

As before, denote by $f(.)$ the distorting function, which we will suppose to be invertible and anti-symmetric, and denote its inverse by $g(.)$. We approximate it parametrically via a polynomial given by

$$g(y) = m_1 y + m_2 y^3 + \cdots + m_M y^{2M-1}. \tag{10.4}$$

Note that here we are not imposing that $f'(0) = 1$ as previously, and the reason will become clear soon. Denote the vector containing the polynomial coefficients by $\mathbf{m}$, the original undistorted signal by $\mathbf{x}$ and the observed distorted signal by $\mathbf{y}$. The relation between $\mathbf{x}$ and $\mathbf{y}$ can be written in matrix form as

$$\mathbf{x} = \mathbf{Y}_{\mathrm{s}}\mathbf{m}, \tag{10.5}$$

where matrix $\mathbf{Y}_s$ is (unlike in Chapter 7) given by

$$\mathbf{Y}_{\mathrm{s}} = \begin{bmatrix} y_1 & y_1^3 & \cdots & y_1^{2M-1} \\ y_2 & y_2^3 & \cdots & y_2^{2M-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_N & y_N^3 & \cdots & y_N^{2M-1} \end{bmatrix}, \tag{10.6}$$

being the subscript "s" used to denote the sparsity context.

Therefore, having in mind that $\mathbf{x}$ is more sparse than $\mathbf{y}$ in the DCT domain, up to this point the problem can be formulated as finding coefficients $m_1, \ldots, m_M \geq 0$ of polynomial $g(.)$ such that the restored signal $\mathbf{Y}_{\mathrm{s}}\mathbf{m}$ is maximally sparse in the DCT domain. Note that another condition must be placed over the restored signal in order to solve the problem, otherwise $\mathbf{m} = \mathbf{0}$ will always be the best solution that fits into the sparsity criteria! In previous works [6, 7, 67] the condition $f'(0) = 1$ was imposed, but preliminary tests indicated that in this context this restriction leads to a very "noisy" objective function, which is difficult to optimize. Now we require that $\|\mathbf{Y}_{\mathrm{s}}\mathbf{m}\|_2 = 1$, where $\|\cdot\|_2$ denotes the ℓ_2 norm of a given vector, to avoid the null solution.

Despite nonlinearities creating new frequencies within the original signal, it is not so obvious that this simple sparsity criterion will lead to a reasonable result. Nevertheless, it is proved in [4] that under some mild conditions over original signal $\mathbf{x}$ and distorting function $f(.)$ this criterion is sufficient to guarantee perfect recovery of the original signal.

10.2.1 How do we measure sparsity?

The most obvious way to measure the sparsity of a signal is via its ℓ_0 "norm"[1], which is simply the number of its nonzero entries. One can then try to find $\mathbf{m}$ that minimizes the ℓ_0 norm of vector $\mathbf{\Psi Y}_s\mathbf{m}$, that is, the number of nonzero coefficients in the DCT of the restored signal. However, it is well known that optimization problems that deal directly with the ℓ_0 norm are NP-hard [91], and thus impossible to solve in a feasible time. Furthermore, because the original audio signal is not exactly sparse in the DCT domain, the use of the ℓ_0 norm would yield meaningless results. There are other functions that behave like a quantification of sparsity in some contexts, being the ℓ_1 norm the most common [90]. Another possibility is to use an ℓ_p norm with p close to zero [90]. Here we consider another criteria, a smoothed version of the ℓ_0 norm [92], computed as

$$F_\sigma(\mathbf{v}) = N - \sum_{i=1}^{N} \exp\left(-\frac{v_i^2}{2\sigma^2}\right), \tag{10.7}$$

where N is the dimension of vector $\mathbf{v}$, v_i are its entries for $i = 1, \ldots, N$ and σ is a parameter that controls the smoothness of function $F_\sigma(.)$. It is easy to verify that

$$\lim_{\sigma \to 0} F_\sigma(\mathbf{v}) = \|\mathbf{v}\|_0. \tag{10.8}$$

Therefore, the smaller is σ, the best the approximation is, but function $F_\sigma(.)$ is less smooth; on the other hand, a larger value of σ implies a worse approximation but a smoother function.

The problem to be solved can be summarized in the following way:

$$\begin{aligned} &\min_{\mathbf{m}} F_\sigma(\mathbf{\Psi Y}_s\mathbf{m}) \\ &\text{subject to } m_i \geq 0, \text{ for } i = 1, \ldots, M \\ &\text{and } \|\mathbf{Y}_s\mathbf{m}\|_2 = \mathbf{m}^T\mathbf{Y}_s^T\mathbf{Y}\mathbf{m} = 1. \end{aligned} \tag{10.9}$$

The proposed optimization problem is a nonlinear and nonconvex one, with equality and inequality constraints. Since both objective function and restrictions are continuously differentiable in their whole domain, the Karush-Kuhn-Tucker (KKT) Theorem [18] is applicable, ensuring that any critical point of the objective function in the constrained domain is also a critical point of the unconstrained

[1] In fact this is not a norm, since it does not satisfy the homogeneity condition. However, this name is currently used in the literature [90].

function $L(\mathbf{m}, \nu, \boldsymbol{\mu})$, given by

$$L(\mathbf{m}, \nu, \boldsymbol{\mu}) = F_\sigma(\boldsymbol{\Psi}\mathbf{Y}_s\mathbf{m}) - \nu\mathbf{m}^T\mathbf{Y}_s^T\mathbf{Y}\mathbf{m} - \sum_{i=1}^{M} \mu_i m_i. \tag{10.10}$$

Function $L(\mathbf{m}, \nu, \boldsymbol{\mu})$ is called the *Lagrangian* and constants ν and $\boldsymbol{\mu}$ are the KKT multipliers. The optimization algorithm employed is explained in more details in the next section.

10.3 Results

In this section, we present some results obtained with this method. Tests were performed with the same three real signals as before: `flute.wav`, `classical.wav` and `voice.wav`, distorted by functions following the model, that is, inverse of polynomial functions, and by more general ones, like the arctangent.

In the polynomial case, because of the unitary energy restriction imposed on the restored signal, the estimated coefficients will almost surely differ from the original ones by a constant factor. In order to better compare the estimated coefficients with the original ones we do not consider directly the output of the algorithm, but its normalization with respect to the respective first coefficient.

An excerpt of 3,000 time samples around the time sample with greatest absolute value was given as input to the algorithm, corresponding to approximately 70 ms at the usual frequency sampling rate of 44,100 Hz. This choice of block length was motivated by the fact that preliminary tests using 1,000 time samples (the same block size considered in Chapter 9 when the signal is modeled via the AR model) have not shown good results. By increasing the block size, the obtained result was better and beyond 3,000 time samples small variability in the results was obtained.

The initialization was random, by uniformly choosing $\mathbf{m}$ that satisfied the unit energy restriction of the restored signal. Since as the value of σ decreases the objective function becomes less smooth, the risk of being trapped at a local minimum increases, so the algorithm was run 10 times for each combination of σ and number of coefficients to estimate, and the value that produced the smallest value of the objective function was selected.

In order to solve the problem in Equations 10.9, the function `fmincon` implemented in the Optimization Toolbox™ of MATLAB™ was employed. It uses, by default, the well known interior-point algorithm [18] to find the critical points of the Lagrangian in Equation 10.10, being tailored to solve the following optimization

problem:

$$\min_{\mathbf{x}} G(\mathbf{x}) \text{ such that } \begin{cases} c(\mathbf{x}) \leq 0, \\ c_{eq}(\mathbf{x}) = 0, \\ \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}, \\ \mathbf{b}_l \leq \mathbf{x} \leq \mathbf{b}_u, \end{cases} \tag{10.11}$$

where functions $c(.)$ and $c_{eq}(.)$ can be nonlinear and the "eq" subscript denotes equality constraints.

10.3.1 Real signals and distortion following the model

The three signals were distorted by the inverse of the function $g(y) = y + 5y^3 + 30y^5$, so the set of coefficients we wish to recover (after the aforementioned normalization) is $\mathbf{m} = [1 \;\; 5 \;\; 30]^T$. Tests were performed by estimating 2, 3, 4, or 5 coefficients, with σ equal to 0.01, 0.001, and 0.0001. For the signal `flute.wav` all combinations of the parameters above are shown, together with the Rnonlin grade of the respective restored signal. Since $\sigma = 0.01$ did not produce good results, for the two other signals we consider only σ equal to 0.001 and 0.0001, and since the estimation of 5 coefficients with these values of σ always produced good results, we also omit this case for the last two signals, for the sake of conciseness. The results of each experiment is described in the label of the respective figure in the next pages.

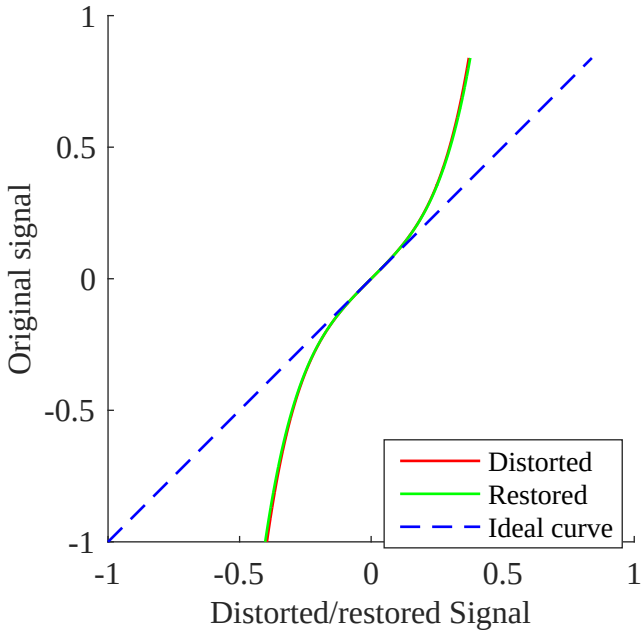

Figure 10.1: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.01$. Estimated coefficients (after normalization): $\mathbf{m} = [1\ 0.10394]^T$. Rnonlin grade of restored signal: 0.87895.

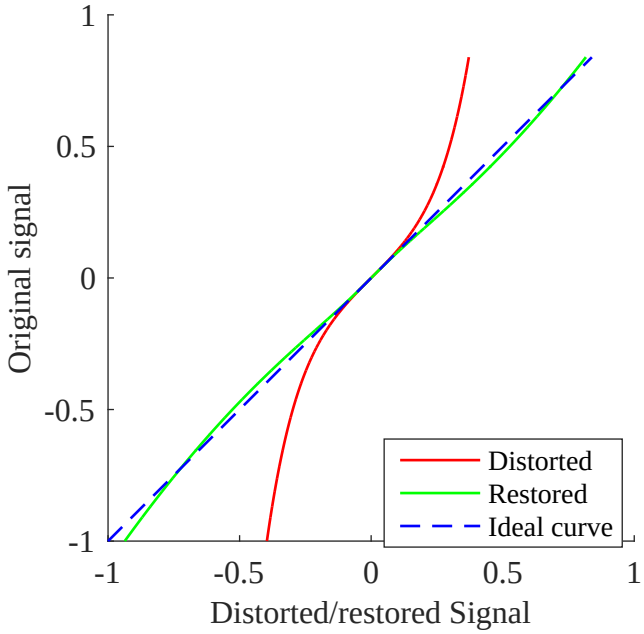

Figure 10.2: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1\ 8.6893]^T$. Rnonlin grade of restored signal: 0.95226.

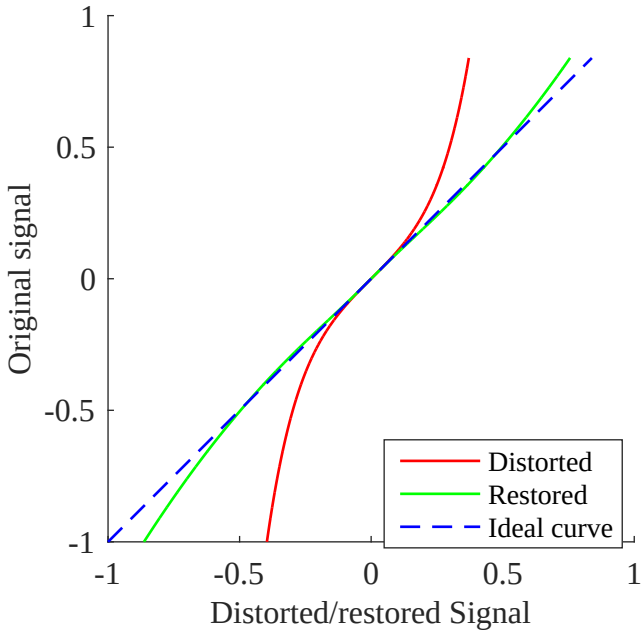

Figure 10.3: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 7.5153]^T$. Rnonlin grade of restored signal: 0.96208.

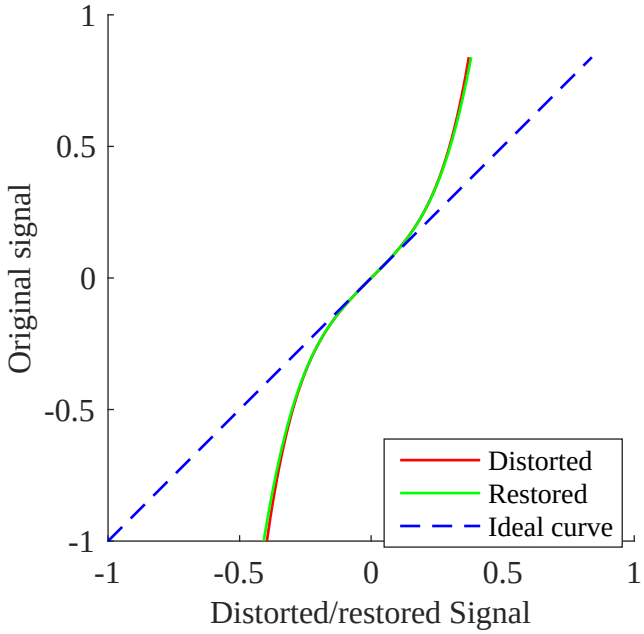

Figure 10.4: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.01$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 4.2288\times10^{-5} \quad 1.2489]^T$. Rnonlin grade of restored signal: 0.87893.

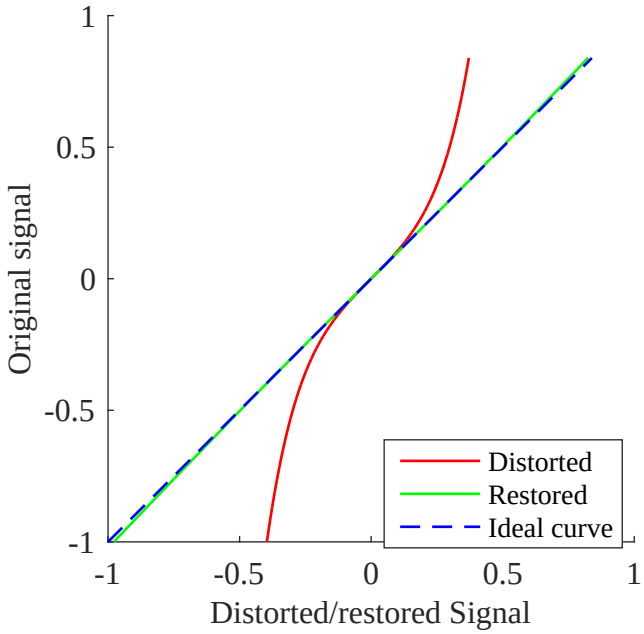

Figure 10.5: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.20316 \quad 26.4716]^T$. Rnonlin grade of restored signal: 0.99924.

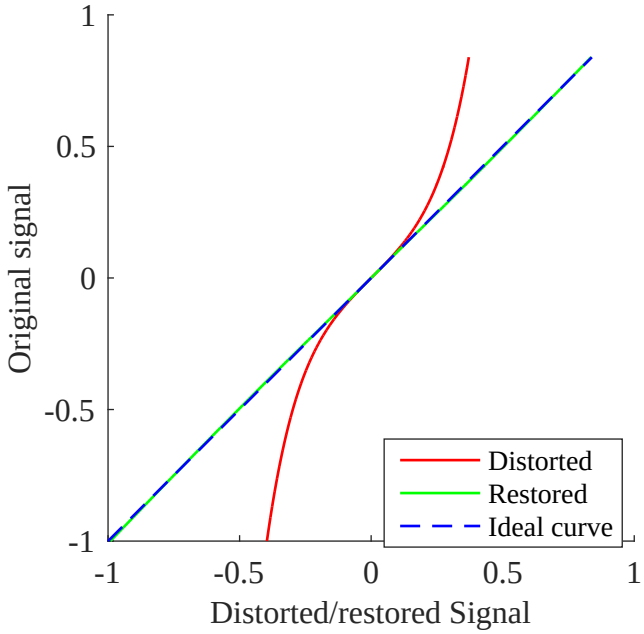

Figure 10.6: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.5478 \quad 25.6305]^T$. Rnonlin grade of restored signal: 0.9982.

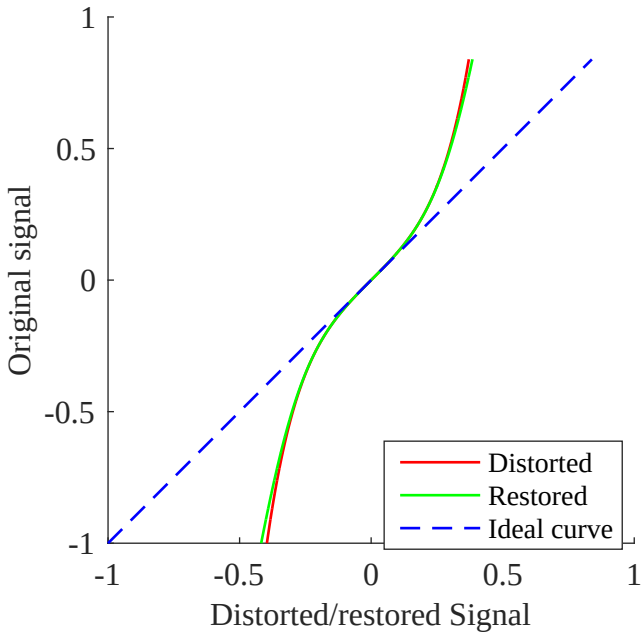

Figure 10.7: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.01$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 8.61824\times10^{-7} \quad 7.37578\times10^{-6} \quad 14.2044]^T$. Rnonlin grade of restored signal: 0.87937.

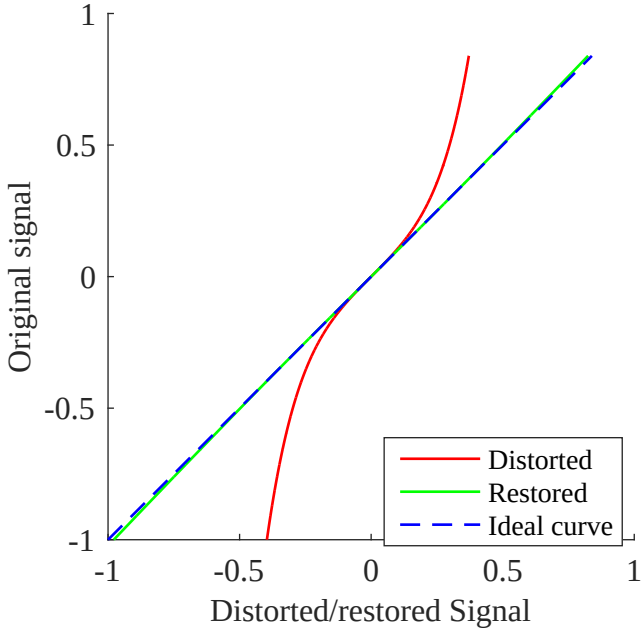

Figure 10.8: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.2033 \quad 26.4707 \quad 0.000159133]^T$. Rnonlin grade of restored signal: 0.99924.

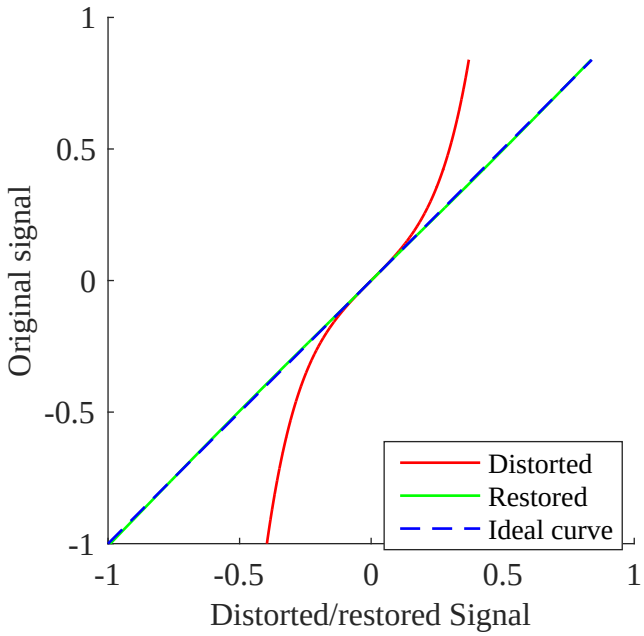

Figure 10.9: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.54781 \quad 25.6305 \quad 1.09001\times10^{-5}]^T$. Rnonlin grade of restored signal: 0.9982.

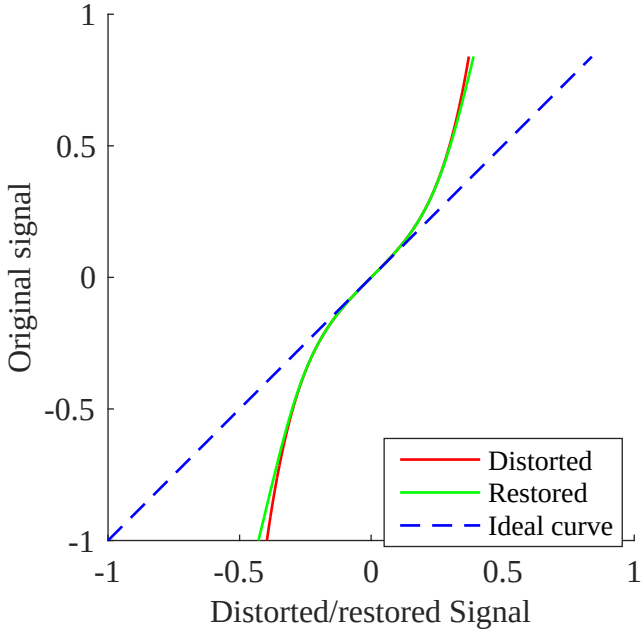

Figure 10.10: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.01$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 2.473372\times10^{-6} \quad 1.658071\times10^{-5} \quad 0.0001989655 \quad 133.1624]^T$. Rnonlin grade of restored signal: 0.87985.

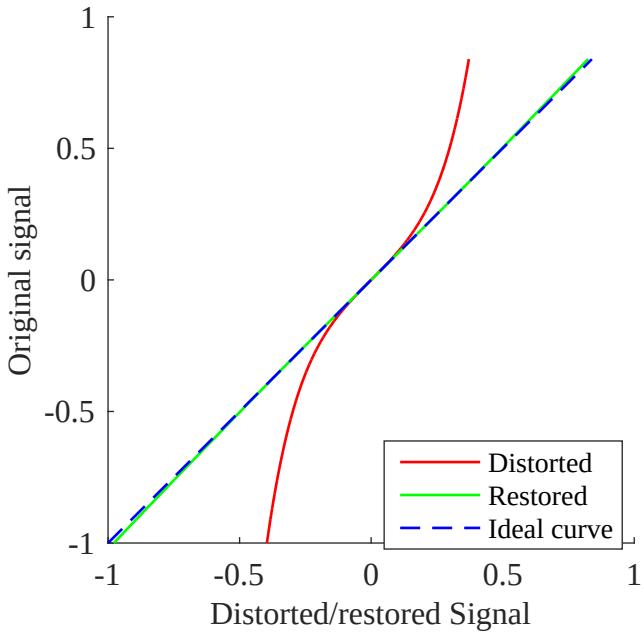

Figure 10.11: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.20311 \quad 26.4721 \quad 0.000339132 \quad 0.00132257]^T$. Rnonlin grade of restored signal: 0.99924.

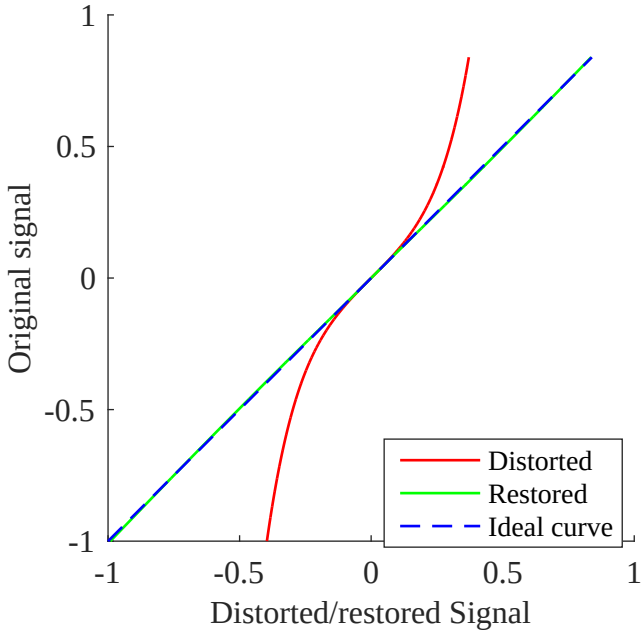

Figure 10.12: Signal `flute.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.5478 \quad 25.6305 \quad 5.57641\times10^{-7} \quad 1.61131\times10^{-6}]^T$. Rnonlin grade of restored signal: 0.9982.

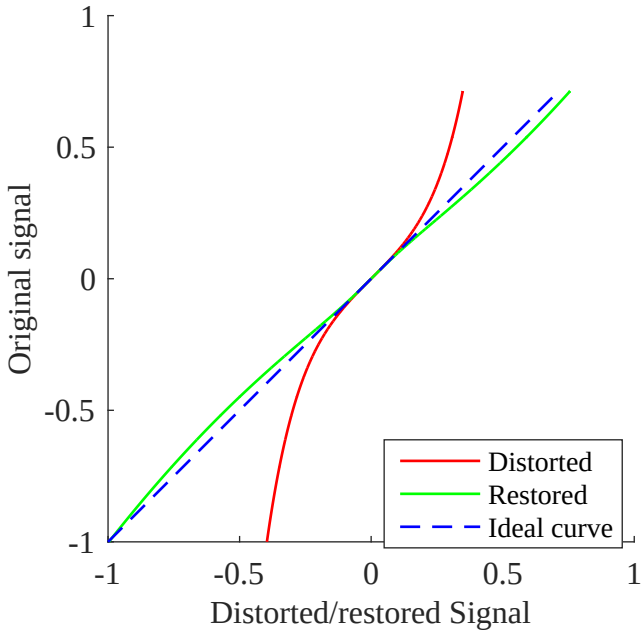

Figure 10.13: Signal `classical.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 9.6572]^T$. Rnonlin grade of restored signal: 0.99682.

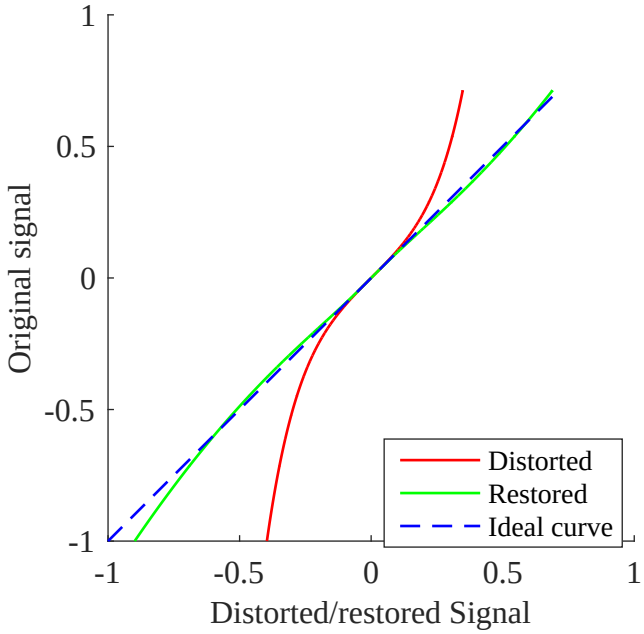

Figure 10.14: Signal `classical.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 8.0819]^T$. Rnonlin grade of restored signal: 0.99851.

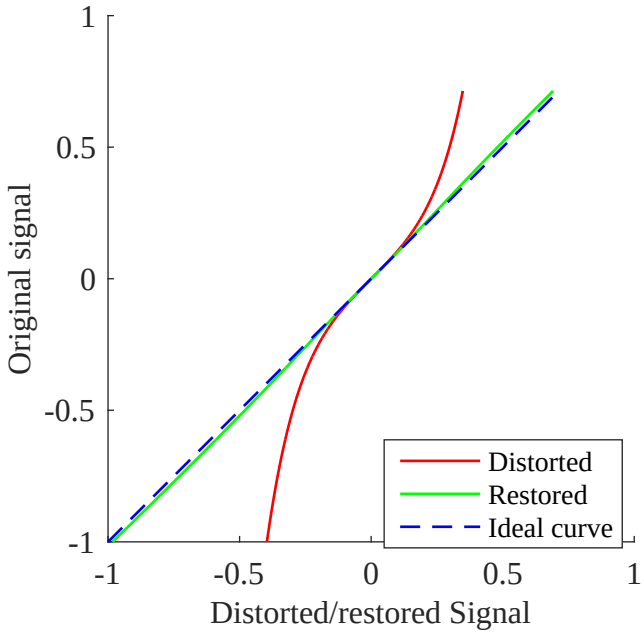

Figure 10.15: Signal `classical.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 3.61329 \quad 37.0926]^T$. Rnonlin grade of restored signal: 0.99957.

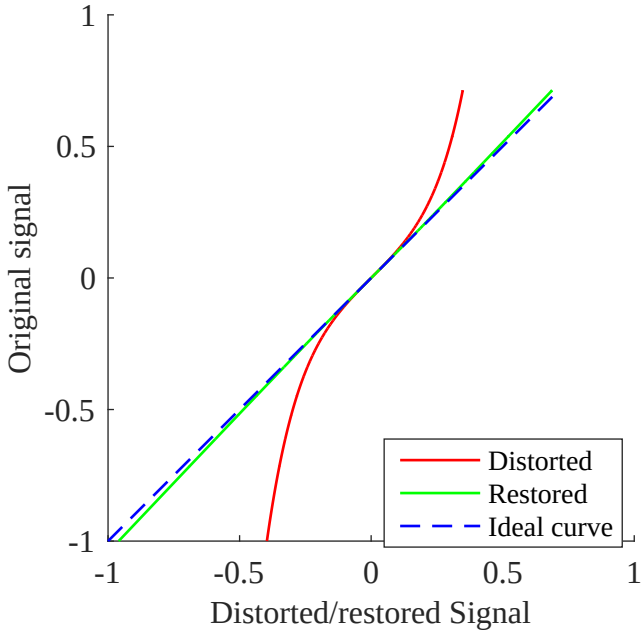

Figure 10.16: Signal `classical.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 4.56602 \quad 28.6338]^T$. Rnonlin grade of restored signal: 0.99986.

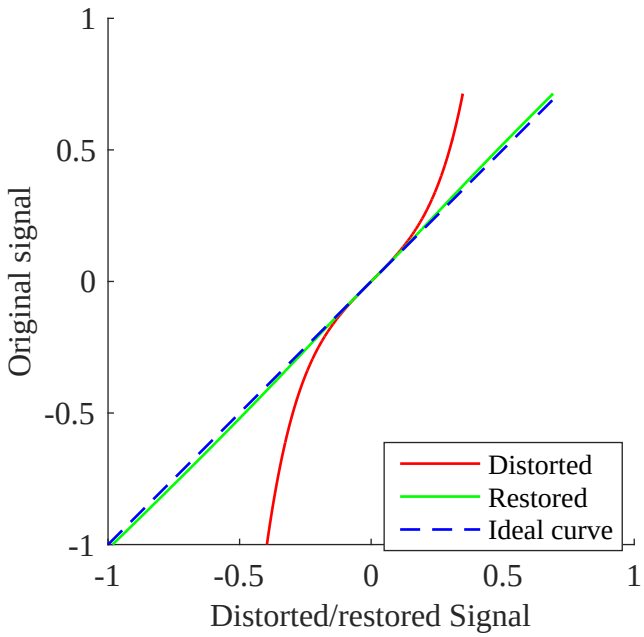

Figure 10.17: Signal `classical.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 3.61332 \quad 37.0924 \quad 0.000258148]^T$. Rnonlin grade of restored signal: 0.99957.

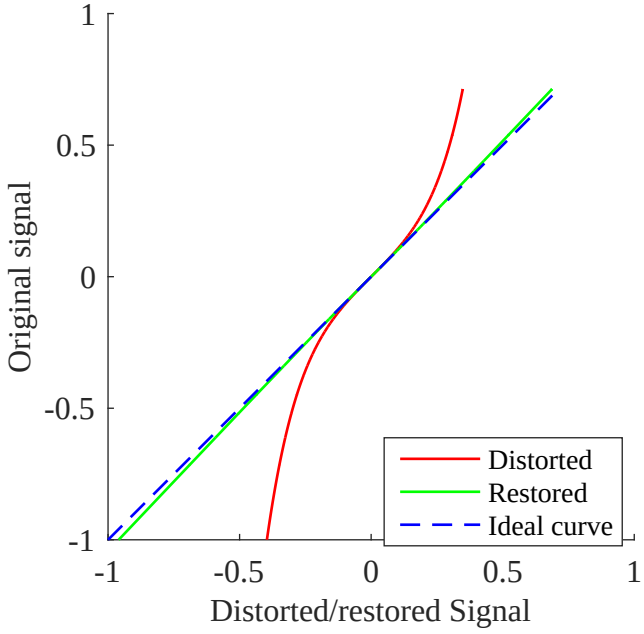

Figure 10.18: Signal `classical.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 4.56602 \quad 28.6339 \quad 3.99215\times10^{-6}]^T$. Rnonlin grade of restored signal: 0.99986.

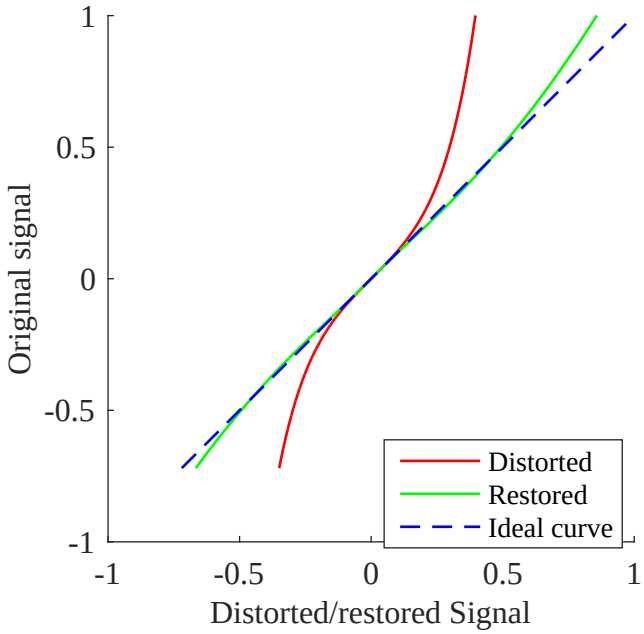

Figure 10.19: Signal `voice.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1\ 7.4192]^T$. Rnonlin grade of restored signal: 0.98252.

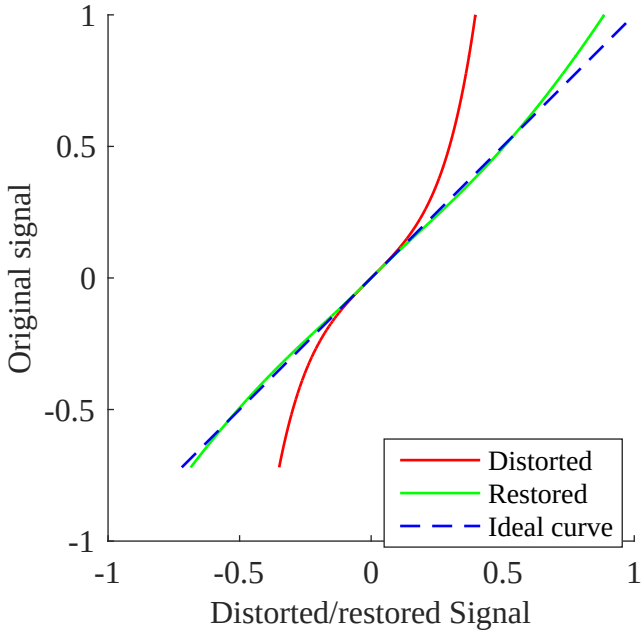

Figure 10.20: Signal `voice.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 7.8708]^T$. Rnonlin grade of restored signal: 0.9828.

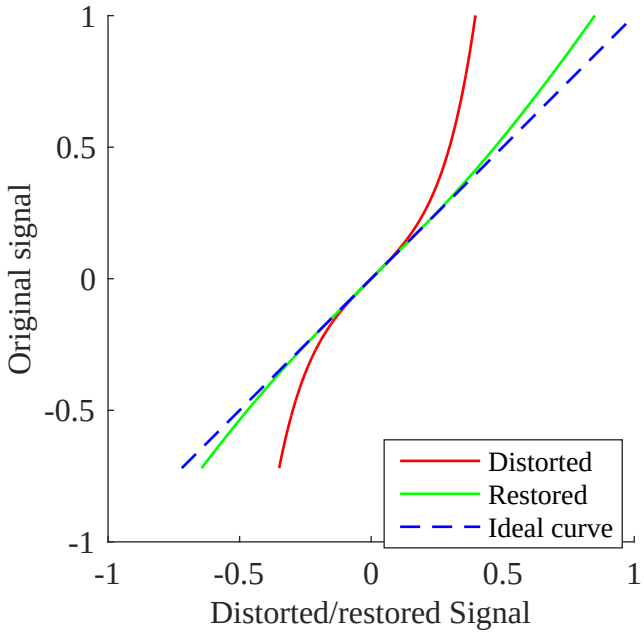

Figure 10.21: Signal `voice.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.52482 \quad 11.2933]^T$. Rnonlin grade of restored signal: 0.98717.

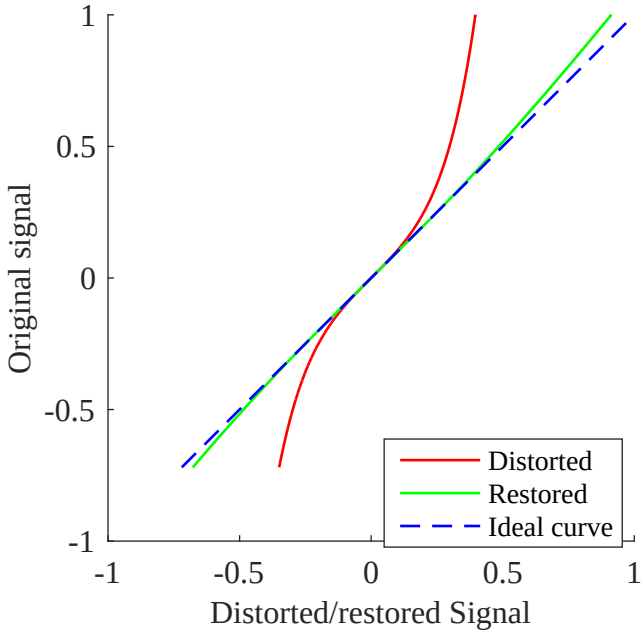

Figure 10.22: Signal `voice.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.51225 \quad 17.795]^T$. Rnonlin grade of restored signal: 0.99321.

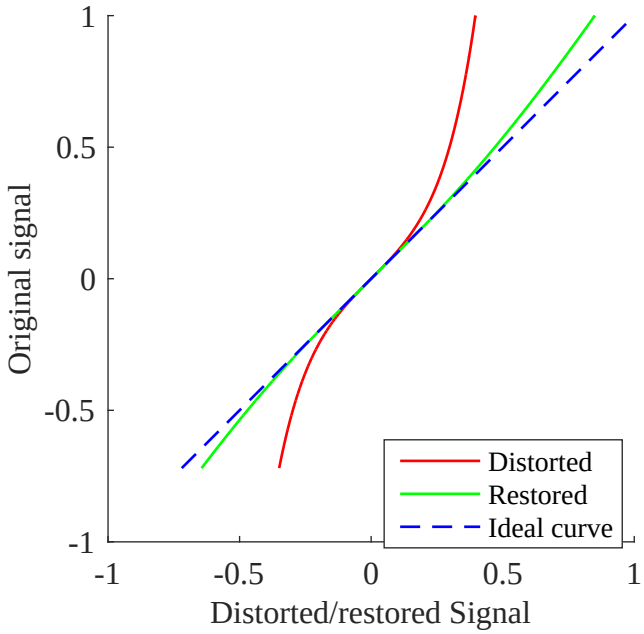

Figure 10.23: Signal `voice.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.52468 \quad 11.2941 \quad 0.000318054]^T$. Rnonlin grade of restored signal: 0.98717.

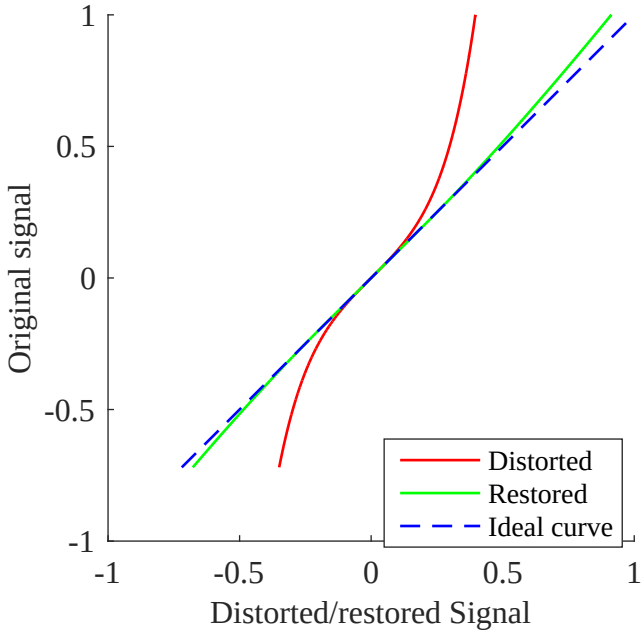

Figure 10.24: Signal `voice.wav` and distortion with polynomial inverse: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.51225 \quad 17.795 \quad 7.32559 \times 10^{-6}]^T$. Rnonlin grade of restored signal: 0.99321.

10.3.2 Real signals and more general distortions

Now we present results for the same three signals, but distorted by $f(x) = \arctan(\lambda x)/\lambda$; 2, 3, 4, or 5 polynomial coefficients were estimated when σ equal to 0.001 and 0.0001, and λ equal to 3 and 5 for signal `flute.wav`. For the two other signals, only 3, 4 and 5 coefficients (successful cases) were estimated for both values of σ, and λ only equal to 5 (more difficult case), for the sake of conciseness.

From the set of results, we conclude that the method is also capable of dealing with more general distortions that do not follow the polynomial model. But the Rnonlin grades of restored signals were inferior to those attained via the Bayesian method + AR model in Chapter 9, specially for more severe distortions.

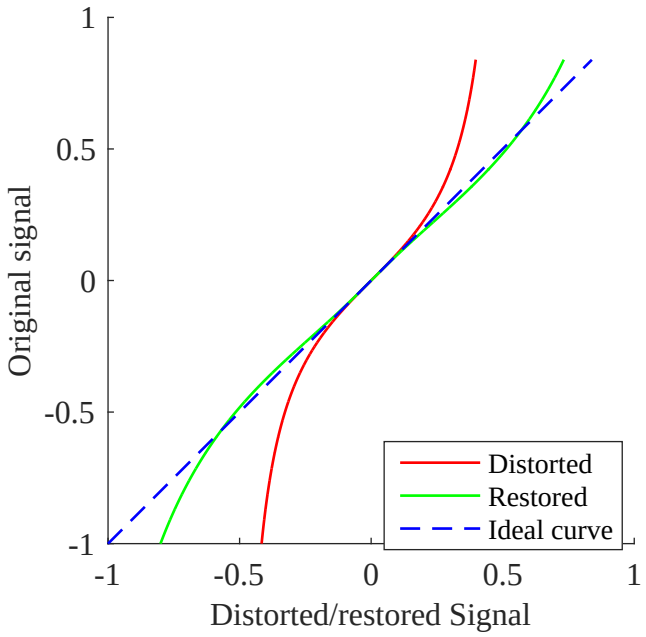

Figure 10.25: Signal `flute.wav` and arctan distortion with $\lambda = 3$: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1\ 5.3253]^T$. Rnonlin grade of restored signal: 0.93325.

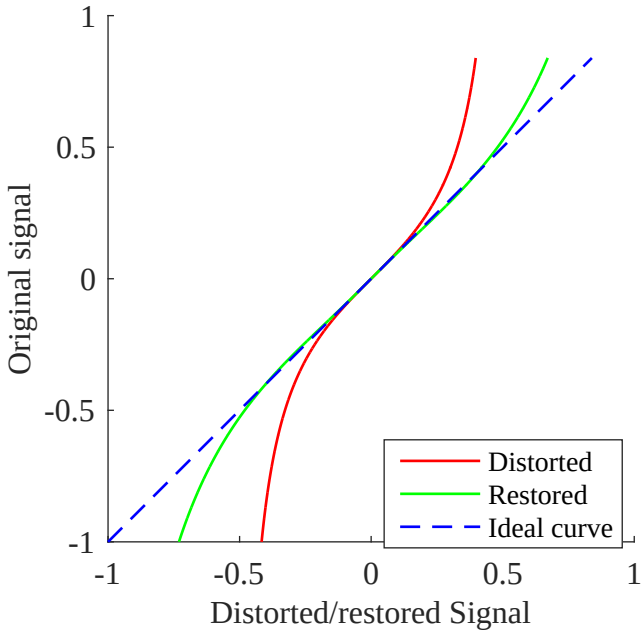

Figure 10.26: Signal `flute.wav` and arctan distortion with $\lambda = 3$: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 4.3529]^T$. Rnonlin grade of restored signal: 0.94191.

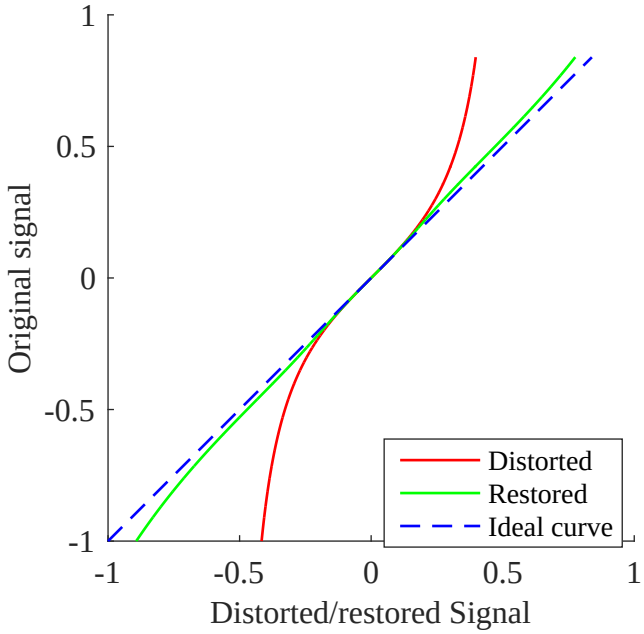

Figure 10.27: Signal `flute.wav` and arctan distortion with $\lambda = 3$: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 0.00693206 \quad 38.046]^T$. Rnonlin grade of restored signal: 0.9554.

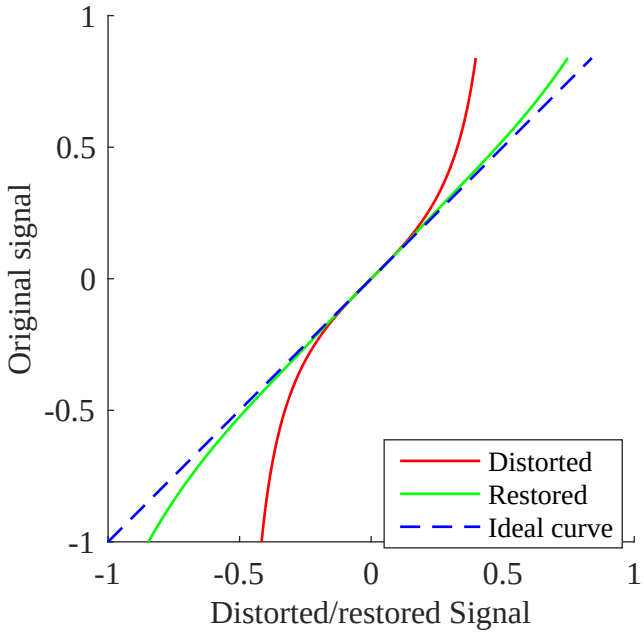

Figure 10.28: Signal `flute.wav` and arctan distortion with $\lambda = 3$: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 1.52905 \quad 25.5723]^T$. Rnonlin grade of restored signal: 0.97866.

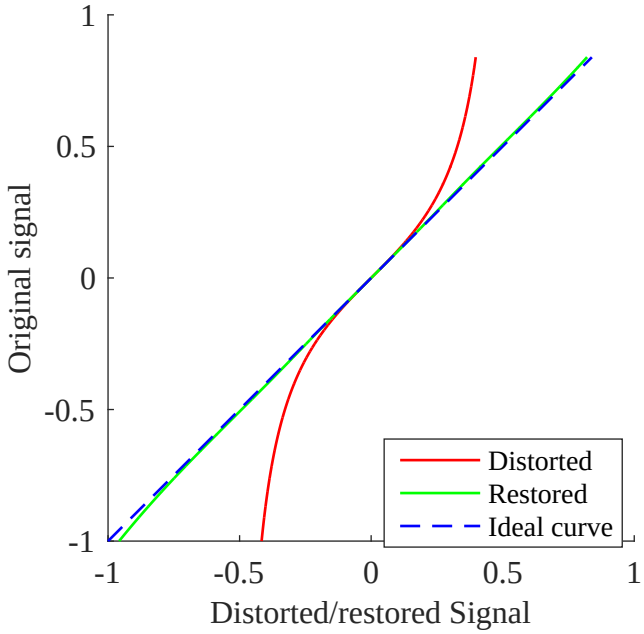

Figure 10.29: Signal `flute.wav` and arctan distortion with $\lambda = 3$: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 2.93198 \quad 1.590276 \times 10^{-5} \quad 152.1739]^T$. Rnonlin grade of restored signal: 0.99265.

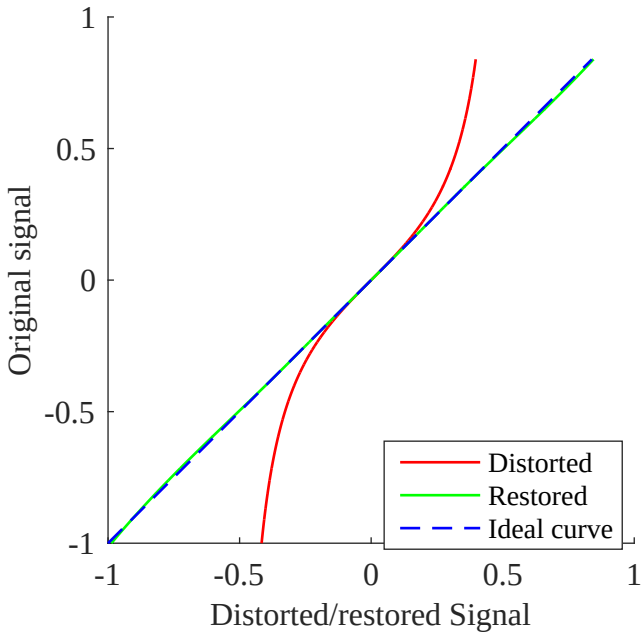

Figure 10.30: Signal `flute.wav` and arctan distortion with $\lambda = 3$: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 3.228501 \quad 1.156322\times10^{-6} \quad 155.9281]^T$. Rnonlin grade of restored signal: 0.99428.

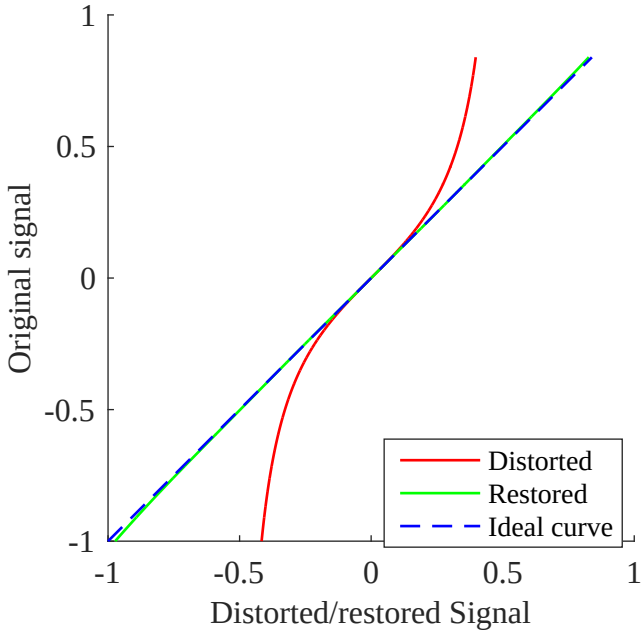

Figure 10.31: Signal `flute.wav` and arctan distortion with $\lambda = 3$: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 3.243514 \quad 7.969379 \quad 0.0004540494 \quad 592.43]^T$. Rnonlin grade of restored signal: 0.99873.

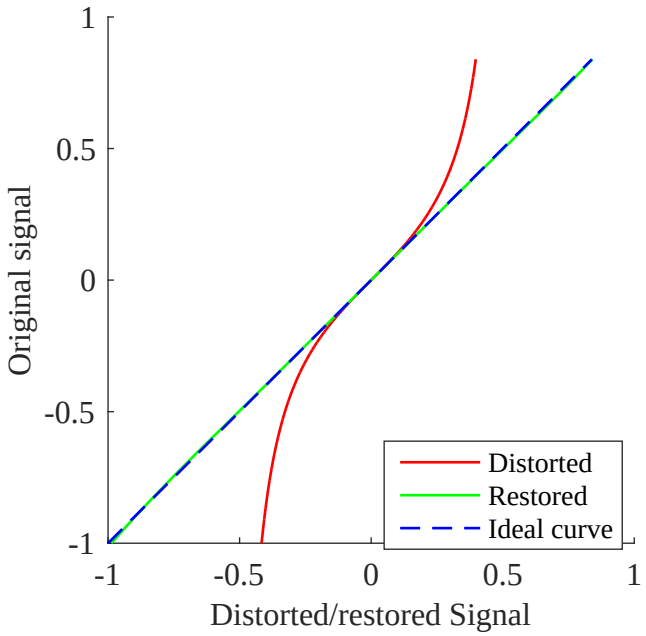

Figure 10.32: Signal `flute.wav` and arctan distortion with $\lambda = 3$: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 3.09376 \quad 10.88234 \quad 9.293036 \quad 509.0448]^T$. Rnonlin grade of restored signal: 0.99834.

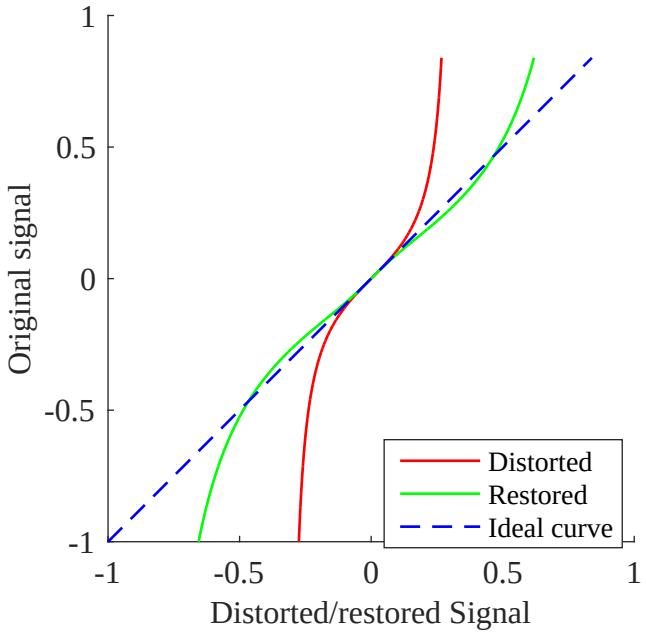

Figure 10.33: Signal `flute.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 18.388]^T$. Rnonlin grade of restored signal: 0.88007.

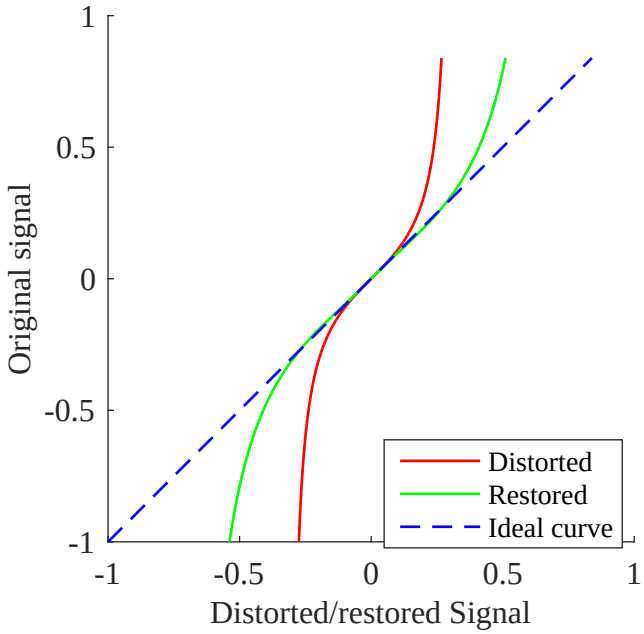

Figure 10.34: Signal `flute.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 2 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 12.7266]^T$. Rnonlin grade of restored signal: 0.90389.

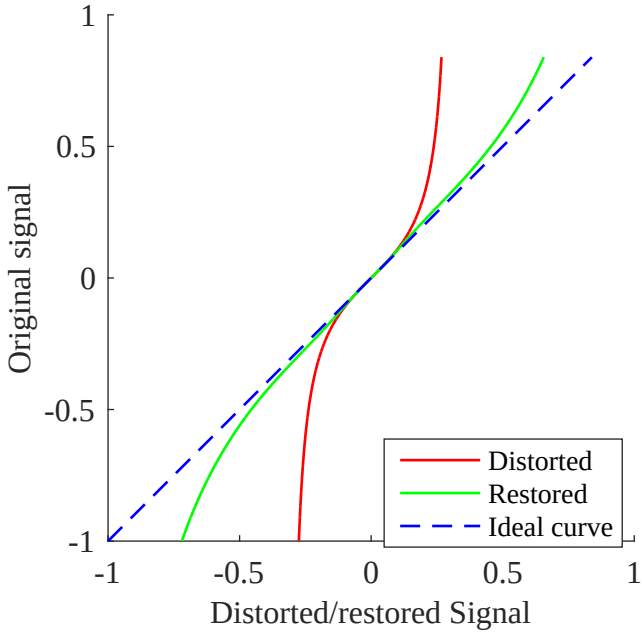

Figure 10.35: Signal `flute.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 3.496801 \times 10^{-7} \quad 284.5015]^T$. Rnonlin grade of restored signal: 0.93519.

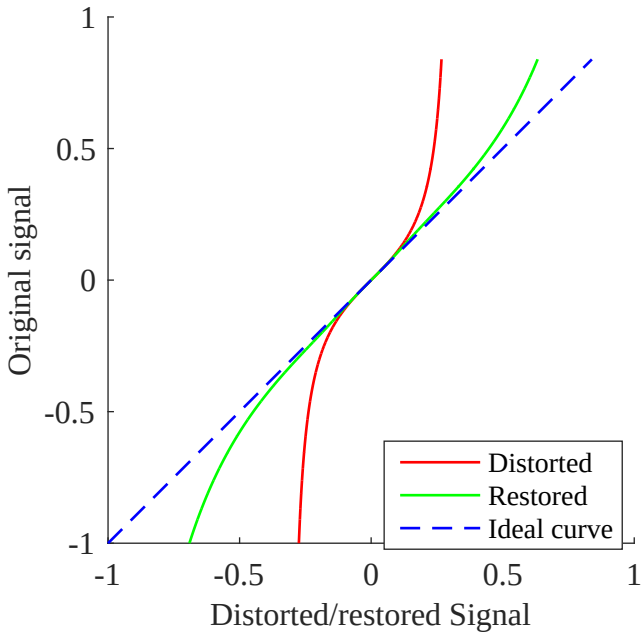

Figure 10.36: Signal `flute.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 1.895975 \quad 240.9501]^T$. Rnonlin grade of restored signal: 0.94183.

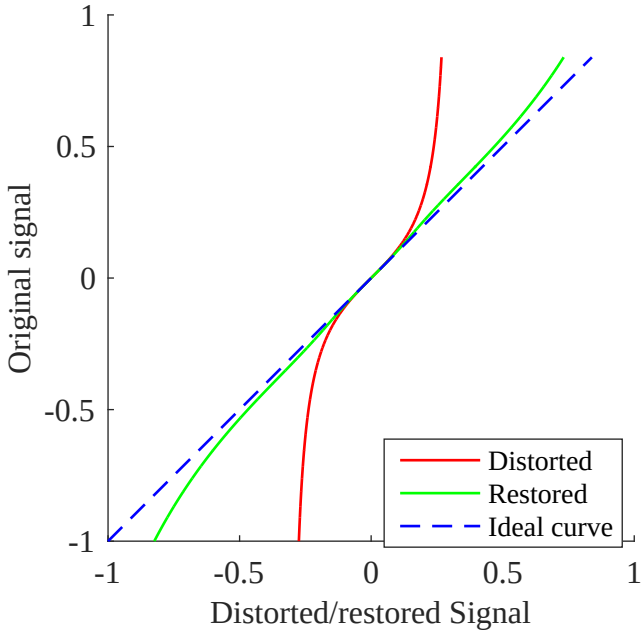

Figure 10.37: Signal `flute.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 4.8439853 \quad 7.3989657\times10^{-5} \quad 3807.2556]^T$. Rnonlin grade of restored signal: 0.94378.

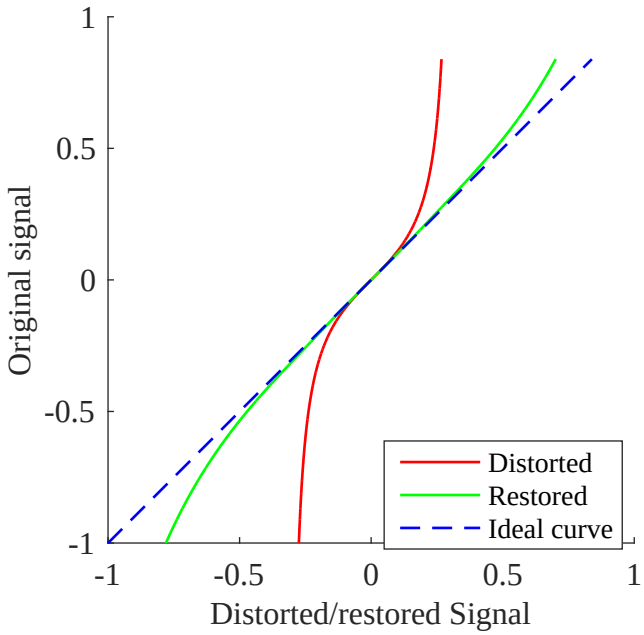

Figure 10.38: Signal `flute.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 8.1256837 \quad 1.2031784\times10^{-5} \quad 2855.4944]^T$. Rnonlin grade of restored signal: 0.96974.

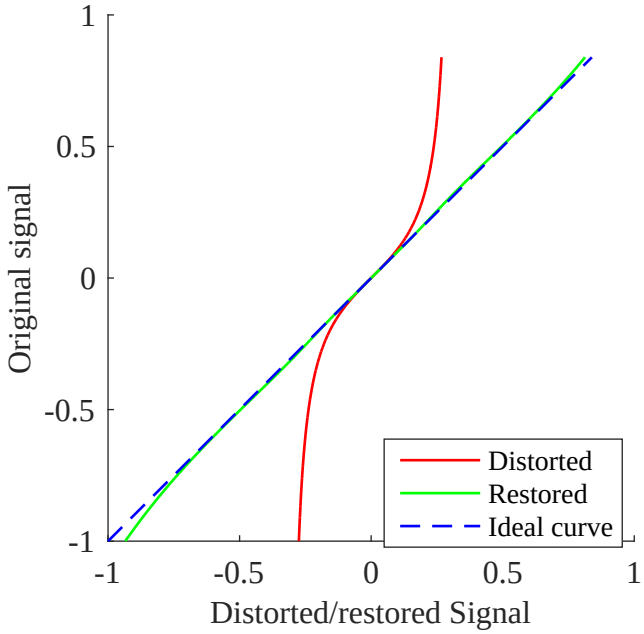

Figure 10.39: Signal `flute.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 9.7696998 \quad 3.68211803\times10^{-5} \quad 0.00056596077 \quad 51400.2147]^T$. Rnonlin grade of restored signal: 0.97501.

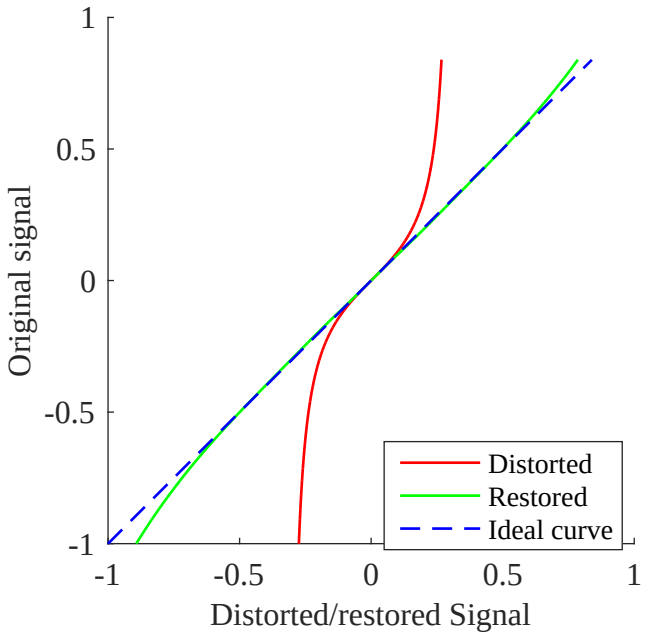

Figure 10.40: Signal `flute.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 11.6885289 \quad 1.01438552\times10^{-5} \quad 8.40545167\times10^{-5} \quad 42223.3473]^T$. Rnonlin grade of restored signal: 0.9846.

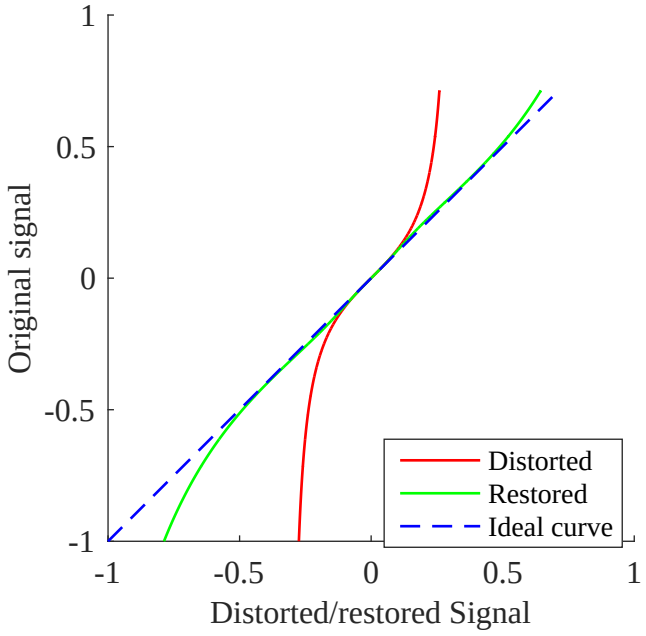

Figure 10.41: Signal `classical.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 4.630399\times10^{-8} \quad 327.907]^T$. Rnonlin grade of restored signal: 0.99621.

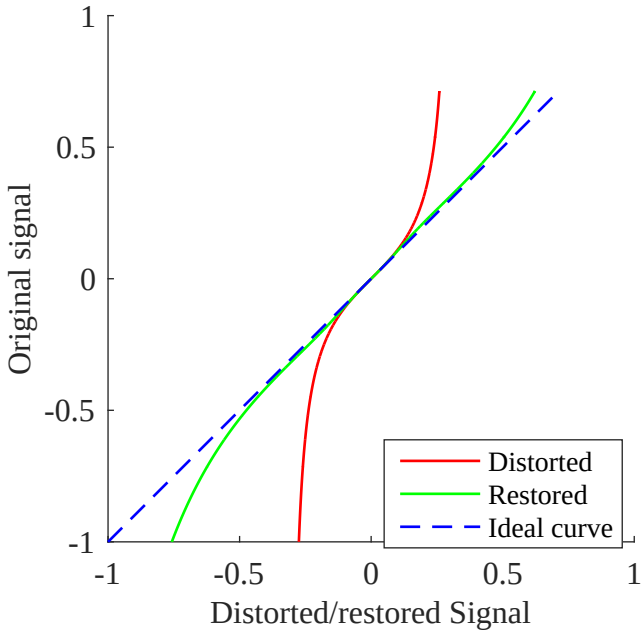

Figure 10.42: Signal `classical.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 4.809496 \times 10^{-8} \quad 308.858]^T$. Rnonlin grade of restored signal: 0.99666.

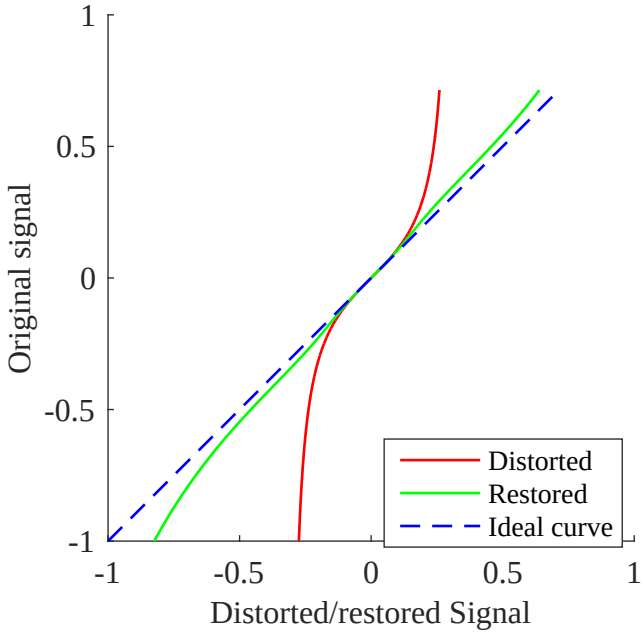

Figure 10.43: Signal `classical.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1$ 2.8919144 0.00023571741 4152.9576$]^T$. Rnonlin grade of restored signal: 0.99433.

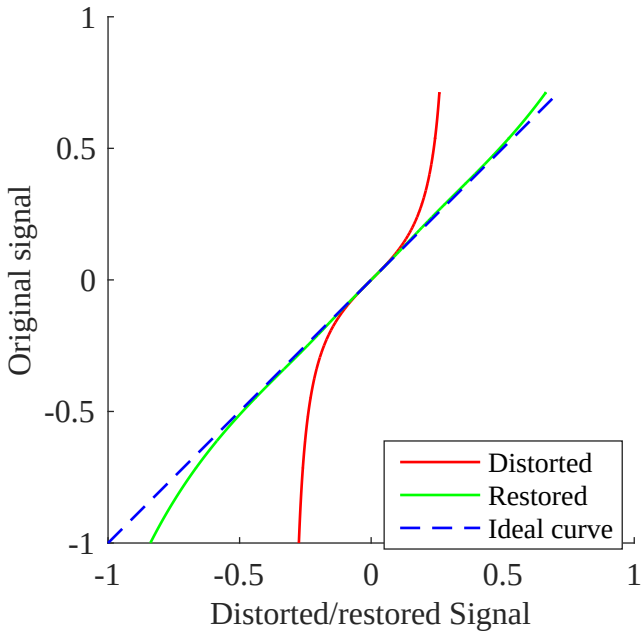

Figure 10.44: Signal `classical.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 7.356875 \quad 3.6340337 \times 10^{-6} \quad 3492.8213]^T$. Rnonlin grade of restored signal: 0.99931.

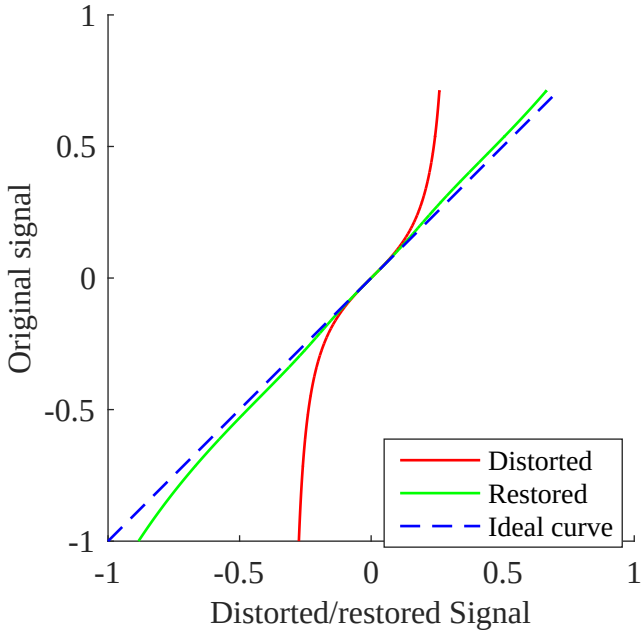

Figure 10.45: Signal `classical.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1$ $5.22644732 \quad 67.9592461 \quad 0.0011397553 \quad 44369.3479]^T$. Rnonlin grade of restored signal: 0.99767.

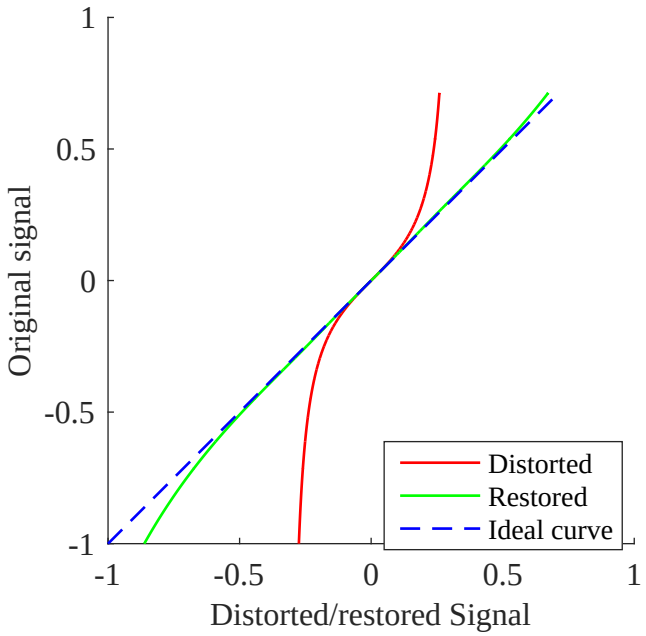

Figure 10.46: Signal `classical.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 7.41929168 \quad 86.2401877 \quad 400.492152 \quad 28287.5355]^T$. Rnonlin grade of restored signal: 0.99978.

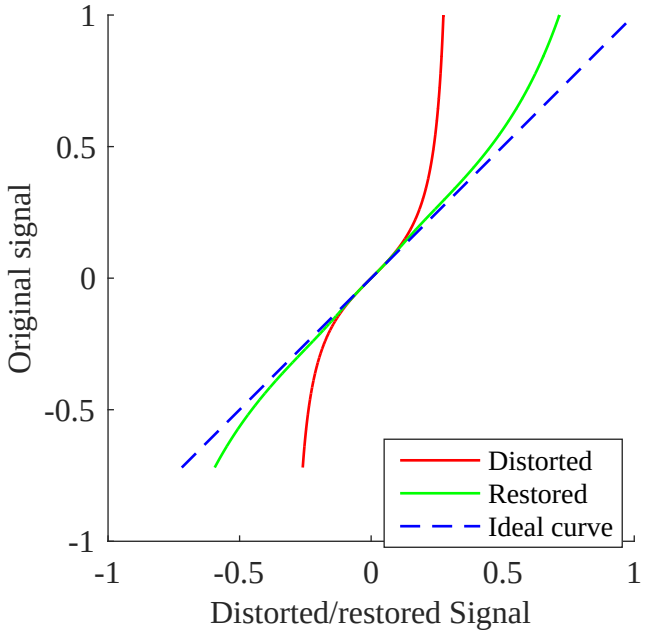

Figure 10.47: Signal `voice.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 1.534064\times10^{-7} \quad 281.9483]^T$. Rnonlin grade of restored signal: 0.97526.

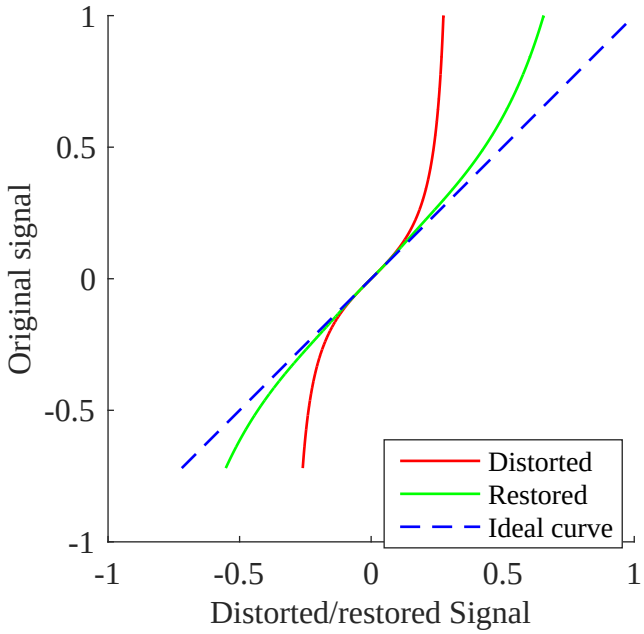

Figure 10.48: Signal `voice.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 1.893394 \quad 218.4783]^T$. Rnonlin grade of restored signal: 0.97178.

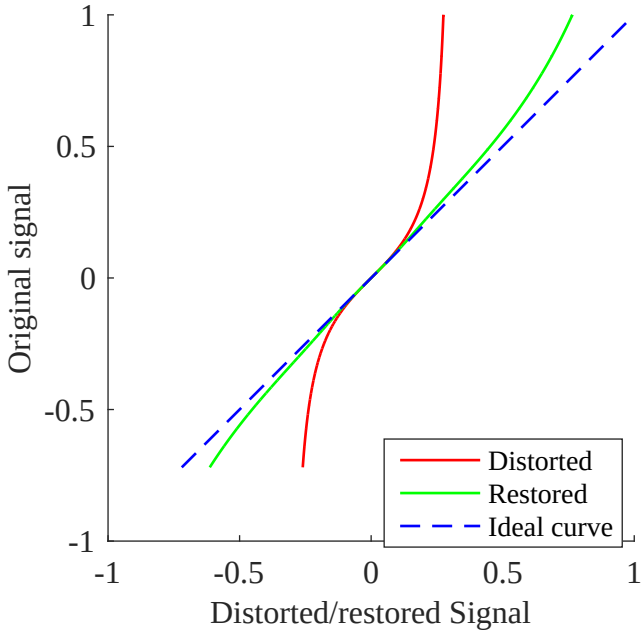

Figure 10.49: Signal `voice.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.8241809 \quad 0.00022433054 \quad 3131.1817]^T$. Rnonlin grade of restored signal: 0.98109.

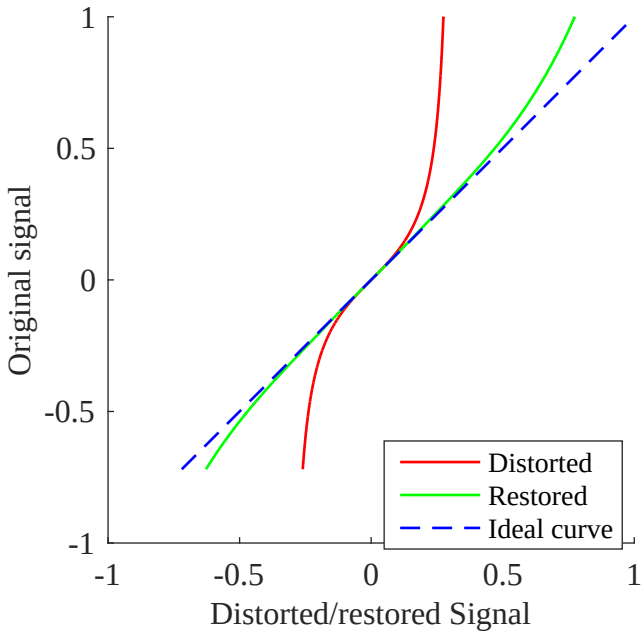

Figure 10.50: Signal `voice.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 4 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 8.2542508 \quad 3.4241203\times10^{-6} \quad 2779.2316]^T$. Rnonlin grade of restored signal: 0.98549.

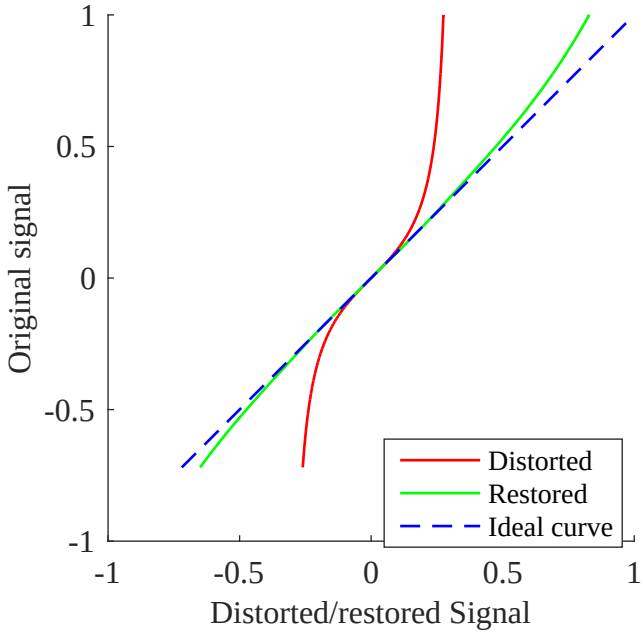

Figure 10.51: Signal `voice.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 10.3912932 \quad 6.38860371 \quad 0.00312219062 \quad 36923.8279]^T$. Rnonlin grade of restored signal: 0.987.

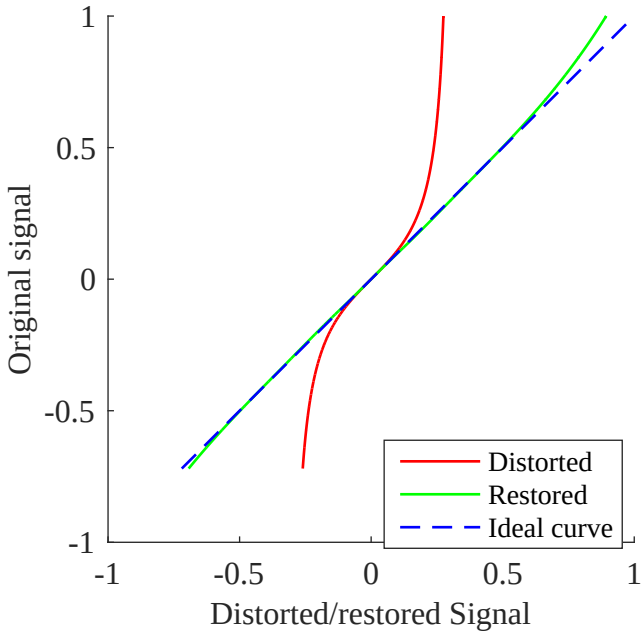

Figure 10.52: Signal `voice.wav` and arctan distortion with $\lambda = 5$: Distorted and restored signals plotted against the original one, by estimating 5 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 11.4995107 \quad 8.66801619\times10^{-6} \quad 7.84890812\times10^{-5} \quad 42730.0166]^T$. Rnonlin grade of restored signal: 0.99234.

10.3.3 Signals corrupted with noise

A great advantage of this method over the previously presented Bayesian one is that it is capable of dealing with noise. It is known that in this context the restoration procedures based on the AR model do not perform well [45], while the sparsity context is very adequate to incorporate noise, since it also corrupts the sparsity of the original undistorted signal. We made tests considering additive white noise before and after the nonlinearity, as discussed in more detail below.

Some caution is needed when applying the estimated polynomial to the noisy signal: if the noise was added after the nonlinearity it is not recommended to apply the estimated polynomial directly into the noisy signal, because some amplification of the noise can occur, being then advisable to use some denoising tool beforehand; if the noise is added before the nonlinearity, then the estimated polynomial can be directly applied to the distorted signal, in order to recover the noisy (but not distorted anymore) signal, and if the user desires, some denoising tool can be employed now. The main point illustrated in these results is that the nonlinearity is well estimated even in noisy signals, and *not* that the original noiseless signal is recovered by the proposed method.

The signal `flute.wav` was distorted by the inverse of $g(y) = y + 5y^3 + 30y^5$, and white noise was added before and after the distortion, with SNR of 40, 30 and 20.

The correct number of coefficients was estimated, with values of σ equal to 0.01, 0.001 and 0.0001. Results are shown in the figures below.

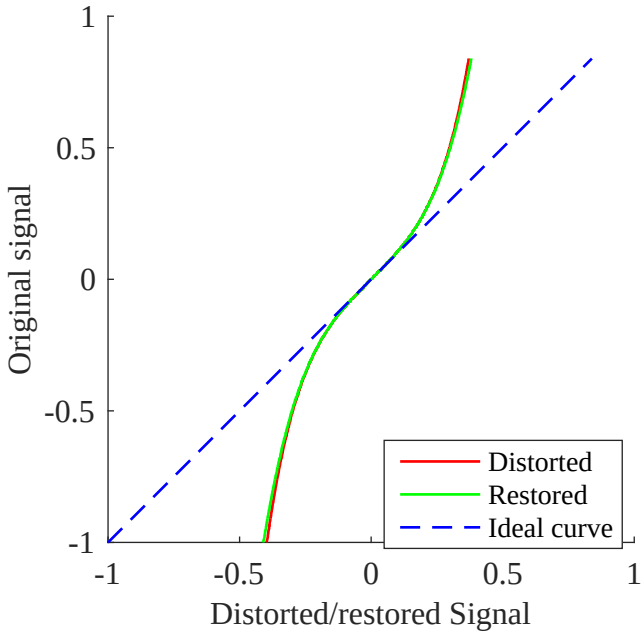

Figure 10.53: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 40 dB added before the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.01$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 0.00021312 \quad 1.3451]^T$.

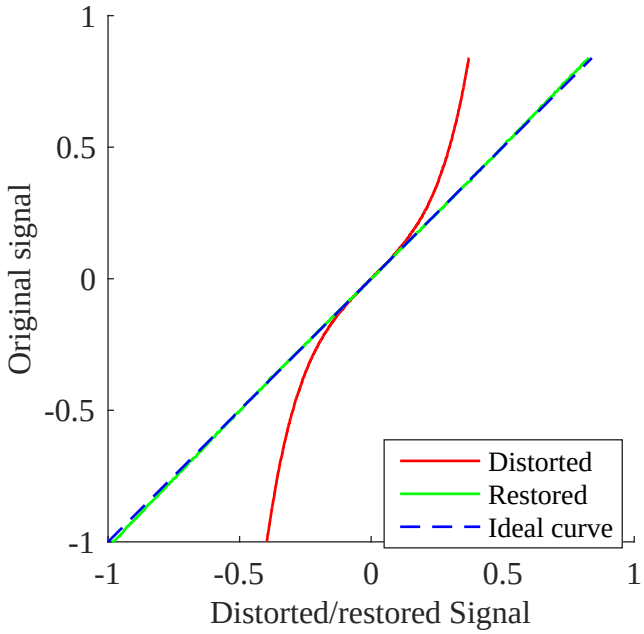

Figure 10.54: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 40 dB added before the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.2954 \quad 26.0273]^T$.

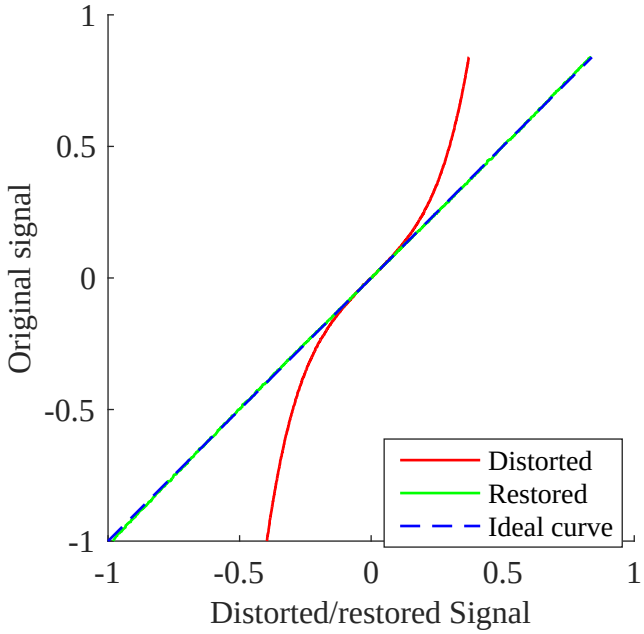

Figure 10.55: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 40 dB added before the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.4108 \quad 26.0804]^T$.

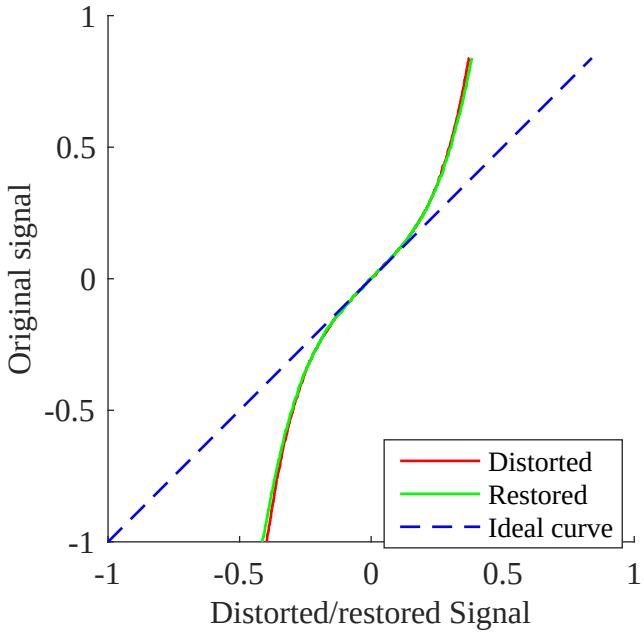

Figure 10.56: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 30 dB added before the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.01$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 4.4298{\times}10^{-5} \quad 1.593]^T$.

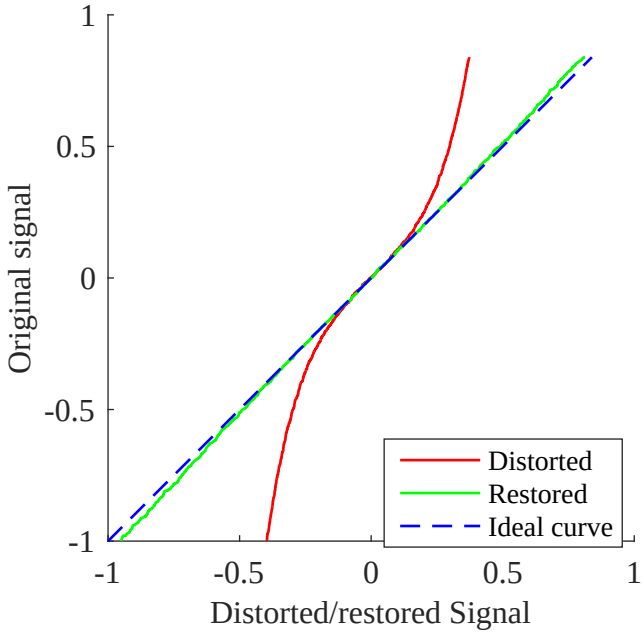

Figure 10.57: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 30 dB added before the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.10902 \quad 23.9181]^T$.

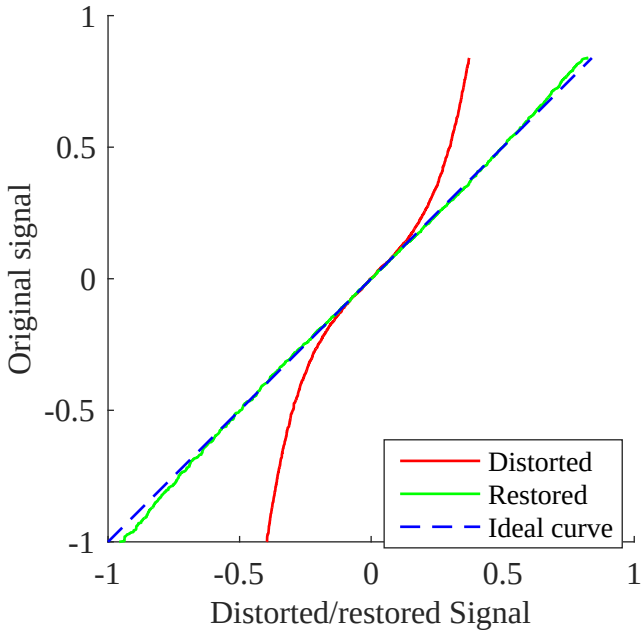

Figure 10.58: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 30 dB added before the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 6.18909 \quad 16.4272]^T$.

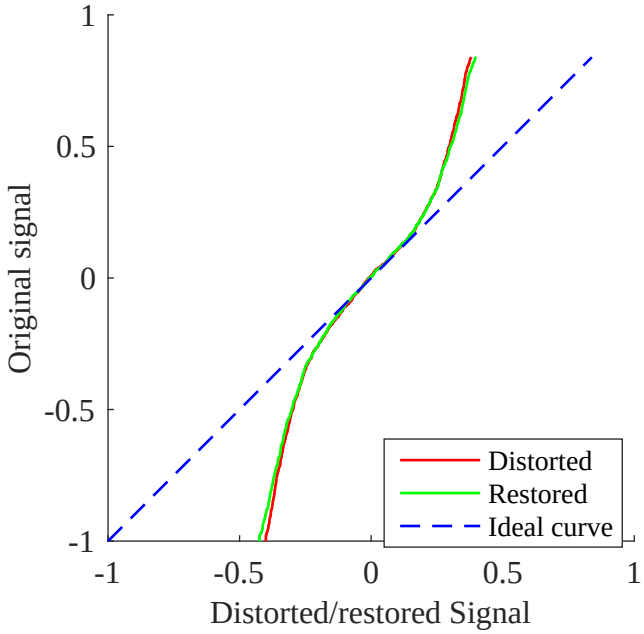

Figure 10.59: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 20 dB added before the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.01$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 0.00019273 \quad 2.2457]^T$.

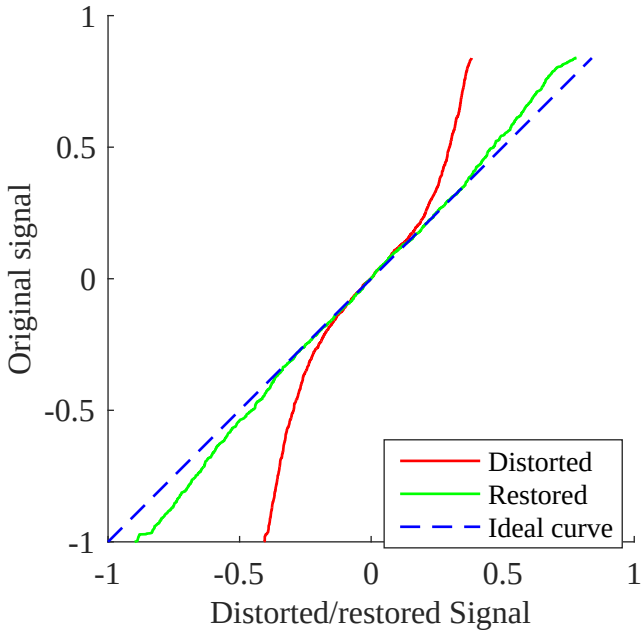

Figure 10.60: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 20 dB added before the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.07302 \quad 14.6999]^T$.

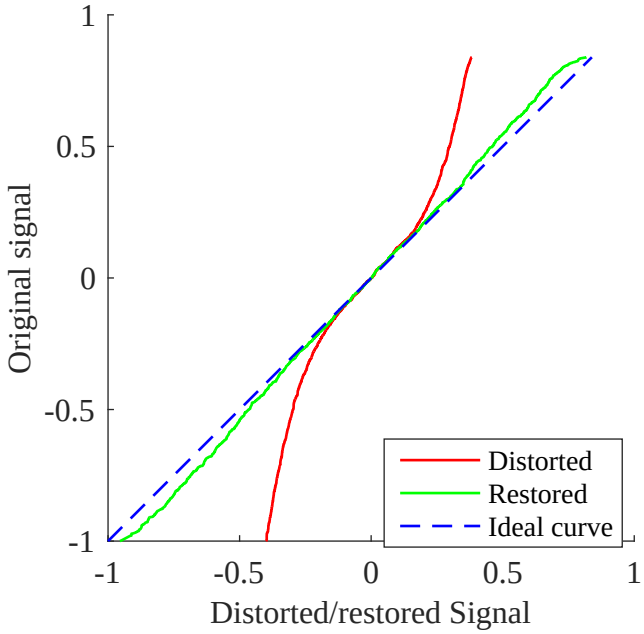

Figure 10.61: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 20 dB added before the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 3.41822 \quad 30.3089]^T$.

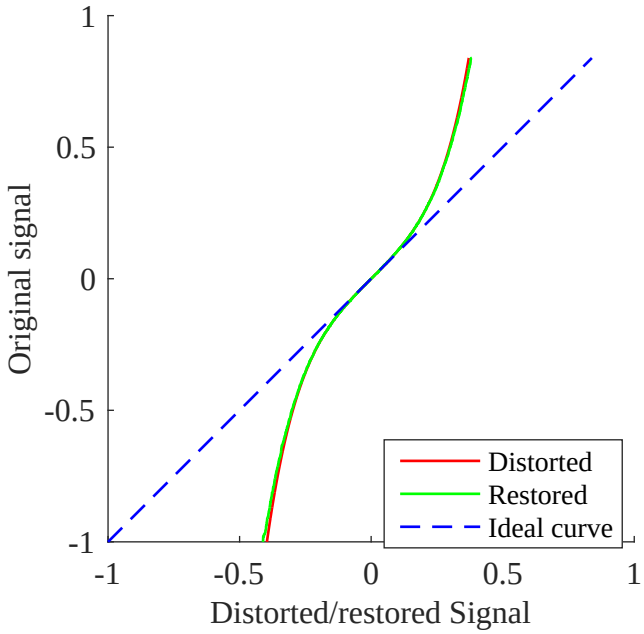

Figure 10.62: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 40 dB added after the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.01$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 4.05\times10^{-5} \quad 1.283]^T$.

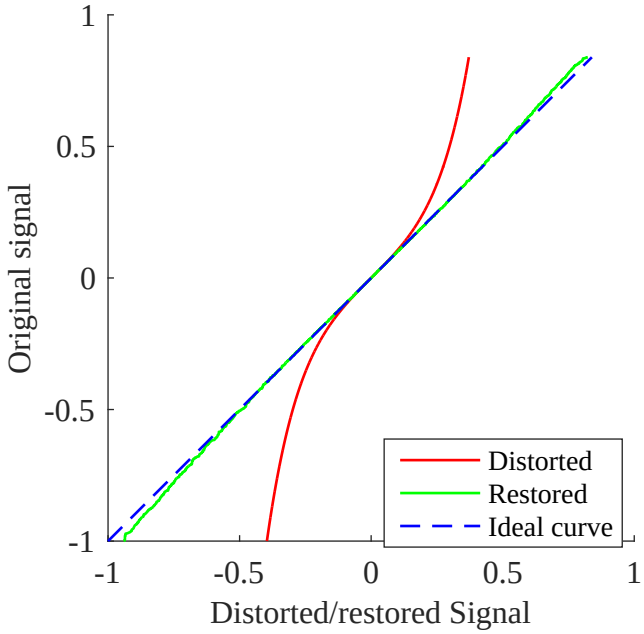

Figure 10.63: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 40 dB added after the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.57332 \quad 21.0047]^T$.

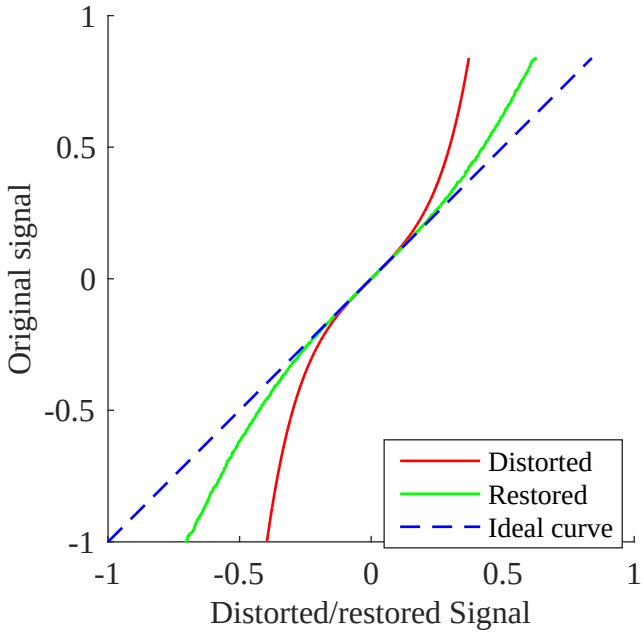

Figure 10.64: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 40 dB added after the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 4.8327 \quad 1.703{\times}10^{-6}]^T$.

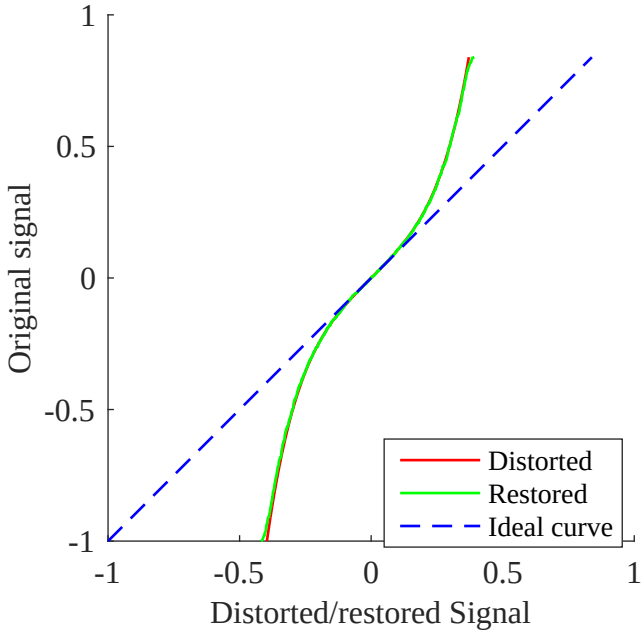

Figure 10.65: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 30 dB added after the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.01$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.333{\times}10^{-5} \quad 0.7367]^T$.

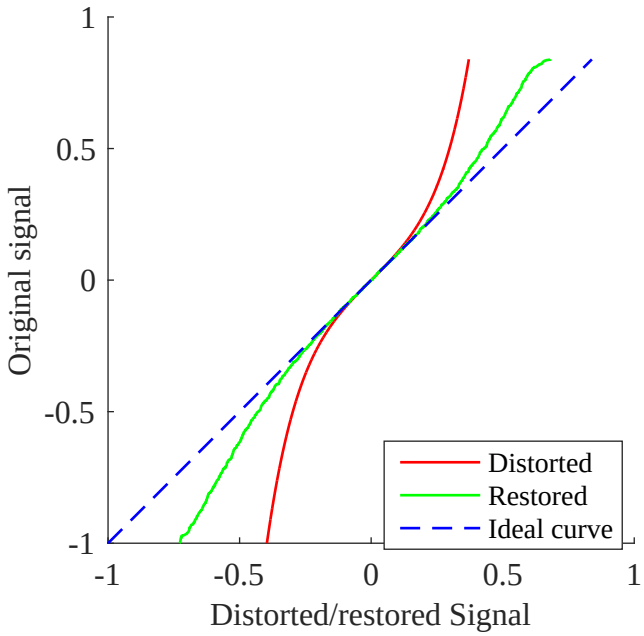

Figure 10.66: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 30 dB added after the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 5.0044 \quad 3.4084{\times}10^{-6}]^T$.

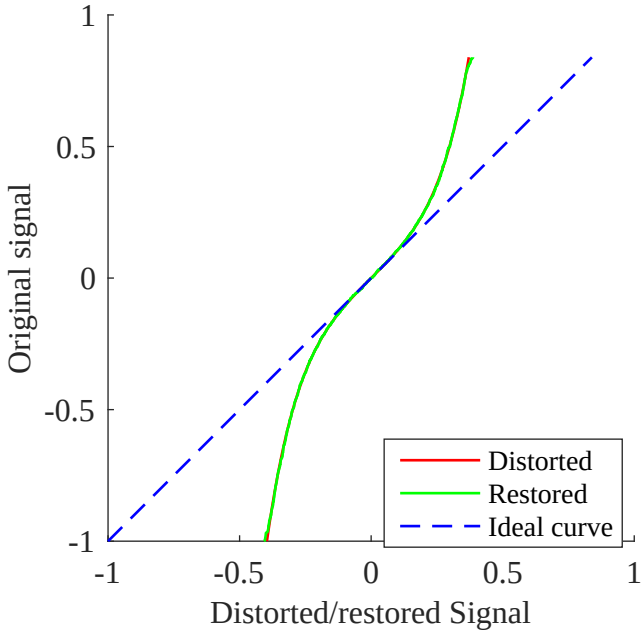

Figure 10.67: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 30 dB added after the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 1.1117{\times}10^{-7} \quad 4.1482{\times}10^{-7}]^T$.

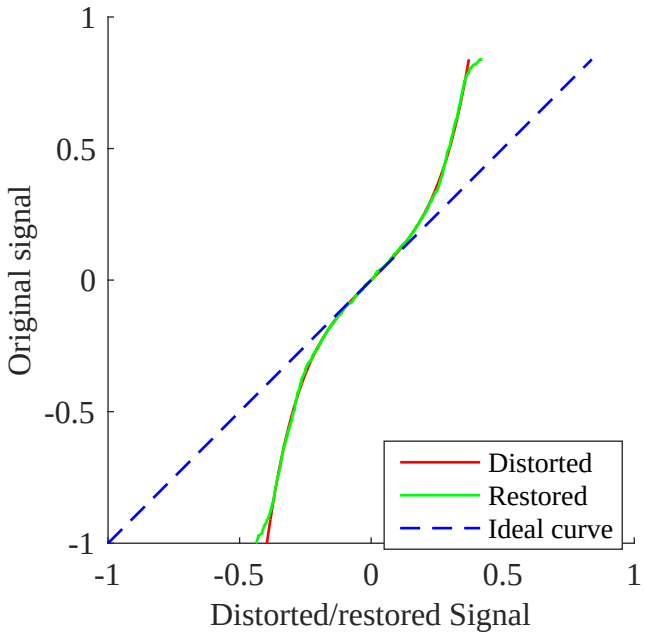

Figure 10.68: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 20 dB added after the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.01$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 1.036\times10^{-6} \quad 4.9218\times10^{-6}]^T$.

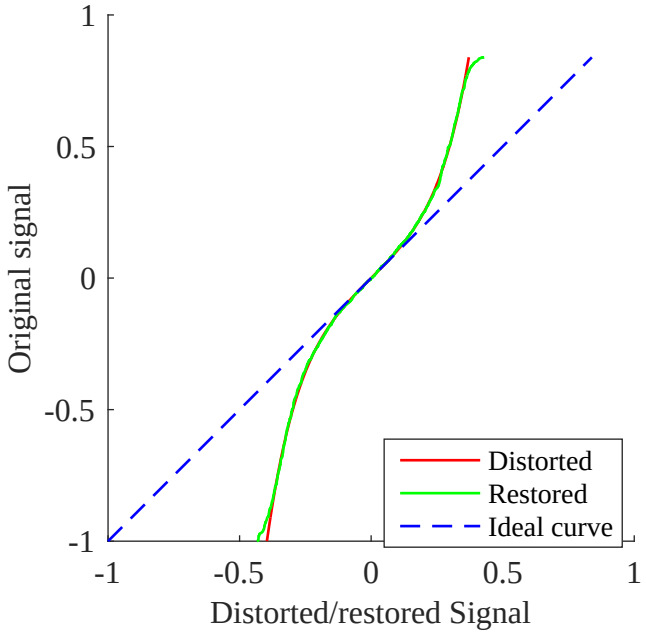

Figure 10.69: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 20 dB added after the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 3.5346\times10^{-6} \quad 1.9985\times10^{-6}]^T$.

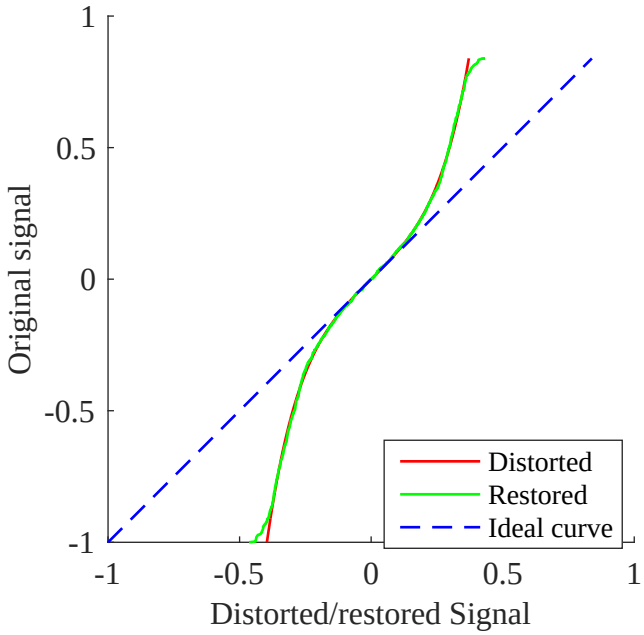

Figure 10.70: Signal `flute.wav`, distortion with polynomial inverse and noise with SNR of 20 dB added after the nonlinearity: Distorted and restored signals plotted against the original one, by estimating 3 polynomial coefficients and with $\sigma = 0.0001$. Estimated coefficients (after normalization): $\mathbf{m} = [1 \quad 1.5062\times10^{-5} \quad 6.9605\times10^{-5}]^T$.

We note that the estimate of $\mathbf{m}$ is better when the noise is added before the nonlinearity: even for a SNR of 20 dB the estimated values are quite close to the correct ones. However, when the noise is added after the nonlinearity, the estimate is not so accurate: for a SNR of 20 dB the estimated values of $\mathbf{m}$ are far from the true ones, for a SNR of 30 dB and $\sigma = 0.001$ only the first coefficient was correctly estimated, and for a SNR of 40 dB and $\sigma = 0.001$ the estimate is quite good.

10.4 Conclusion and future works

The set of good results presented here together with the great computational efficiency of the proposed algorithm indicates that this approach is very promising. When dealing with distortions following the model, even when the number of coefficients being estimated is different from the original one, the result is quite good, a fact that is confirmed by the plots shown above, the Rnonlin grades and by the informal subjective tests performed. When the number of coefficients being estimated is larger than the real value, the remaining estimated values are close to zero, a fact that is reasonable to expect. When more general distortions were considered, the method is also capable of estimating a set of polynomial coefficients that largely mitigates the distortion. And finally, when dealing with noisy signals, the method

is also capable of correctly estimating the polynomial coefficients, at least in some scenarios.

The main advantage of this framework when comparing to the Bayesian one is its computational efficiency: for example, the extensive tests with the signal `flute.wav` with distortion following the model, including the computation of the Rnonlin grades, lasted less than 3 minutes[2]. On the other hand, this method gives only a point estimate, whereas with the Bayesian one much more information about the credibility of the estimate is available.

Since our main goal is to treat nonlinear distortions with memory, generalizing this proposed solution to this scenario is the main theme to be addressed in future works. Preliminary attempts were performed, but it is not clear under which conditions regarding the distortion the same solution will also perform well. Moreover, since the linear filter after the nonlinear distortion in the Hammerstein model does not affect the sparsity of the original signal, maybe some modification of the objective function will lead to a better result.

[2]Recall that in Chapter 9 the estimation of a single set of coefficients using the Gibbs sampler lasted approximately 41 s.

Chapter 11

Bayesian again: Treating memoryless nonlinear distortions from sparse and Bayesian viewpoints

This chapter is an extension of the previous one, where the problem of memoryless nonlinear distortion in audio signals was tackled from a different perspective that dispenses the AR model. Preliminary tests using the former technique to treat nonlinear distortions with memory indicated that some modifications should be done in the objective function, which then turns very complicated, and amenable to exhibit multiple local minima. This approach is also under development, but we chose to firstly consider the sparsity-based approach in a Bayesian context. In [93], the problem of Compressive Sensing is tackled from a Bayesian perspective, using adequate priors to model the sparsity in some domain of the signal one wishes to recover. Here we present an adaptation of this work to the case of nonlinear measurements where the nonlinearity is unknown, together with some modifications needed to correctly estimate the nonlinearity.

This Chapter is organized as follows: firstly in Section 11.1 we describe how it is possible to induce sparsity via an adequate choice of prior distributions; next, in Section 11.2 the problem of memoryless nonlinearities is formulated in a Bayesian context, and in Section 11.3 the optimization procedure employed is described; finally, in Section 11.4 some results are presented, and in Section 11.5 further developments are proposed.

11.1 How to induce sparsity via prior distributions?

Consider the classical least squares problem:

$$\min_{\mathbf{x}} ||\mathbf{A}\mathbf{x} - \mathbf{y}||_2, \tag{11.1}$$

where $||\cdot||_2$ denotes the ℓ_2 norm. One possible way to interpret this problem is the following: maybe the linear system $\mathbf{A}\mathbf{x} = \mathbf{y}$ has no solution, but one wishes to find the *best* solution, that is, the value (or possibly values) of $\mathbf{x}$ that are closer to being a genuine solution of the considered linear system.

From a statistical viewpoint, we can think of Equation 11.1 as a linear regression problem: vector $\mathbf{y}$ contains a set of observations that one expects to be linearly related to a set of (unknown) coefficients stored in vector $\mathbf{x}$ via (known) matrix $\mathbf{A}$; and such measurements are possibly corrupted with noise that one supposes to be Gaussian with variance equal to σ_e^2. Therefore, the observed data can be written in the following way:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}, \tag{11.2}$$

where the entries of vector $\mathbf{e}$ are iid Gaussian with 0 mean and variance σ_e^2. It is then possible to write the likelihood for the desired parameters:

$$p(\mathbf{y}|\mathbf{x}) \propto \exp\left(-\frac{1}{2\sigma_e^2}||\mathbf{A}\mathbf{x} - \mathbf{y}||_2\right). \tag{11.3}$$

Maximizing Equation 11.3 with respect to $\mathbf{x}$ in order to obtain its maximum likelihood estimator leads exactly to the problem in Equation 11.1.

It is known that in some cases the maximum likelihood estimator to $\mathbf{x}$ can amplify the noise present in the observations [17, 37], so it is convenient to regularize the problem. Suppose that one knows *a priori* that the coefficients in $\mathbf{x}$ cannot be arbitrarily large. It is then reasonable to suppose that $\mathbf{x}$ is drawn from a Gaussian distribution with $\mathbf{0}$ mean and some covariance matrix $\mathbf{\Gamma}$, that is,

$$p(\mathbf{x}) \propto \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{\Gamma}^{-1}\mathbf{x}\right). \tag{11.4}$$

By multiplying Equation 11.4 with the likelihood in Equation 11.3, we obtain the posterior distribution of $\mathbf{x}$, via Bayes' Theorem:

$$p(\mathbf{x}|\mathbf{y}) \propto \exp\left(-\frac{1}{2\sigma_e^2}||\mathbf{A}\mathbf{x} - \mathbf{y}||_2 - \frac{1}{2}\mathbf{x}^T\mathbf{\Gamma}^{-1}\mathbf{x}\right), \tag{11.5}$$

which if maximized leads to the maximum a posteriori estimator, given by

$$\min_{\mathbf{x}} ||\mathbf{A}\mathbf{x} - \mathbf{y}||_2 + ||\mathbf{\Gamma}\mathbf{x}||_2. \tag{11.6}$$

This is the well known *Tikhonov*[1] *regularization*, and a closed form to this estimator is available, but it will not be of use here.

However, this regularization only carries information about the overall magnitude of $\mathbf{x}$, not about its components individually. If one also knows that only a few entries of $\mathbf{x}$ are different from zero, for example, this regularization is not adequate, and one can rather use the following prior "distribution"[2]:

$$p(\mathbf{x}) \propto \exp\left(-\tau||\mathbf{x}||_0\right), \tag{11.7}$$

where $||\cdot||_0$ is the ℓ_0 "norm", which leads to the following optimization problem:

$$\min_{\mathbf{x}} ||\mathbf{A}\mathbf{x} - \mathbf{y}||_2 + \tau||\mathbf{x}||_0. \tag{11.8}$$

This problem is known to be NP-hard [91], and as discussed previously in Chapter 10, since the considered signals are not exactly sparse, we must employ a more flexible prior distribution. One can change the ℓ_0 "norm" by its tightest convex relaxation, the ℓ_1 norm, which leads to the optimization problem below, much easier to solve and known to also induce (approximate) sparsity in vector $\mathbf{x}$ [90]:

$$\min_{\mathbf{x}} ||\mathbf{A}\mathbf{x} - \mathbf{y}||_2 + \lambda||\mathbf{x}||_1. \tag{11.9}$$

In the statistical community, this regularization is called the LASSO (least absolute shrinkage and selection operator), and was introduced in 1996 [94]. See also [95]. In a Bayesian context, Equation 11.9 can be interpreted as if independent Laplace priors have been imposed onto every entry of $\mathbf{x}$:

$$p(\mathbf{x}) \propto \exp\left(-\lambda||\mathbf{x}||_1\right). \tag{11.10}$$

Using directly Laplace priors is not convenient, because after the required change of variables, the obtained distribution will be very complicated. Fortunately there is a way of writing the Laplace distribution as a mixture of Gaussian and Gamma distributions, which will be better detailed in the next Section.

[1] Andrey Nikolayevich Tikhonov, Gzhatsk, October 30, 1906 – Moscow, October 7, 1993.

[2] Note that this is not a proper probability distribution. Since using improper prior distributions is recurrent in Bayesian statistics and we are not interested in the respective posterior distribution but only the ensuing optimization problem, we will not further discuss this point.

11.2 Formulation of the problem in a Bayesian context

Let $\mathbf{x}$ be an excerpt of an audio signal of length N, supposed to be sparse in the DCT domain (a reasonable assumption, as explained in Chapter 10). Let $\mathbf{\Psi}$ be the DCT matrix, such that $\mathbf{w} = \mathbf{\Psi x}$ are the DCT coefficients of signal $\mathbf{x}$.

As before, let $f(.)$ be the nonlinear distorting function, that is, the observed distorted signal $\mathbf{y}$ is given by $f(\mathbf{x})$, where function $f(.)$ is applied entry-wise in vector $\mathbf{x}$. Suppose $f(.)$ to be anti-symmetric and monotonically increasing. Given such hypotheses, which guarantee that $f(.)$ is invertible, denote its inverse by $g(.)$. We parametrize it via its truncated Taylor series around zero:

$$g(y) = m_1 y + m_2 y^3 + m_3 y^5 + \cdots + m_M y^{2M-1}, \tag{11.11}$$

and assemble the corresponding coefficients in vector $\mathbf{m}$. Estimating $g(.)$ is equivalent to estimating vector $\mathbf{m}$.

Remember that nonlinearities are known to create new frequency components in the original signal, as discussed in Chapter 5. Since the signals being considered are compressible, we cannot assure that previously to the distortion some frequency components are exactly zero, but only sufficiently small. This fact about nonlinearities can be adapted to this scenario by noting that these sufficiently small frequency components will not be so small after the distortion. In other words, the distorted signal is "less compressible" in the DCT domain. This justifies the imposition of a Laplace prior onto every component of $\mathbf{w}$ that will be constructed in a hierarchical way, as previously stated:

$$\mathbf{w}|\boldsymbol{\gamma} \sim \prod_{i=1}^{N} N(w_i|0, \gamma_i) = N(\mathbf{w}|\mathbf{0}, \mathrm{diag}(\boldsymbol{\gamma})) \tag{11.12}$$

$$\boldsymbol{\gamma}|\lambda \sim \prod_{i=1}^{N} \Gamma(\gamma_i|1, \lambda/2), \tag{11.13}$$

where $\Gamma(\cdot|\alpha, \beta)$ denotes the Gamma distribution with shape and scale parameters given respectively by α and β, and $\mathrm{diag}(\boldsymbol{\gamma})$ denotes a diagonal matrix with vector $\boldsymbol{\gamma}$ along the diagonal. It can be shown that the following result is valid [93]:

$$p(\mathbf{w}|\lambda) = \int_0^{+\infty} p(\mathbf{w}|\boldsymbol{\gamma}) p(\boldsymbol{\gamma}|\lambda)\, d\boldsymbol{\gamma} = \frac{\lambda^{N/2}}{2^N} \exp\left(-\sqrt{\lambda} \sum_{i=1}^{N} |w_i|\right), \tag{11.14}$$

that is, by integrating $\boldsymbol{\gamma}$ out we are really imposing a Laplace prior onto $\mathbf{w}$. The hyper-parameter λ controls the sparsity degree of $\mathbf{x}$ in the DCT domain, and in

order to add another degree of flexibility, λ can be viewed as a realization of the following hyper-prior:

$$\lambda|\nu \sim \Gamma(\lambda|\nu/2, \nu/2). \tag{11.15}$$

This prior is very flexible, since it can provide very vague or very specific information about λ, depending on the chosen value of ν. Moreover, its value can be also estimated from the data, as it will become clear.

Since the original undistorted signal is given by $\mathbf{x} = \mathbf{\Psi}^T\mathbf{w}$, its distribution is given by [17]:

$$\mathbf{x}|\boldsymbol{\gamma} \sim N(\mathbf{x}|\mathbf{0}, \mathbf{\Psi}^T\text{diag}(\boldsymbol{\gamma})\mathbf{\Psi}), \tag{11.16}$$

and after the change of variables $\mathbf{x} = g(\mathbf{y})$ (already performed in Chapter 7), we conclude that:

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\gamma}, \mathbf{m}) = &\frac{\prod_{i=1}^{N} |g'(y_i)|}{(2\pi)^{N/2}|\det(\mathbf{\Psi}^T\text{diag}(\boldsymbol{\gamma})\mathbf{\Psi})|^{1/2}} \times \\ &\exp\left\{-\frac{1}{2}g(\mathbf{y})^T(\mathbf{\Psi}^T\text{diag}(\boldsymbol{\gamma})\mathbf{\Psi})^{-1}g(\mathbf{y})\right\}. \end{aligned} \tag{11.17}$$

This distribution seems cumbersome, but some of its terms can be further simplified:

- By noting that $\det(\mathbf{\Psi}) = 1$, the determinant term can be rewritten as:

$$\begin{aligned} |\det(\mathbf{\Psi}^T\text{diag}(\boldsymbol{\gamma})\mathbf{\Psi})|^{1/2} &= |\det(\mathbf{\Psi}^T)\det(\text{diag}(\boldsymbol{\gamma}))\det(\mathbf{\Psi})|^{1/2} \\ &= |\det(\text{diag}(\boldsymbol{\gamma}))|^{1/2} \qquad (11.18) \\ &= \prod_{i=1}^{N} \gamma_i^{1/2}; \qquad (11.19) \end{aligned}$$

- The inverse matrix inside the exponential is given by

$$(\mathbf{\Psi}^T\text{diag}(\boldsymbol{\gamma})\mathbf{\Psi})^{-1} = \mathbf{\Psi}^T\text{diag}(1/\boldsymbol{\gamma})\mathbf{\Psi} = \sum_{i=1}^{N} \frac{1}{\gamma_i}\psi_i^T\psi_i, \tag{11.20}$$

  where ψ_i is the i-th line of matrix $\mathbf{\Psi}$. Therefore,

$$\begin{aligned} &g(\mathbf{y})^T(\mathbf{\Psi}^T\text{diag}(\boldsymbol{\gamma})\mathbf{\Psi})^{-1}g(\mathbf{y}) \\ &= g(\mathbf{y})^T\left(\sum_{i=1}^{N} \frac{1}{\gamma_i}\psi_i^T\psi_i\right)g(\mathbf{y}) \\ &= \sum_{i=1}^{N} \frac{1}{\gamma_i}\underbrace{g(\mathbf{y})^T\psi_i^T\psi_i g(\mathbf{y})}_{:=\xi_i} = \sum_{i=1}^{N} \frac{\xi_i}{\gamma_i}. \end{aligned} \tag{11.21}$$

With these simplifications, we have that

$$p(\mathbf{y}|\boldsymbol{\gamma}, \mathbf{m}) \propto \left(\prod_{i=1}^{N} \frac{|g'(y_i)|}{\gamma_i^{1/2}} \right) \exp\left\{ -\frac{1}{2} \sum_{i=1}^{N} \frac{\xi_i}{\gamma_i} \right\}, \tag{11.22}$$

where

$$\xi_i = g(\mathbf{y})^T \psi_i^T \psi_i g(\mathbf{y}). \tag{11.23}$$

By using Bayes' Theorem, it is then possible to write the posterior distribution of the parameters:

$$p(\boldsymbol{\gamma}, \lambda, \mathbf{m}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\gamma}, \mathbf{m}) p(\boldsymbol{\gamma}|\lambda) p(\lambda) p(\mathbf{m}), \tag{11.24}$$

where $p(\lambda)$ and $p(\boldsymbol{\gamma}|\lambda)$ are given by adequate Gamma distributions, as stated above. Two different priors for $\mathbf{m}$ will be considered, and for the moment we will keep the term $p(\mathbf{m})$. By applying the formulas for the distributions into the right hand side of Equation 11.24, we have that:

$$\begin{aligned} p(\boldsymbol{\gamma}, \lambda, \mathbf{m}|\mathbf{y}) \propto & \left[\left(\prod_{i=1}^{N} \frac{|g'(y_i)|}{\gamma_i^{1/2}} \right) \exp\left\{ -\frac{1}{2} \sum_{i=1}^{N} \frac{\xi_i}{\gamma_i} \right\} \right] \times \left[\prod_{i=1}^{N} \frac{\lambda}{2} e^{-\frac{\lambda}{2}\gamma_i} \right] \times \\ & \left[\frac{(\nu/2)^{\nu/2}}{\Gamma(\nu/2)} \lambda^{\nu/2-1} e^{-\frac{\nu}{2}\lambda} \right] \times p(\mathbf{m}). \end{aligned} \tag{11.25}$$

Note that despite this distribution depending on ν, this variable is not written as an argument of the function, since it can be kept fixed during the optimization procedure in order to ensure some desired behavior of λ, depending on the will of the user.

11.3 Maximization of the posterior distribution

Some attempts to directly treat this distribution have been tried, for example, maximizing its logarithm with respect to $\boldsymbol{\gamma}$, λ, ν and $\mathbf{m}$, as presented in [93]. Unfortunately, regardless of the choice of prior distribution for $\mathbf{m}$, this procedure was shown to be inadequate here because of some convergence issues found in the optimization algorithm, whose reason is a point to be further investigated.

Since $\boldsymbol{\gamma}$ is a nuisance parameter and all computations involving the hierarchical prior for $\mathbf{w}$ have already been performed, we can integrate it out and obtain the distribution $p(\lambda, \mathbf{m}|\mathbf{y})$. Integrating λ out would also be adequate, but this computation is impossible to be carried out by hand, as it will become clear soon. We now describe the procedure to integrate Equation 11.25 in $\boldsymbol{\gamma}$.

If we consider the terms in Equation 11.25 that depend only on $\boldsymbol{\gamma}$, we obtain

after some simplifications:

$$\prod_{i=1}^{N} \frac{1}{\gamma_i^{1/2}} \exp\left\{-\frac{1}{2}\left(\frac{\xi_i}{\gamma_i} + \lambda\gamma_i\right)\right\}, \tag{11.26}$$

which if integrated with respect to $\boldsymbol{\gamma}$ leads to

$$\prod_{i=1}^{N} \int_0^{+\infty} \frac{1}{\gamma_i^{1/2}} \exp\left\{-\frac{1}{2}\left(\frac{\xi_i}{\gamma_i} + \lambda\gamma_i\right)\right\} d\gamma_i. \tag{11.27}$$

Fortunately, this integral can be computed by hand, at the cost of a very long and tedious derivation. In fact, the indefinite integral

$$\int \frac{1}{x^{1/2}} \exp\left\{-\frac{1}{2}\left(\frac{a}{x} + bx\right)\right\} dx \tag{11.28}$$

can be computed, and its solution is given by[3]

$$\sqrt{\frac{\pi}{2b}}\left[e^{-\sqrt{ab}}\left(\Phi\left(\frac{x\sqrt{b}-\sqrt{a}}{\sqrt{2x}}\right)+1\right) + e^{\sqrt{ab}}\left(\Phi\left(\frac{\sqrt{a}+x\sqrt{b}}{\sqrt{2x}}\right)-1\right)\right], \tag{11.29}$$

where Φ is the *error function*, given by

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \tag{11.30}$$

With the indefinite integral, we can obtain

$$\int_0^{+\infty} \frac{1}{x^{1/2}} \exp\left\{-\frac{1}{2}\left(\frac{a}{x} + bx\right)\right\} dx = \sqrt{\frac{2\pi}{b}} e^{-\sqrt{ab}}, \tag{11.31}$$

and after substituting adequate values for a and b, we conclude that:

$$\prod_{i=1}^{N} \int_0^{+\infty} \frac{1}{\gamma_i^{1/2}} \exp\left\{-\frac{1}{2}\left(\frac{\xi_i}{\gamma_i} + \lambda\gamma_i\right)\right\} d\gamma_i = \frac{(2\pi)^{N/2}}{\lambda^{N/2}} \exp\left\{-\sqrt{\lambda}\sum_{i=1}^{N}\sqrt{\xi_i}\right\}. \tag{11.32}$$

[3]Thanks a lot, WolframAlpha! But in fact this procedure is not needed, since this function can be recognized as the kernel of the Generalized Inverse Gaussian (GIG) distribution [96], and its respective normalizing constant can be used to obtain the desired result.

Therefore, the integrated posterior distribution is given by

$$\begin{aligned} p(\lambda, \mathbf{m}|\mathbf{y}) &= \int_0^{+\infty} p(\boldsymbol{\gamma}, \lambda, \mathbf{m}|\mathbf{y})\ d\boldsymbol{\gamma} \\ &\propto \left[\prod_{i=1}^{N} |g'(y_i)|\right] \times \left[\frac{\lambda}{2}\right]^N \times \left[\frac{(\nu/2)^{\nu/2}}{\Gamma(\nu/2)}\lambda^{\nu/2-1}e^{-\frac{\nu}{2}\lambda}\right] \\ &\quad \times p(\mathbf{m}) \times \left[\lambda^{-N/2}\exp\left\{-\sqrt{\lambda}\sum_{i=1}^{N}\sqrt{\xi_i}\right\}\right]. \end{aligned} \tag{11.33}$$

Instead of maximizing this distribution with respect to λ, $\mathbf{m}$ and ν, we equivalently maximize its logarithm, given by:

$$\begin{aligned} \log p(\lambda, \mathbf{m}|\mathbf{y}) &= \sum_{i=1}^{N} \log|g'(x_i)| + N\log\left(\frac{\lambda}{2}\right) + \frac{\nu}{2}\log\left(\frac{\nu}{2}\right) - \log\Gamma\left(\frac{\nu}{2}\right) \\ &\quad + \left(\frac{\nu}{2} - 1\right)\log(\lambda) - \frac{\nu}{2}\lambda + \log p(\mathbf{m}) \\ &\quad - \frac{N}{2}\log(\lambda) - \sqrt{\lambda}\sum_{i=1}^{N}\sqrt{\xi_i}. \end{aligned} \tag{11.34}$$

Unfortunately, regardless of the prior distribution for $\mathbf{m}$, it is not possible to obtain analytical expressions for the values of λ, $\mathbf{m}$ and ν that maximize this function, and a numerical algorithm is employed to this end; the procedure is presented in more details in Section 11.4, together with some results. To close the theoretical exposition, we discuss the prior distributions for $\mathbf{m}$.

11.3.1 Non-informative prior for m

Recall that the prior distributions in Bayesian Statistics can be seen as regularization of ill-posed problems, as discussed in the beginning of this chapter. Up to this point, the regularization of variable $\mathbf{m}$ was performed in this way: firstly recall, as discussed in Chapter 7, that if some regularization is not imposed, any multiple of the inverse of the real distorting function is a possible estimate; to avoid this issue, we imposed $g'(0) = 1$, which implies that its first coefficient is equal to one.

Therefore, the prior distribution imposed over the coefficients $\mathbf{m}_{2:M}$ was a vague one, given by

$$p_{\text{NI}}(\mathbf{m}_{2:M}) \propto \exp\left(-\frac{1}{2\sigma_m^2}\mathbf{m}_{2:M}^T\mathbf{m}_{2:M}\right), \tag{11.35}$$

where $\mathbf{m}_{2:M}$ denotes the entries of $\mathbf{m}$ from the second to the last, and the value of σ_m^2 was chosen to be large enough to make this prior vague. The interpretation of this distribution is that little is known about $\mathbf{m}_{2:M}$, and thus we do not want to restrict its possible values. We will denote this prior by p_{NI}, being the subscript

"NI" used to indicate when the non-informative prior is being used.

With this restriction over $\mathbf{m}$, function $g(.)$ can be rewritten as

$$g(y) = y + m_2 y^3 + m_3 y^5 + \cdots + m_M y^{2M-1}, \tag{11.36}$$

and the relation between $\mathbf{x}$ and $\mathbf{y}$ is written linearly in $\mathbf{m}_{2:M}$ as

$$\mathbf{x} = \mathbf{y} + \mathbf{Y}\mathbf{m}_{2:M}, \tag{11.37}$$

where the matrix $\mathbf{Y}$ is the same as in Chapter 7.

11.3.2 Informative prior for m

Recall that in Chapter 10 we have not restricted any particular entry of $\mathbf{m}$, but rather the norm of the restored signal. This was motivated by the fact that $\mathbf{m} = \mathbf{0}$ will always be a trivial solution to the proposed optimization problem – a solution that must be avoided. Directly imposing this condition by means of a degenerate prior distribution for $\mathbf{m}$ means to say that $\|g(\mathbf{y})\|_2 = 1$, an unnecessary geometric restriction over $\mathbf{m}$ that we are able to avoid.

Firstly, note that with this restriction it is not necessary to equate the value of m_1 to one, thus we will consider the entire vector $\mathbf{m}$ such that the relation between $\mathbf{x}$ and $\mathbf{y}$ can be written as

$$\mathbf{x} = \mathbf{Y}_{\mathrm{s}}\mathbf{m}, \tag{11.38}$$

where matrix $\mathbf{Y}_{\mathrm{s}}$ is the same as in Chapter 10.

The desired information we wish to incorporate into the model can be induced in a soft way: it is not strictly necessary to have the norm of the restored signal exactly equal to one, but only to restrict its most probable values in order to avoid several solutions to the optimization problem. We can then induce that $\|g(\mathbf{y})\|_2 \approx 1$ via the prior distribution given by

$$p_{\mathrm{I}}(\mathbf{m}) \propto \exp\left(-k(\|g(\mathbf{y})\|_2^2 - 1)^2\right) = \exp\left(-k(\mathbf{m}^T\mathbf{Y}_{\mathrm{s}}^T\mathbf{Y}_{\mathrm{s}}\mathbf{m} - 1)^2\right), \tag{11.39}$$

being the subscript "I" used to indicate the use of an informative prior. The parameter k controls how much the deviates of $\|g(\mathbf{y})\|_2^2$ from 1 are penalized and can be controlled by the user.

11.4 Results

In this section we present some results of the preliminary development presented up to this point. Tests were performed with both artificial and real signals, distorted

with artificial distortions following the model, that is, with polynomial inverse.

As in Chapter 10, the maximization of the posterior distribution in Equation 11.34 was performed via the `fmincon` function implemented in the Optimization Toolbox$^{\text{TM}}$ of MATLAB$^{\text{TM}}$.

The distorting function was the inverse of polynomial $g(y) = y + 5y^3 + 30y^5$. When using the noninformative prior we expect to recover exactly the last two coefficients of the vector $\mathbf{m} = [1 \quad 5 \quad 30]^T$, but the informative prior will give a set of coefficients tailored to ensure that the norm of restored signal is approximately equal to one. In order to better compare both estimated coefficients, in this second situation we normalize the estimate by its first value, as in Chapter 10.

When the set of polynomial coefficients is estimated via the noninformative prior we denote it by $\mathbf{m}_{\text{NI}}$, whereas if the informative prior is used it will be denoted as $\mathbf{m}_{\text{I}}$.

11.4.1 Artificial signal

A signal of length 500 was generated following the proposed model, explained in detail below:

- The value of ν was fixed at one;
- A value of λ was generated by sampling from a Gamma distribution with both parameters equal to 1/2;
- The values of γ_i for $i = 1, \ldots, 500$ were generated by independently sampling from a Gamma distribution with parameters 1 and $\lambda/2$, and assembled in vector $\boldsymbol{\gamma}$;
- The original undistorted signal $\mathbf{x}$ was generated by sampling from a Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Psi}^T \text{diag}(\boldsymbol{\gamma}) \boldsymbol{\Psi}$;
- The distorted signal $\mathbf{y}$ was obtained by artificially distorting signal $\mathbf{x}$ with the inverse of polynomial $g(y) = y + 5y^3 + 30y^5$.

As in [93], here in the case of nonlinear measurements it is also better to keep the value of ν fixed instead of variable in the objective function. In all runs of the algorithm, the value of ν was fixed at 10. It was noted that this arbitrarily chosen value almost does not impact the optimization procedure. When using the informative prior for $\mathbf{m}$ the value of k was fixed at 5, and when using the noninformative prior the value σ_m^2 was fixed at 1,000. In all cases the runtime of the optimization algorithm was very similar, from 1 to 2 minutes, approximately. This time can be improved if the computation of quantities ξ_i is performed in another way, a point that is further discussed in the next section.

The algorithm was initialized with a random value for λ following a Gamma distribution with both parameters equal to 5 and the coefficients of the polynomial equal to zero.

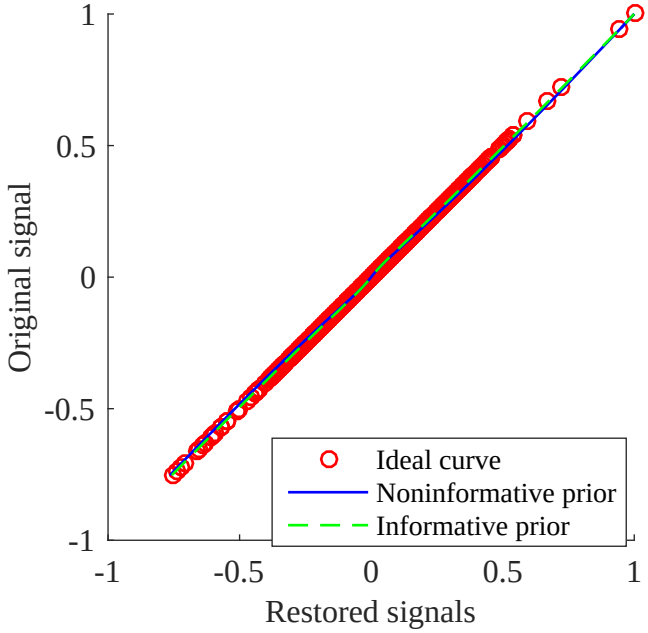

Figure 11.1: Artificial signal and distortion with polynomial inverse: signals restored via both priors for $\mathbf{m}$ plotted against the original one. Estimated coefficients (after normalization): $\mathbf{m}_{\mathrm{NI}} = [10.4098 \quad 28.0037]$ and $\mathbf{m}_{\mathrm{I}} = [1 \quad 8.4968 \quad 31.7339]$.

We note that despite the estimated coefficients not being exactly equal to the true ones, both restored signals are visually very similar, and also similar to the ideal curve. We now proceed to the tests with real signals.

11.4.2 Real signal

Tests similar to the previous were also performed with real signals. The tested signal was `flute.wav`, artificially degraded by the inverse of polynomial $g(y) = y + 5y^3 + 30y^5$. The estimation of the polynomial coefficients was done by using both noninformative and informative priors. An excerpt of 500 signal time samples around the time sample with the greatest magnitude was given as input to the algorithm.

The initialization was similar to the one in the previous section, but now the algorithm runtime is much larger, about 6 minutes for the noninformative prior and about 3 minutes for the informative prior. This difference is due to the fact that now the signal does not follow the model exactly, and then more iterations of the algorithm are needed.

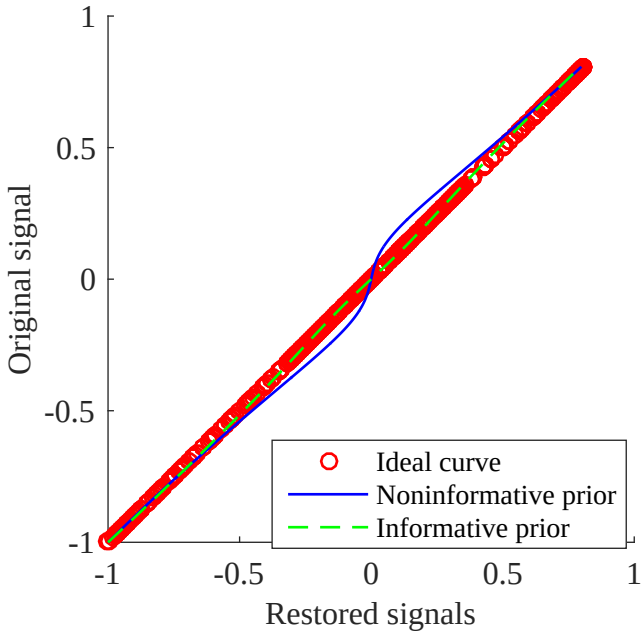

Figure 11.2: Signal `flute.wav` and distortion with polynomial inverse: signals restored via both priors for $\mathbf{m}$ plotted against the original one. Estimated coefficients (after normalization): $\mathbf{m}_{\mathrm{NI}} = [67.7055 \quad 0.125876]$ and $\mathbf{m}_{\mathrm{I}} = [1 \quad 2.04694 \quad 40.9771]$. Rnonlin grades of distorted, restored via noninformative prior and restored via informative prior, respectively: 0.87759, 0.84921 and 0.95251.

We note that the polynomial coefficients obtained via the informative prior are closer to the true ones than when obtained via the noninformative prior, a fact that is reflected in the quality of the corresponding restored signals. However, the better estimate here is not so accurate as the one obtained when using the AR model, and a possible reason to this fact is discussed in the next section.

11.5 Conclusion and future works

This is a very preliminary work and still a lot of investigation must be performed in order to better understand some aspects of the proposed algorithm. The good results obtained in the preliminary tests indicated that this approach could lead to a very effective treatment of the restoration of nonlinear distortions, even after the incorporation of memory into the model.

We note that the estimates obtained with this model are less accurate than the ones obtained when assuming an AR model for the undistorted signal. A possible reason for this fact is that audio signals are better characterized by AR models than by their compressibility in the DCT domain. Nevertheless, the second model is more robust to noisy signals, as shown in the results of Chapter 10.

A major advantage of this algorithm when compared with the one proposed in

Chapter 10 is that there is no need to arbitrarily choose some essential parameter like σ, which controls the smoothness of the objective function. The only arbitrary choice here is parameter ν, which we observed to be much less critical in the optimization procedure.

We list below some future works:

- Intuitively, the same optimization procedure proposed in [93] without integrating $\boldsymbol{\gamma}$ should work here, but apparently the nonlinearity is a complicating factor. Better understanding why this procedure fails is necessary.

- The Laplace prior is not the only one suitable to enforcing sparsity, and other possibilities should be tested. In [97] some possible sparsity inducing priors are presented, maybe more adequate to our case than the Laplace prior.

- As can be seen in Equation 11.14, parameter λ is responsible for controlling the sparsity of the restored signal in the DCT domain. It can also be noted that this parameter is constant for all the DCT components. However, when dealing with audio signals, one expects that low frequencies are more significant than high frequencies. This information can be incorporated into the model in order to improve the estimation procedure, e.g. by using a value of λ which depends on the frequency bin.

- The sparsity context is very good to treat noise, since noise also reduces the sparsity of the original signal in the DCT domain. In [93], the measurements were considered noisy, but in order to simplify the adaptation, here we assumed them to be noiseless. Incorporating noise into the model is also a future work.

- In the present implementation, quantities ξ_i are computed in a `for` loop at each iteration of the optimization algorithm, which slows down the algorithm. Increasing the computational efficiency is a future work.

- Up to this point we are profiting only from the estimated maximum of the posterior distribution. This distribution contains a lot of information that is being ignored, for example, a credence interval for both λ and the polynomial coefficients. Better exploring the full potential of the Bayesian approach is a future work.

- Finally, incorporating memory into the model is our major goal, since they allow for a more realistic approximation to actual nonlinearities found in audio signals.

Part IV

RESTORATION OF AUDIO SIGNALS DEGRADED WITH LOW FREQUENCY DECAYING PULSES

Chapter 12

First attempt: a parametric description for the pulse

In this fourth part of the thesis we outdistance the restoration of audio signals with nonlinear distortions to treat other defects. A very annoying degradation present in some analogical media are the long pulses with significant low-frequency content. This kind of defect is caused by deep scratches or breakages in vinyl and gramophone disks or cylinder recordings, or damage in optical films soundtracks, among others. It has the usual form of a high-amplitude discontinuity in the waveform followed by a pulse of decaying frequency and amplitude. Physically, this can be understood as the response of the playback device to the physical degradation in the medium. Since this degradation is very severe, the device is removed from its linear range of operation and presents this atypical behavior.

This topic is the first one we worked, continuing the work initiated in [98] and having resulted in conference paper [5], in which this chapter is based. The contributions of this thesis to the topic were some modifications onto the proposal distributions for some parameters, a more efficient implementation and tests with real signals, both naturally and artificially distorted.

The chapter is organized as follows: In Section 12.1 some previous works are presented, followed by the description of our proposed model in Section 12.2. The algorithm to remove the degradation is described in Section 12.3 and some results are presented in Section 12.4.

12.1 Previous works

The first algorithm to treat this defect was proposed in [99, 100], and is based on the hypothesis of the similarity among the pulses present in a signal: since one of its possible causes are big scratches on the surface of the media, every time the needle

passes through them, similar disturbances will occur, differing only in its location and amplitude. These two quantities are estimated by comparing the degraded signal with a database of pulses. The authors reported good results when this hypothesis is valid, but its scope is limited to pulses similar to the ones present in the database. Moreover, if two or more pulses are superimposed, the method fails.

A statistical approach capable of treating more general cases can be found in [37, 101]. This method assumes that both the underlying signal and the pulse are modeled by AR processes that are superimposed. The original signal is then estimated by separating both processes. Some disadvantages are requiring that the location of the pulse is known and that the AR model for the pulse is not very accurate.

In [102] a much simpler method is proposed, based on a nonlinear filtering technique called Two-Pass Split Window (TPSW). This filtering is used to obtain a rough estimate of the pulse format, being then smoothed by a piecewise polynomial fitting. Although this method requires less computational power, the location of the pulse must still be known. And finally, in [103] is proposed a solution using the Empirical Mode Decomposition (EMD), a way of decomposing a signal waveform into a set of simpler functions, called the Intrinsic Mode Functions.

In this work we propose a method that jointly localizes and estimates the shape of the pulse. It also does not require any previous knowledge about the parameters of the AR process that models the underlying signal. The price we must pay is an increase in the required computational power. We describe now the model for the pulse and its relation with the underlying signal.

12.2 A model for the long pulse

The proposed model for the degradation describes it as an initial discontinuity, followed by damped oscillations and of decaying frequency, which we will call *tail*. The initial discontinuity is modeled as white noise, beginning at time sample n_0 of the original signal and lasting for M time samples, with variance σ_d^2 fixed:

$$\mathbf{v}_d(n) = r(n)[u(n-n_0) - u(n-n_0-M)], \tag{12.1}$$

where $u(n)$ is the unit step, $r(n) \sim N(0, \sigma_d^2)$ and n_0, M and σ_d^2 are unknown a priori.

The model for the tail is based on [102], and is mathematically described by

$$\mathbf{v}_t(n) = V_t e^{-n/(f_s \tau_e)} \sin\left(2\pi n \frac{f_n}{f_s} + \phi\right) [u(n-n_0-M-1)], \tag{12.2}$$

where

$$f_n = (f_{\max} - f_{\min})e^{-n/(f_s\tau_f)} + f_{\min}. \tag{12.3}$$

Variables n_0 and M are the same as before, and the new variables are defined below:

- V_t is related to the tail amplitude;
- f_s is the signal sampling rate (usually 44.1 kHz);
- τ_e is the time constant (in seconds) associated with the pulse envelope decay;
- τ_f is the time constant (in seconds) associated with the pulse frequency decay;
- $f_{\max}$ and $f_{\min}$ are, respectively, the maximum and minimum tail oscillation frequencies (in Hz);
- ϕ is the pulse initial phase.

All these quantities (except for f_s) are also unknown beforehand. Our goal is then to estimate $n_0, M, \sigma_d^2, V_t, \tau_e, \tau_f, f_{\max}, f_{\min}$, and ϕ in order to recover the original signal.

An important point that must be discussed is the estimation of the underlying AR model's parameters: ideally they should also be estimated together with the other unknowns, but this implies an enormous increase in the required computational time. The estimation of $\mathbf{a}$ and σ_e^2 by themselves is not costly, but there are several quantities along the process that depend on them. If those two parameters are not constant over the entire procedure, such quantities must be computed several times. To overcome this, we will analyze a block of signal that contains not only the initial discontinuity and the tail, but also a region preceding the initial estimate of the beginning of the degradation. The parameters of the AR model will then be estimated from this region before the algorithm initialization. Note, however, that this does not require the exact location of the degradation beginning.

An important feature of this model for the pulse tail is that it is deterministic: if its parameters are known, then the pulse format if uniquely determined. This property will become important soon, when computing some conditional distributions.

To simplify the notation, we assemble the parameters of the initial discontinuity, the tail and the AR model for the original signal in vectors $\boldsymbol{\theta}_d$, $\boldsymbol{\theta}_t$, and $\boldsymbol{\theta}_x$, respectively:

$$\boldsymbol{\theta}_d = [n_0 \quad M \quad \sigma_d^2]^T, \tag{12.4}$$

$$\boldsymbol{\theta}_t = [V_t \quad \tau_e \quad \tau_f \quad f_{\max} \quad f_{\min} \quad \phi]^T, \tag{12.5}$$

$$\boldsymbol{\theta}_x = [\mathbf{a}^T \quad \sigma_e^2]^T. \tag{12.6}$$

The concatenation of these three vectors is denoted by $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_d^T \quad \boldsymbol{\theta}_t^T \quad \boldsymbol{\theta}_x^T]^T. \tag{12.7}$$

We will denote the original and corrupted signal blocks as $\mathbf{x}$ and $\mathbf{y}$, respectively, and their length as N. To describe the relationship between these vectors, we create three sets of indexes, $\mathbf{i}_0$, $\mathbf{i}_1$ and $\mathbf{i}_2$, indicating the time samples in $\mathbf{y}$ that belong to the regions preceding the degradation, to the initial discontinuity of the pulse and to its tail, respectively. We define then sub-vectors $\mathbf{x}_0, \mathbf{y}_0, \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2$, and $\mathbf{y}_2$ containing the corresponding time samples in each set of indexes and satisfying

$$\mathbf{y}_0 = \mathbf{x}_0, \tag{12.8}$$

$$\mathbf{y}_1 = \mathbf{x}_1 + \mathbf{v}_d, \tag{12.9}$$

$$\mathbf{y}_2 = \mathbf{x}_2 + \mathbf{v}_t. \tag{12.10}$$

We can also define the auxiliary matrices $\mathbf{K}$, $\mathbf{U}_1$, and $\mathbf{U}_2$, containing the columns of the $N \times N$ identity matrix with indexes $\mathbf{i}_0$, $\mathbf{i}_1$, and $\mathbf{i}_2$, respectively. We can then write

$$\mathbf{x} = \mathbf{K}\mathbf{x}_0 + \mathbf{U}_1\mathbf{x}_1 + \mathbf{U}_2\mathbf{x}_2. \tag{12.11}$$

Matrix $\mathbf{A}$ defined in Chapter 4 containing the coefficients of the AR model can be partitioned in a similar way as $\mathbf{A}_0 = \mathbf{A}\mathbf{K}$, $\mathbf{A}_1 = \mathbf{A}\mathbf{U}_1$, and $\mathbf{A}_2 = \mathbf{A}\mathbf{U}_2$. These partitioning will be useful for some computations in the next sections.

12.3 Description of the algorithm

As usual, we wish to recover the original underlying signal $\mathbf{x}$ from the observed degraded signal $\mathbf{y}$ and their mutual relationship, described by the pulse model presented in the last section. The assumption that $\mathbf{x}$ is well described by an AR model allows us to write an statistical description for it, and consequently, for $\mathbf{y}$, $\boldsymbol{\theta}_t$, and $\boldsymbol{\theta}_d$. By using the Bayes' Theorem, we can compute the posterior distribution for the desired quantities, $p(\boldsymbol{\theta}_t, \boldsymbol{\theta}_d, \mathbf{x}|\boldsymbol{\theta}_x, \mathbf{y})$, the maximum of which we want to find. Unfortunately, the computation of this distribution leads to a very complicated formula, from which it is hard to extract useful information. To circumvent this issue, we obtain samples of it, by means of a Gibbs sampler with some Metropolis steps, implemented in the following way:

a) Initialize values $\boldsymbol{\theta}_d^{(0)}$ and $\boldsymbol{\theta}_t^{(0)}$

b) For k from 1 to N_{iter}:

i) Sample $\boldsymbol{\theta}_t^{(k)}$ and $\mathbf{x}^{(k)}$ from distribution

$$p(\boldsymbol{\theta}_t, \mathbf{x}|\boldsymbol{\theta}_d^{(k-1)}, \boldsymbol{\theta}_x, \mathbf{y})$$

ii) Sample $n_0^{(k)}$ and $M^{(k)}$ from distribution

$$p(n_0, M|\sigma_d^{2^{(k-1)}}, \boldsymbol{\theta}_t^{(k)}, \mathbf{x}^{(k)}, \boldsymbol{\theta}_x, \mathbf{y})$$

iii) Sample $\sigma_d^{2^{(k)}}$ from distribution

$$p(\sigma_d^2|n_0^{(k)}, M^{(k)}, \boldsymbol{\theta}_t^{(k)}, \mathbf{x}^{(k)}, \boldsymbol{\theta}_x, \mathbf{y}).$$

The mean of the posterior distribution is then estimated by averaging the samples obtained after burn-in time (see Chapter 3). In the next sections we compute the required conditional distributions.

12.3.1 Sampling from $p(\boldsymbol{\theta}_t, \mathbf{x}|\boldsymbol{\theta}_d, \boldsymbol{\theta}_x, \mathbf{y})$

The first step of the Gibbs sampler requires this distribution, which we will compute now. We use Bayes' Theorem to split it in more easily obtainable distributions:

$$\begin{aligned} p(\boldsymbol{\theta}_t, \mathbf{x}|\boldsymbol{\theta}_d, \boldsymbol{\theta}_x, \mathbf{y}) &= p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_d, \boldsymbol{\theta}_x, \mathbf{y}) p(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y}) \\ &\propto [p(\mathbf{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta}_t)] p(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y}). \end{aligned} \tag{12.12}$$

We have now two distributions to compute: $p(\mathbf{y}|\boldsymbol{\theta})$ and $p(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$. We will separate their computation in different subsections to keep the text organized.

12.3.1.1 Computation of $p(\mathbf{y}|\boldsymbol{\theta})$

The computation of this quantity is quite complicated, and will be split in several steps, to improve the readability of the text. Two auxiliary results are stated here, and will be referenced when necessary. Most of the auxiliary computations are omitted from the text in order to improve readability.

12.3.1.1.1 Gaussian integrals

Integrals of the form

$$I = \int_{\mathbb{R}^D} \exp\left\{-\frac{1}{2}(a + \mathbf{b}^T\mathbf{y} + \mathbf{y}^T\mathbf{C}\mathbf{y})\right\} \, d\mathbf{y} \tag{12.13}$$

are quite recurrent in Statistics, and in this chapter we will come across one. By means of a very boring computation (essentially completing the squares and com-

paring with the PDF of a multivariate Gaussian distribution) we can prove that

$$I = \frac{(2\pi)^{D/2}}{\det(\mathbf{C})^{1/2}} \exp\left\{-\frac{1}{2}\left(a - \frac{\mathbf{b}^T\mathbf{C}^{-1}\mathbf{b}}{4}\right)\right\}. \tag{12.14}$$

12.3.1.1.2 Product of multivariate Gaussians PDFs

If we consider a multidimensional random variable whose PDF is given by the product of two multivariate Gaussian PDFs, we can conclude that it is also a multivariate Gaussian. More precisely, let

$$f(\mathbf{x}) \propto f_1(\mathbf{x}) f_2(\mathbf{x}), \tag{12.15}$$

where $f_1(\mathbf{x}) = N(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $f_2(\mathbf{x}) = N(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$. Therefore,

$$f(\mathbf{x}) \propto \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right\}, \tag{12.16}$$

and the expression inside the exponential is quadratic in $\mathbf{x}$. We can then complete the squares in order to write it in the form

$$f(\mathbf{x}) \propto \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}, \tag{12.17}$$

but this procedure leads to a very boring computation. We obtain parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in another way: after deriving and equating to zero the argument of the exponential in Equation 12.16 we conclude that $\boldsymbol{\mu} = (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2)$, and by taking the negative Hessian of this same quantity we arrive at $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}$. Therefore, $f(\mathbf{x}) = N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

12.3.1.2 Back to the computation of $p(\mathbf{y}|\boldsymbol{\theta})$

We then rewrite $p(\mathbf{y}|\boldsymbol{\theta})$ as

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int_{\mathbb{R}^N} p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})\ d\mathbf{x} = \int_{\mathbb{R}^N} p(\mathbf{x}|\boldsymbol{\theta}) p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})\ d\mathbf{x}. \tag{12.18}$$

Note that $\mathbf{x}$ does not depend on $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_t$, since both sets of parameters describe only the degradation, and not the underlying original signal. Therefore, we can substitute $p(\mathbf{x}|\boldsymbol{\theta})$ by $p(\mathbf{x}|\boldsymbol{\theta}_x)$, which by Equations 4.12 and 4.14 is approximately Gaussian with mean $\mathbf{0}$ and covariance matrix $\sigma_e^2(\mathbf{A}^T\mathbf{A})^{-1}$. Once again we will abuse the approximation in Equation 4.14 and consider it as an equality, implying that

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{x}|\boldsymbol{\theta}_x) = N(\mathbf{x}|\mathbf{0}, \sigma_e^2(\mathbf{A}^T\mathbf{A})^{-1}). \tag{12.19}$$

Note that there is another abuse of notation here, since we are not allowed to write that a PDF is equal to a distribution. Since the notation and the computations in this chapter are sufficiently complicated, small abuses like this will be recurrent and must cause no confusion to the reader.

To compute $p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})$, note first that the time samples in $\mathbf{y}_0$ are uncorrupted and the time samples in $\mathbf{y}_2$ are uniquely determined by the parameters of the tail in $\boldsymbol{\theta}_t$. Therefore, their distribution can be modeled by adequate multi-dimensional Dirac's[1] delta distributions. And since the time samples in $\mathbf{y}_1$ are the time samples in $\mathbf{x}_1$ plus random Gaussian noise, we can write

$$p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) = \delta(\mathbf{y}_0 - \mathbf{x}_0) \times N(\mathbf{y}_1|\mathbf{x}_1, \sigma_d^2\mathbf{I}_M) \times \delta(\mathbf{y}_2 - (\mathbf{x}_2 + \mathbf{v}_t)), \tag{12.20}$$

since this distribution is conditioned on $\boldsymbol{\theta}$ and $\mathbf{x}$.

Therefore, the product $p(\mathbf{x}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})$ is given by

$$N(\mathbf{x}|\mathbf{0}, \sigma_e^2(\mathbf{A}^T\mathbf{A})^{-1}) \times [\delta(\mathbf{y}_0 - \mathbf{x}_0) \times N(\mathbf{y}_1|\mathbf{x}_1, \sigma_d^2\mathbf{I}_M) \times \delta(\mathbf{y}_2 - (\mathbf{x}_2 + \mathbf{v}_t))]. \tag{12.21}$$

Note that the integral to be computed in Equation 12.18 is with respect to $\mathbf{x}$, and the second Gaussian in the last Equation is in $\mathbf{y}_1$ variable, depending on $\mathbf{x}_1$ only in its mean. To make explicit the dependence on $\mathbf{x}_1$, we can use the symmetry of the Gaussian distribution and consider that $N(\mathbf{y}_1|\mathbf{x}_1, \sigma_d^2\mathbf{I}_M) = N(\mathbf{x}_1|\mathbf{y}_1, \sigma_d^2\mathbf{I}_M)$, in the sense that the PDF of both distributions is given by the same formula. By using the decomposition of $\mathbf{x}$ given in Equation 12.11, we obtain:

$$\begin{aligned} p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta}) &= p(\mathbf{x}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) = \\ &N(\mathbf{K}\mathbf{x}_0 + \mathbf{U}_1\mathbf{x}_1 + \mathbf{U}_2\mathbf{x}_2|\mathbf{0}, \sigma_e^2(\mathbf{A}^T\mathbf{A})^{-1}) \times \\ &\quad [\delta(\mathbf{y}_0 - \mathbf{x}_0) \times N(\mathbf{x}_1|\mathbf{y}_1, \sigma_d^2\mathbf{I}_M) \times \delta(\mathbf{y}_2 - (\mathbf{x}_2 + \mathbf{v}_t))], \end{aligned} \tag{12.22}$$

and finally,

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\theta}) = &\int_{\mathbb{R}^N} p(\mathbf{x}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})\ d\mathbf{x} = \\ &\int_{\mathbb{R}^N} N(\mathbf{K}\mathbf{x}_0 + \mathbf{U}_1\mathbf{x}_1 + \mathbf{U}_2\mathbf{x}_2|\mathbf{0}, \sigma_e^2(\mathbf{A}^T\mathbf{A})^{-1}) \times \\ &\quad [\delta(\mathbf{y}_0 - \mathbf{x}_0) \times N(\mathbf{x}_1|\mathbf{y}_1, \sigma_d^2\mathbf{I}_M) \times \delta(\mathbf{y}_2 - (\mathbf{x}_2 + \mathbf{v}_t))]\ d\mathbf{x} = \\ &\int_{\mathbb{R}^M} N(\mathbf{K}\mathbf{x}_0 + \mathbf{U}_1\mathbf{x}_1 + \mathbf{U}_2\mathbf{x}_2|\mathbf{0}, \sigma_e^2(\mathbf{A}^T\mathbf{A})^{-1})|_{\mathbf{x}_0=\mathbf{y}_0 \text{ and } \mathbf{x}_2=\mathbf{y}_2-\mathbf{v}_t} \times \\ &\quad N(\mathbf{x}_1|\mathbf{y}_1, \sigma_d^2\mathbf{I}_M)\ d\mathbf{x}_1, \end{aligned} \tag{12.23}$$

[1] Paul Adrien Maurice Dirac, Bristol, August 8, 1902 – Tallahassee, October 20, 1984.

where in the last equality we computed the integral over $\mathbf{x}_0$ and $\mathbf{x}_2$, and the unusual expression $N(\mathbf{K}\mathbf{x}_0+\mathbf{U}_1\mathbf{x}_1+\mathbf{U}_2\mathbf{x}_2|\mathbf{0},\sigma_e^2(\mathbf{A}^T\mathbf{A})^{-1})|_{\mathbf{x}_0=\mathbf{y}_0 \text{ and } \mathbf{x}_2=\mathbf{y}_2-\mathbf{v}_t}$ denotes the conditional distribution when $\mathbf{x}_0=\mathbf{y}_0$ and $\mathbf{x}_2=\mathbf{y}_2-\mathbf{v}_t$, which depends only on $\mathbf{x}_1$. To compute this integral we substitute the formulas for the respective Gaussian PDFs and use the results in Section 12.3.1.1.1, with $\mathbf{y}=\mathbf{x}_1$. After a very tedious but straightforward computation, we obtain that

$$p(\mathbf{y}|\boldsymbol{\theta}) = \frac{\lambda^M}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}\det(\boldsymbol{\Phi})}\exp\left\{-\frac{1}{2\sigma_e^2}E_{\min}\right\}, \tag{12.24}$$

where

$$E_{\min} = \lambda^2\mathbf{y}_1^T\mathbf{y}_1 + \mathbf{z}^T\begin{bmatrix}\mathbf{A}_0^T\\ \mathbf{A}_2^T\end{bmatrix}[\mathbf{A}_0\ \mathbf{A}_2]\mathbf{z} + (\mathbf{x}_1^{\text{MAP}})^T\boldsymbol{\Theta}, \tag{12.25}$$

$$\mathbf{z} = \begin{bmatrix}\mathbf{y}_0\\ \mathbf{y}_2-\mathbf{v}_t\end{bmatrix}, \tag{12.26}$$

$$\mathbf{x}_1^{\text{MAP}} = \boldsymbol{\Phi}^{-1}\boldsymbol{\Theta}, \tag{12.27}$$

$$\boldsymbol{\Phi} = \lambda\mathbf{I}_M + \mathbf{A}_1^T\mathbf{A}_1, \tag{12.28}$$

$$\boldsymbol{\Theta} = \lambda\mathbf{y}_1 - \mathbf{A}_1^T[\mathbf{A}_0\ \mathbf{A}_2]\mathbf{z}. \tag{12.29}$$

12.3.1.3 Computation of $p(\mathbf{x}|\boldsymbol{\theta},\mathbf{y})$

This computation is much easier, and was performed indirectly in the last section. By using Bayes' Theorem, we have

$$p(\mathbf{x}|\boldsymbol{\theta},\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta}), \tag{12.30}$$

which is exactly the distribution computed in Equation 12.22, that is,

$$\begin{aligned} p(\mathbf{x}|\boldsymbol{\theta},\mathbf{y}) = {} & N(\mathbf{K}\mathbf{x}_0+\mathbf{U}_1\mathbf{x}_1+\mathbf{U}_2\mathbf{x}_2|\mathbf{0},\sigma_e^2(\mathbf{A}^T\mathbf{A})^{-1})\times \\ & [\delta(\mathbf{y}_0-\mathbf{x}_0)\times N(\mathbf{x}_1|\mathbf{y}_1,\sigma_d^2\mathbf{I}_M)\times\delta(\mathbf{y}_2-(\mathbf{x}_2+\mathbf{v}_t))]. \end{aligned} \tag{12.31}$$

This distribution can be further simplified by noting that the first Gaussian depends essentially only on $\mathbf{x}_1$, since its dependence on $\mathbf{x}_0$ and $\mathbf{x}_2$ are restricted by the two Dirac's deltas. As stated in Section 12.3.1.1.2, the product of Gaussians is also Gaussian, and after performing the computations to find its mean and covariance matrix, we obtain

$$p(\mathbf{x}|\boldsymbol{\theta},\mathbf{y}) = \delta(\mathbf{y}_0-\mathbf{x}_0)\times N(\mathbf{x}_1|\mathbf{x}_1^{\text{MAP}},\sigma_e^2\boldsymbol{\Phi}^{-1})\times\delta(\mathbf{y}_2-(\mathbf{x}_2+\mathbf{v}_t)). \tag{12.32}$$

12.3.1.4 Prior distribution for $\boldsymbol{\theta}_t$

As stated in Equation 12.12, the posterior distribution of $\boldsymbol{\theta}_t$ is given by $p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}_t)$, where $p(\mathbf{y}|\boldsymbol{\theta})$ is given in Equation 12.24. Note that the dependence in $\boldsymbol{\theta}_t$ is very complicated, and this fact implies that no choice of prior distribution for $\boldsymbol{\theta}_t$ can make this distribution easier to sample from, suggesting that a Metropolis step within the Gibbs sampler can be used to sample from this distribution. A choice of prior distribution that is a good compromise between simplicity and accuracy is broad Gaussian for variables that can assume any real value, and Inverse Gamma with small parameters for the remaining ones.

12.3.1.5 Proposal distributions to sample $\boldsymbol{\theta}_t$ from

As suggested in [30], a Gaussian distribution centered at the previous value is, in most cases, a good proposal distribution, and is the one adopted here. However, since the posterior distribution for $\boldsymbol{\theta}_t$ is quite complicated, we have no intuition about the degree of dependence between its components, a fact that prevents us from using a covariance matrix that suitably expresses this dependence. Furthermore, we will not be able to independently control the acceptance rate of a single parameter at a time, and if a particular variable presents convergence problems, this behavior can slow down the overall convergence of the algorithm. Therefore, a diagonal covariance matrix seems to be a good choice, since in this case each component of $\boldsymbol{\theta}_t$ is sampled independently. The variance for each variable is chosen to keep its acceptance rate around 50%, as suggested by some authors to guarantee that the sample space of the sampled variable is well explored in a reasonable computational time [30].

12.3.1.6 A further simplification

In order to decide between accepting or rejecting any sampled value for any component of $\boldsymbol{\theta}_t$, the acceptance ratio for the Metropolis-Hastings algorithm (Equation 3.3) must be computed, implying that the complicated Equation 12.24 must be calculated twice. Since it requires the inversion of a matrix and some matrix multiplications, this task is very computationally intensive. However, a simplification in Equation 12.24 can be done: note that λ is likely to be very small, since σ_d^2 is usually several orders of magnitude greater than σ_e^2. In the argument of the exponential in Equation 12.24 this quantity multiplies $\mathbf{y}_1$, whose entries do not typically exceed $3\sigma_d$ beyond the underlying signal, due to their Gaussian distribution. We then ignore all terms inside the exponential involving λ in Equation 12.24, which

becomes proportional to $\exp(-\frac{1}{2}\mathbf{z}^T\mathbf{R}\mathbf{z})$, where

$$\mathbf{R} = \frac{1}{\sigma_e^2}\begin{bmatrix}\mathbf{A}_0^T \\ \mathbf{A}_2^T\end{bmatrix}\mathbf{S}[\mathbf{A}_0\ \mathbf{A}_2], \tag{12.33}$$

with

$$\mathbf{S} = \mathbf{I}_{N-P} - \mathbf{A}_1(\mathbf{A}^{-1}\mathbf{A})^{-1}\mathbf{A}_1^T. \tag{12.34}$$

Both matrices $\mathbf{R}$ and $\mathbf{S}$ were obtained by simply making $\lambda = 0$ in Equation 12.24 and related quantities.

By making this assumption, we are supposing that $\mathbf{y}_1$ contains no useful information to compute $p(\mathbf{y}|\boldsymbol{\theta})$. Note that $\mathbf{R}$ does not depend on $\boldsymbol{\theta}_t$, and thus can be computed only once per iteration of the Gibbs sampler.

With this simplification, the distribution of V_t is easily described, and thus can be directly sampled from. By examining Equation 12.24, with the hypothesis of $\lambda = 0$, as a function of V_t only, we identify a Gaussian distribution with mean

$$\mu_{V_t} = \frac{[\mathbf{y}_0\ \mathbf{y}_2]\begin{bmatrix}\mathbf{A}_0 \\ \mathbf{A}_2\end{bmatrix}\mathbf{S}\mathbf{A}_2\mathbf{p}}{\mathbf{p}^T(\mathbf{A}_2^T\mathbf{S}\mathbf{A}_2)\mathbf{p}} \tag{12.35}$$

and variance

$$\sigma_{V_t}^2 = \frac{1}{\mathbf{p}^T(\mathbf{A}_2^T\mathbf{S}\mathbf{A}_2)\mathbf{p}}, \tag{12.36}$$

where vector $\mathbf{p}$ satisfies $V_t\mathbf{p} = \mathbf{v}_t$.

12.3.1.7 Conclusion

Now, the first step of the Gibbs sampler is completely described. Sampling from the posterior distribution of $\boldsymbol{\theta}_t$ is performed by the Metropolis steps discussed above, where the acceptance rate is more easily computed with the simplification of $\lambda = 0$ in Equation 12.24. One exception is variable V_t, whose distribution in this simplified scenario is Gaussian, as explained above.

After a sample of $\boldsymbol{\theta}_t$ is obtained, we must sample from $p(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$, a distribution that is easy to sample from, as seen in Equation 12.32.

12.3.2 Sampling from $p(n_0, M|\sigma_d^2, \boldsymbol{\theta}_t, \mathbf{x}, \boldsymbol{\theta}_x, \mathbf{y})$

This distribution is much easier to sample from than that previously discussed. By using Bayes' Theorem and supposing independence a priori between n_0 and M, we have

$$p(n_0, M|\sigma_d^2, \boldsymbol{\theta}_t, \mathbf{x}, \boldsymbol{\theta}_x, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(n_0)p(M), \tag{12.37}$$

where

$$p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{x}) = \delta(\mathbf{y}_0 - \mathbf{x}_0) \times N(\mathbf{y}_1|\mathbf{x}_1, \sigma_d^2 \mathbf{I}_M) \times \delta(\mathbf{y}_2 - (\mathbf{x}_2 + \mathbf{v}_t)), \tag{12.38}$$

as stated previously in Equation 12.20. However, as a function of n_0 and M, the equation above defines a very complicated distribution, which cannot be simplified with the choice of a clever prior distribution. Then, $p(n_0)$ was chosen to be uniform over an interval centered at the initial guess $n_0^{(0)}$, while $p(M)$ was chosen as a Poisson with parameter equal to the initial value $M^{(0)}$. In order to sample from the required distribution, another Metropolis step within the Gibbs sampler is employed, where the proposal distributions for n_0 and M are discrete uniform distributions over an interval centered at their respective last sampled and accepted values. The length of the intervals are chosen to keep the acceptance rate around 50%.

12.3.3 Sampling from $p(\sigma_d^2|n_0, M, \boldsymbol{\theta}_t, \mathbf{x}, \boldsymbol{\theta}_x, \mathbf{y})$

This is the last step of the Gibbs sampler. To compute the required distribution, we use the Bayes' Theorem once again and obtain:

$$p(\sigma_d^2|n_0, M, \boldsymbol{\theta}_t, \mathbf{x}, \boldsymbol{\theta}_x, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\sigma_d^2). \tag{12.39}$$

Now, the dependence of the $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ on σ_d^2 is very simple, since this is a scale parameter for the distribution. Therefore, the Inverse Gamma distribution with parameters α_d and β_d is a good choice of prior distribution, since it implies that $p(\sigma_d^2|n_0, M, \boldsymbol{\theta}_t, \mathbf{x}, \boldsymbol{\theta}_x, \mathbf{y})$ is also an Inverse Gamma with parameters given by

$$\alpha = \alpha_d + \frac{M}{2} \tag{12.40}$$

and

$$\beta = \beta_d + \frac{1}{2} \sum_{i=0}^{M-1} \mathbf{v}_d (n_0 + i)^2. \tag{12.41}$$

Hyperparameters α_d and β_d are chosen very close to zero to turn the prior vague, reflecting no previous knowledge about σ_d^2.

12.4 Results

Tests similar to those in [5] were performed, in three different scenarios: artificial and real signals degraded artificially with pulses generated using the presented model; real degraded signal obtained from a damaged cylinder recording. All signals are in PCM format, sampled at $f_s = 44.1$ kHz with 16-bit precision. The algorithm

was implemented and executed in my personal computer ASUS™[2] K45VM with processor Intel Core i7 3610QM™[3] at 2.3 GHz clock and possessing 8 GB of RAM, in MATLAB™[4] version R2014a.

12.4.1 Artificial signal degraded artificially with pulses generated using the proposed model

An artificial signal following the AR model with 5,000 time samples was generated. The filter representing the AR model has poles at frequencies $\pi/16$, $\pi/8$, and $\pi/4$ radians per time sample and their corresponding conjugate frequencies, each one with module 0.99. The variance of the excitation signal was chosen to be $\sigma_e^2 = 5 \times 10^{-7}$. The initial discontinuity of the pulse begins at time sample $n_0 = 500$ and lasts for $M = 10$ time samples, with variance given by $\sigma_d^2 = 0.5$. The parameters in $\boldsymbol{\theta}_t$ are $V_t = 0.3$, $\tau_e = 0.07$ s, $\tau_f = 0.013$ s, $f_{\max} = 60$ Hz, $f_{\min} = 20$ Hz, and $\phi = 0$ rad. These parameters, inspired by [98, 102], were chosen in order to roughly describe a typical pulse present in audio signals.

The algorithm was run for 300 iterations, with a burn-in time of 250 iterations. Therefore, the estimated parameters consist of the average of the last 50 samples. In order to better compare the original, initial and estimated values for $\boldsymbol{\theta}_t$, we assemble them in Table 12.1. Figures 12.1, 12.2, and 12.3 show the convergence of the parameters in $\boldsymbol{\theta}_d$ and first and second halves of $\boldsymbol{\theta}_t$, respectively. Green squares are real values, and red circles represent the estimated values for each parameter. Finally, in Figure 12.4 we can compare the estimated pulse (in red) with the original one (in blue). The degraded signal is also shown, in green.

Table 12.1: Artificial signal with artificial pulse: comparison of real, initial and estimated values for parameters in $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_t$.

	Real values	Initial values	Estimated values
n_0	500	485	500
M	10	15	10
σ_d^2	0.5	0.4	0.4579
V_t	0.3	0.5	0.3003
τ_e	0.07	0.09	0.0705
τ_f	0.013	0.010	0.0143
$f_{\max}$	60	55	55.641
$f_{\min}$	20	25	19.7078
ϕ	0	0.3	-0.0134

[2] ASUSTeK Computer Inc., `http://www.asus.com/`
[3] Intel Corporation, `http://www.intel.com/`
[4] The MathWorks, Inc., `http://www.mathworks.com/`

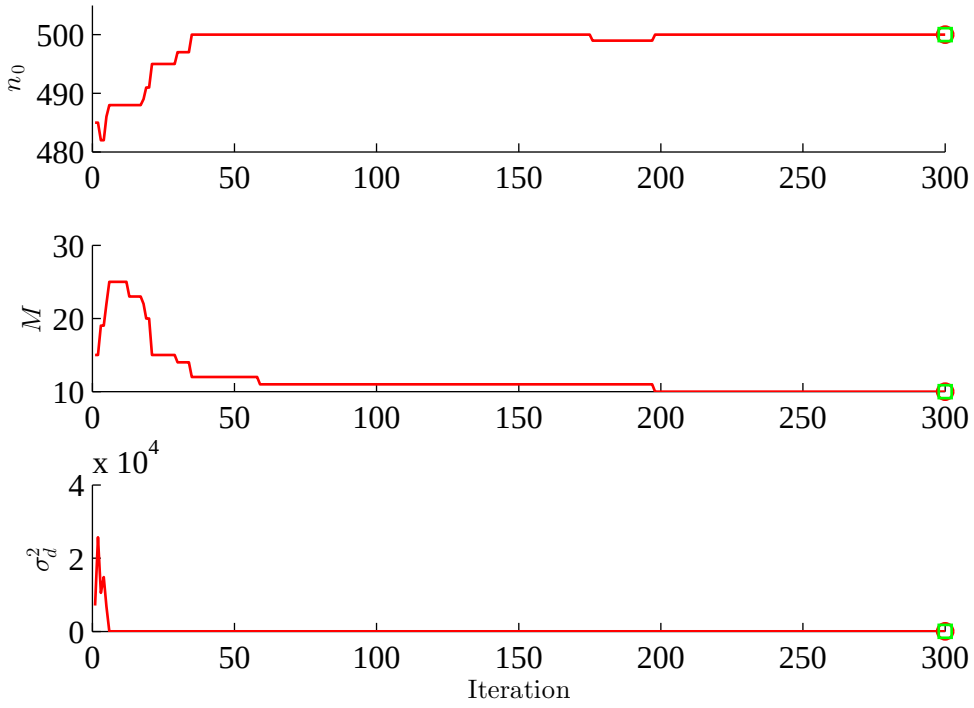

Figure 12.1: Artificial signal with artificial pulse: Convergence of $\boldsymbol{\theta}_d$.

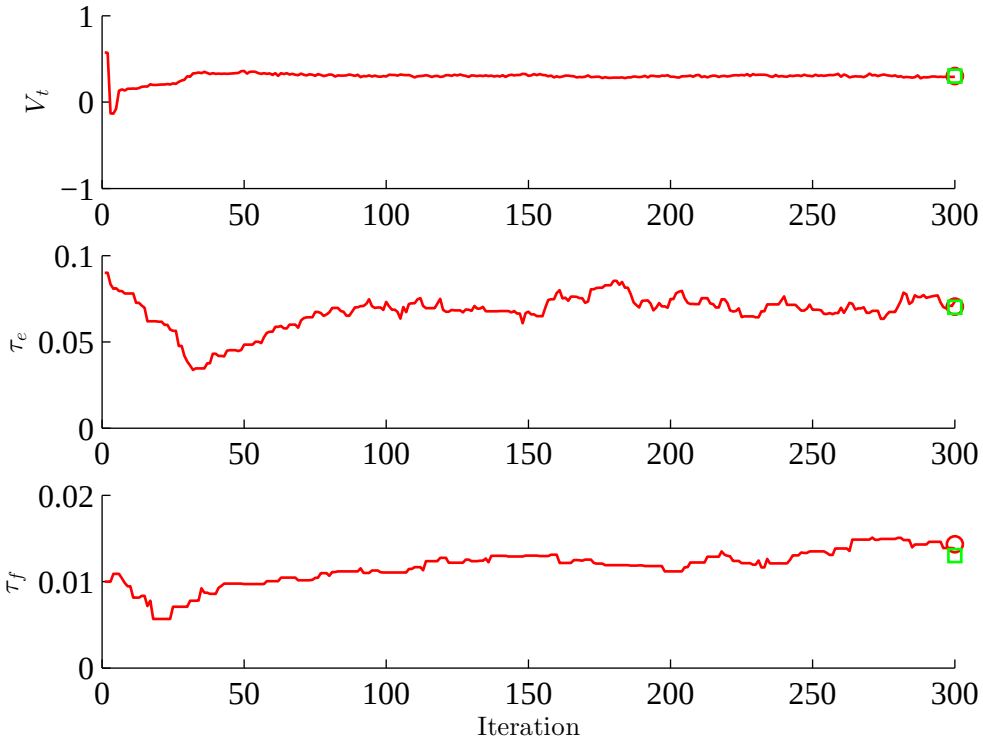

Figure 12.2: Artificial signal with artificial pulse: Convergence of V_t, τ_e, and τ_f.

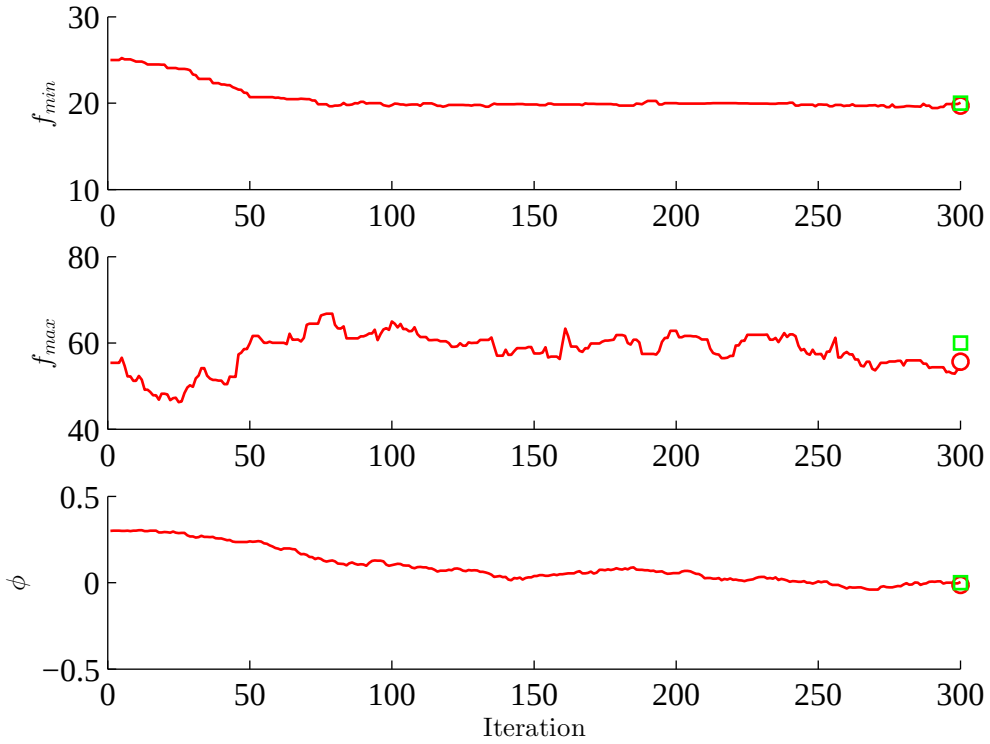

Figure 12.3: Artificial signal with artificial pulse: Convergence of $f_{\min}$, $f_{\max}$, and ϕ.

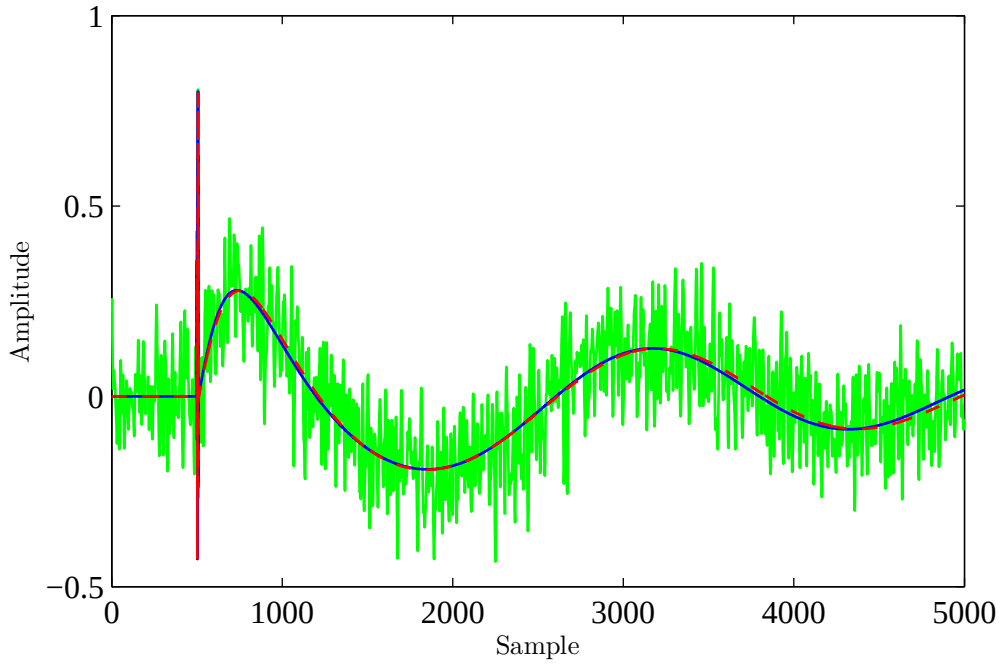

Figure 12.4: Artificial signal with artificial pulse: Comparison of the estimated pulse (red) with the original pulse (blue). The degraded signal is shown in green.

We conclude that the method is capable of identifying the correct parameters with reasonable accuracy, although the variance of proposal distributions must be tuned by hand in order to the algorithm present good convergence properties. Each iteration takes about 0.57 s.

The initial values shown in Table 12.1 were chosen by hand, and are close to the real values in order to also roughly describe a real pulse. Tests not shown here indicated that the initial values of n_0 and M are more critical to the convergence of the algorithm than the initial value of the variables in $\boldsymbol{\theta}_t$. This issue will be circumvented in Chapter 13, where an efficient initialization procedure for the n_0 and M is presented, together with a nonparametric model for the tail of the pulse.

12.4.2 Real signal degraded artificially with pulses generated using the proposed model

The chosen signal was `orchestra.wav`, a 9-s long excerpt of orchestral music with long notes being played, the same used in Chapter 9. An arbitrary excerpt of 10,000 time samples of the signal was separated and corrupted by a pulse following the presented model with the same parameters as in the previous section: the initial discontinuity of the pulse begins at time sample $n_0 = 500$ and lasts for $M = 10$ time samples, with variance given by $\sigma_d^2 = 0.5$. The parameters in $\boldsymbol{\theta}_t$ are $V_t = 0.3$, $\tau_e = 0.07$ s, $\tau_f = 0.013$ s, $f_{\max} = 60$ Hz, $f_{\min} = 20$ Hz, and $\phi = 0$ rad. These parameters were chosen by hand in order to roughly describe a typical pulse present in audio signals. The order of the AR model was $P = 40$.

The algorithm was run for 200 iterations, with a burn-in time of 150 iterations. Therefore, the estimated parameters consists of the average of the last 50 samples. Now the mean time per iteration was around 2.35 s, since the block analyzed is longer. In Table 12.2 we can compare the original, initial and estimated values for $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_t$, and Figures 12.5, 12.6, and 12.7 show the convergence of the parameters in $\boldsymbol{\theta}_d$ and first and second halves of $\boldsymbol{\theta}_t$, respectively. Green squares are real values, and red circles represent the estimated values for each parameter. Finally, in Figure 12.8 we can compare the estimated pulse (in red) with the original one (in blue). The degraded signal is also shown, in green. The initial values are the same as previously.

Table 12.2: Real signal with artificial pulse: comparison of real, initial and estimated values for parameters in $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_t$.

	Real values	Initial values	Estimated values
n_0	500	485	500
M	10	15	10
σ_d^2	0.5	0.4	0.5059
V_t	0.3	0.5	0.3062
τ_e	0.07	0.09	0.0696
τ_f	0.013	0.010	0.0126
$f_{\max}$	60	55	54.118
$f_{\min}$	20	25	19.6077
ϕ	0	0.3	0.2106

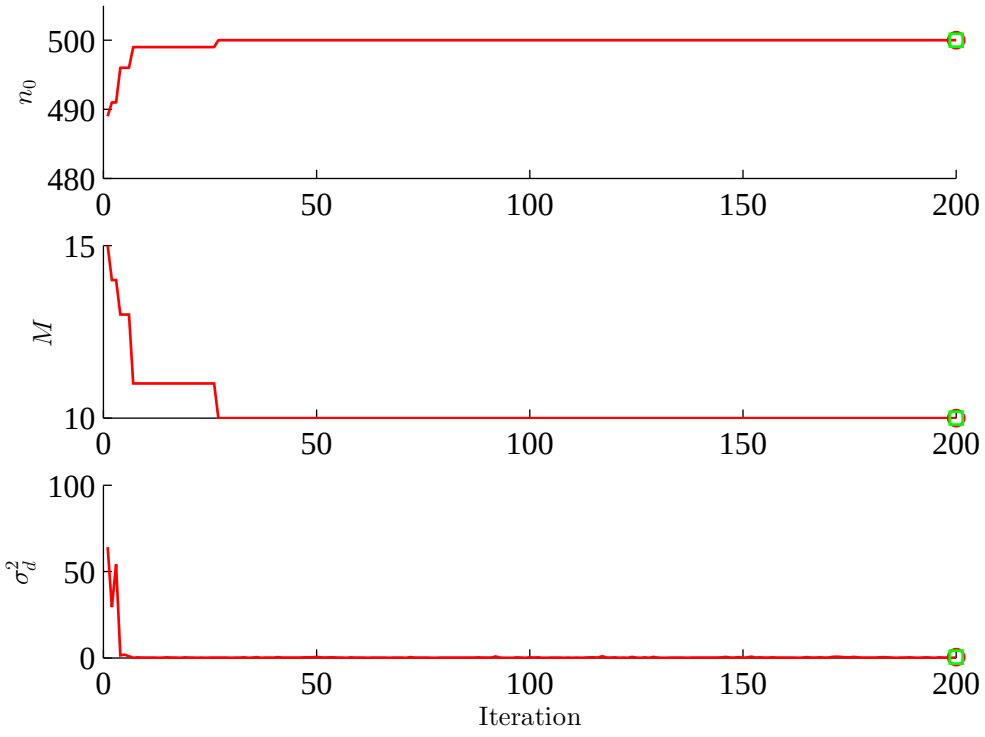

Figure 12.5: Real signal with artificial pulse: Convergence of $\boldsymbol{\theta}_d$.

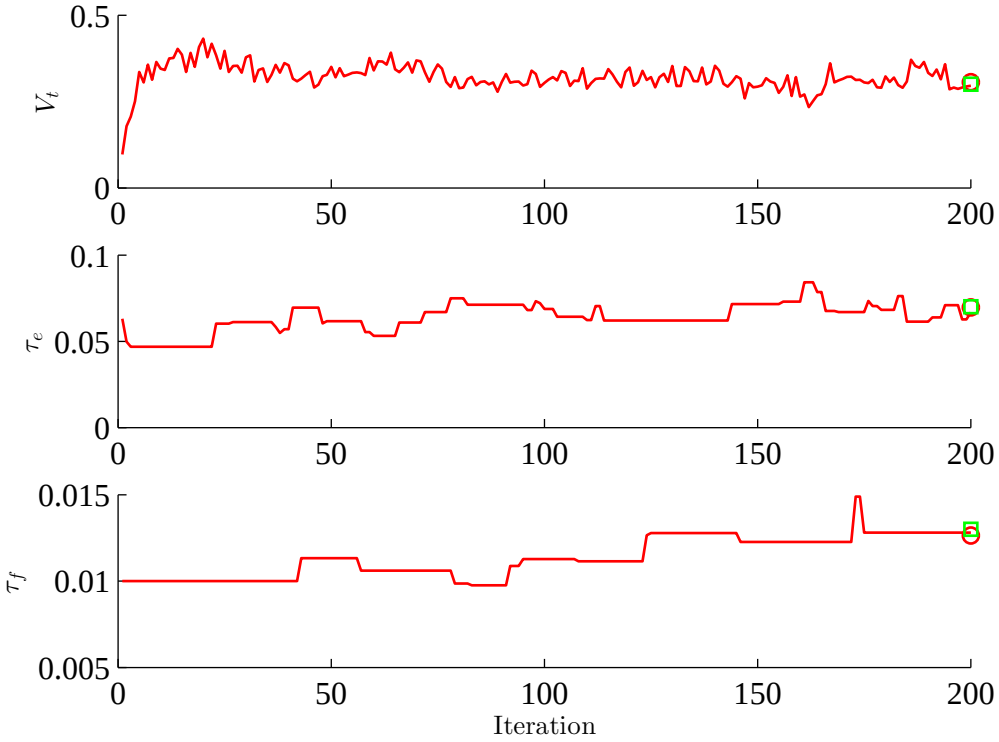

Figure 12.6: Real signal with artificial pulse: Convergence of V_t, τ_e, and τ_f.

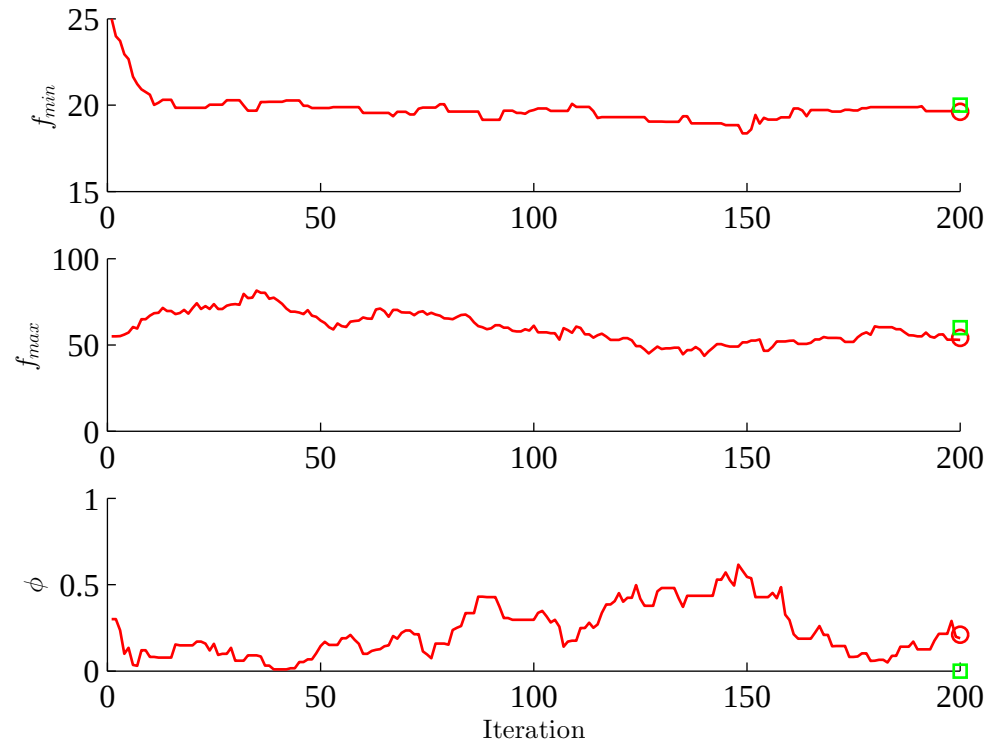

Figure 12.7: Real signal with artificial pulse: Convergence of $f_{\min}$, $f_{\max}$, and ϕ.

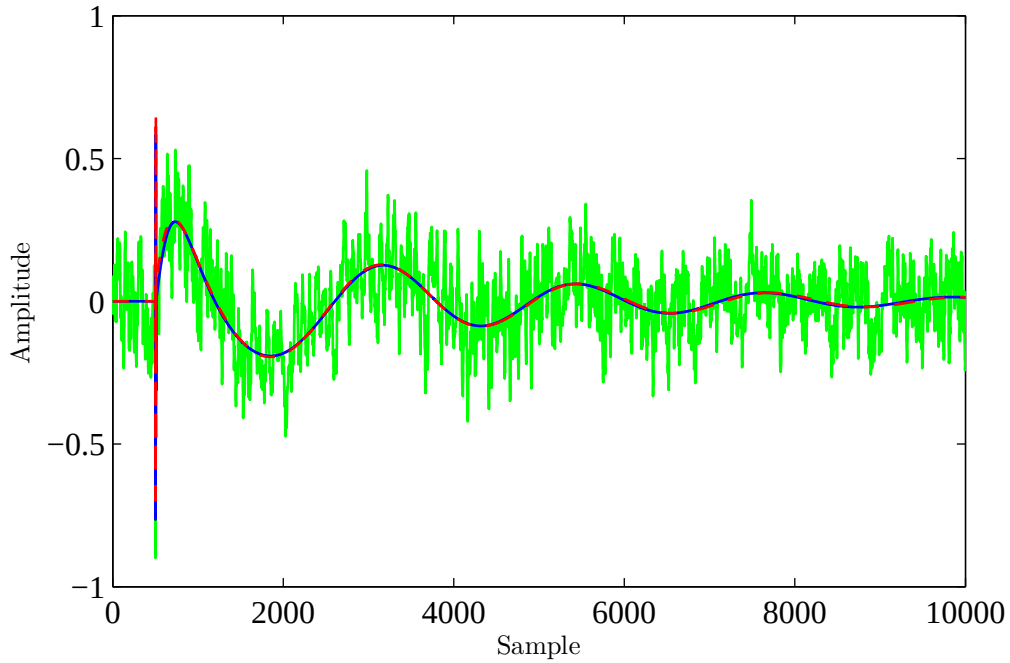

Figure 12.8: Real signal with artificial pulse: Comparison of the estimated pulse (red) with the original pulse (blue). The degraded signal is shown in green.

Note that in this more realistic scenario the convergence of the parameters (mainly the ones in $\boldsymbol{\theta}_t$) is more problematic than in the previous case. In fact, some fine tuning of the variances of the proposal distributions was necessary in order to obtain a proper exploration of the sample space, and to arrive at the desired result.

It is important to note that a single AR model of order $P = 40$ is being used for a block of 10,000 time samples, whereas as discussed in Chapter 4 it should be valid within a block of approximately 1,000 time samples. This robustness of the method is a remarkable property, and a possible reason for this fact is that the tail of the pulse varies much slower than the underlying original signal, implying that their internal correlation structures are very different, being the fine description in small scale given by the AR model not necessary anymore. Indeed, in Chapter 13 the signal located at the tail portion of the pulse is approximated by white noise (an AR model of order $P = 0$), and good results are also obtained.

12.4.3 Real degraded cylinder recording

We now test the proposed method in a real degraded audio. At the webpage [104] of the National Library of Catalonia we can find a collection of recordings from the beginning of the twentieth Century made in wax cylinders. All those recordings are severely degraded, and some of them also contain long pulses. The chosen

record here is from cylinder number 154, approximately from 1907, where the long pulse occurs right at the beginning when the music being played is presented. A transcription of this short excerpt is "*Danza número 2 de Grieg, por el eminente pianista Sr. Marshall*".

An excerpt of 7,000 time samples around the pulse was isolated, and the parameters of the AR model were estimated from the first 630 time samples. The algorithm was run for 500 iterations, with a burn-in time of 400 iterations, the required parameters being then estimated as the average of the last 100 samples. The mean time per iteration was around 1.3173 s. In Table 12.3 we can compare the initial (set by hand) and estimated values for $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_t$, and Figures 12.9, 12.10, and 12.11 show the convergence of the parameters in $\boldsymbol{\theta}_d$ and first and second halves of $\boldsymbol{\theta}_t$, respectively. Finally, in Figure 12.12 we can compare the estimated pulse (in red) with the original signal (in green).

Therefore, the proposed method also performs well in this realistic scenario, despite the fine tuning of the variances of proposal distributions being necessary in order to obtain a proper exploration of the sample space.

An improvement of the technique here presented is proposed in the next chapter, together with a list of future works.

Table 12.3: Real signal with real pulse: comparison of initial and estimated values for parameters in $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_t$.

	Initial values	Estimated values
n_0	630	642
M	350	268
σ_d^2	0.1	0.4853
V_t	0.3	0.2499
τ_e	0.07	0.0140
τ_f	0.013	0.0064
$f_{\max}$	60	132.0232
$f_{\min}$	20	29.3860
ϕ	0	-0.4316

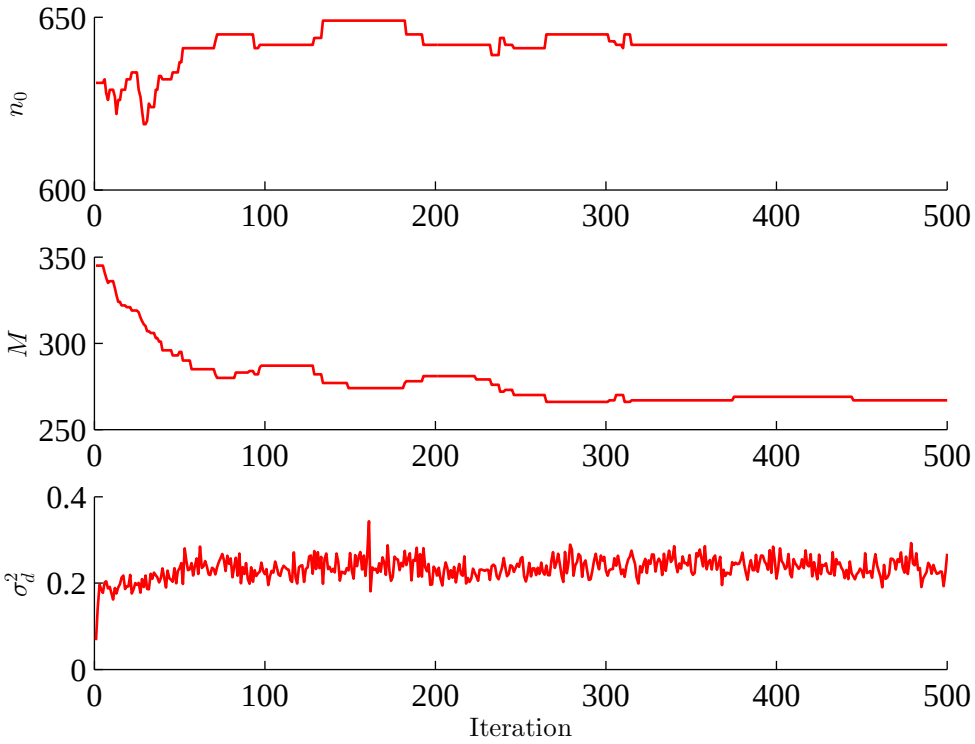

Figure 12.9: Real signal with real pulse: Convergence of $\boldsymbol{\theta}_d$.

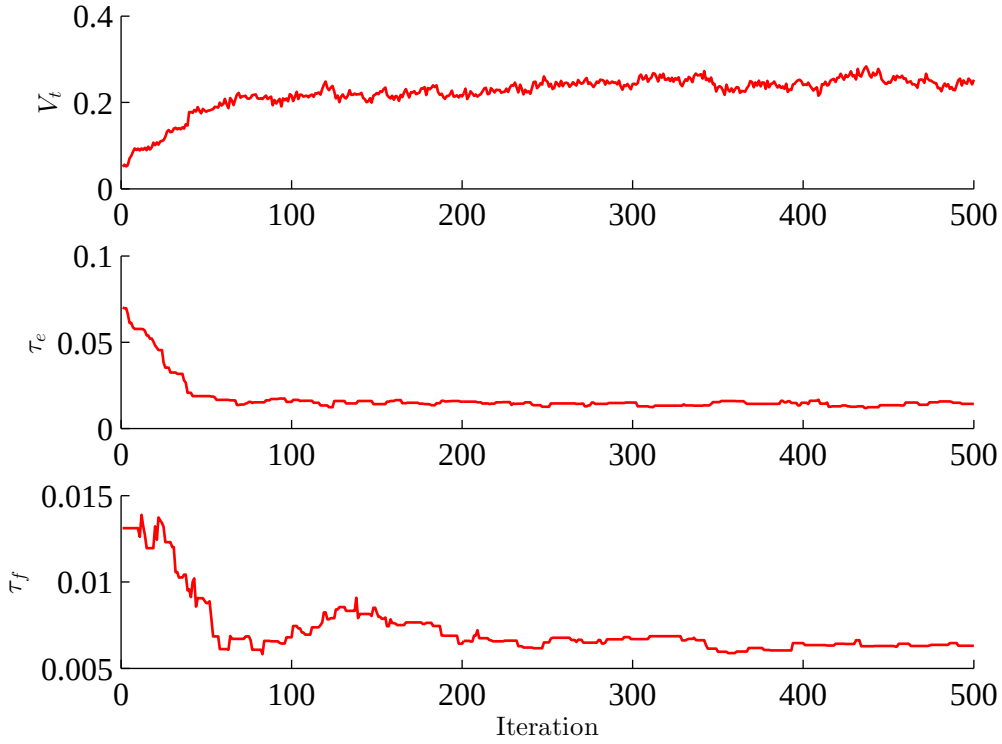

Figure 12.10: Real signal with real pulse: Convergence of V_t, τ_e, and τ_f.

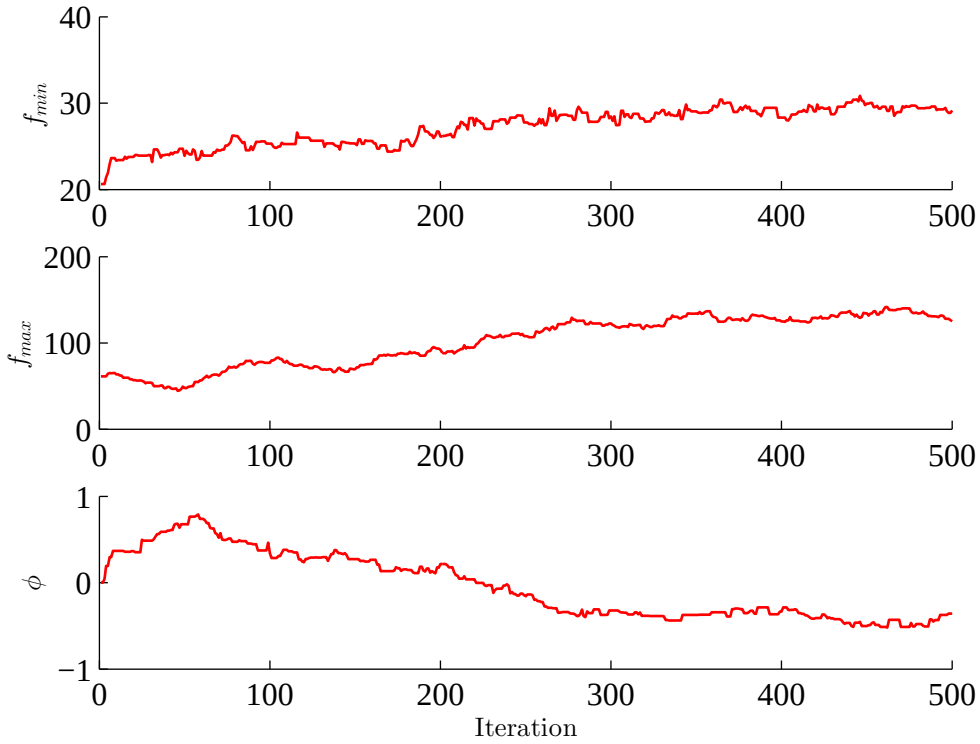

Figure 12.11: Real signal with real pulse: Convergence of $f_{\min}$, $f_{\max}$, and ϕ.

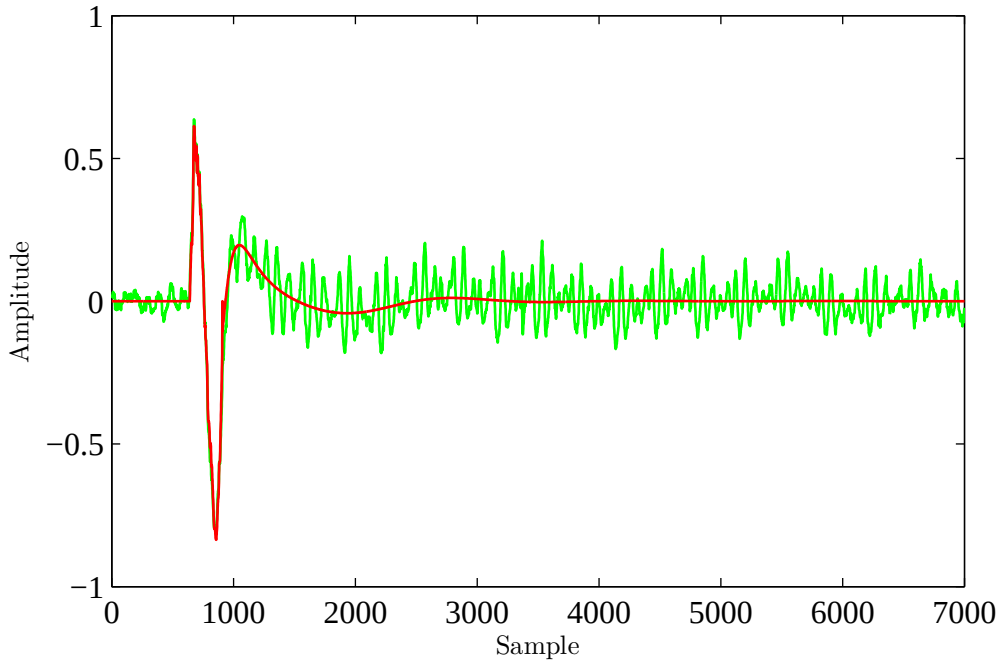

Figure 12.12: Real signal with real pulse: Comparison of the estimated pulse (red) with the original signal (green).

Chapter 13

Further developments in the long pulse problem: tail modeled by a Gaussian Process and an efficient initialization procedure for some variables

In the last chapter we presented a first attempt to restore audio signals degraded by long pulses, continuing the idea firstly presented in [98], where the tail of the pulse was modeled parametrically by a function depending on six parameters that must be estimated in order to remove its effect on the audio signal. Moreover, the initial discontinuity and its length were modeled by another three parameters, and the previously presented method lacks an effective initialization procedure for these variables. We present now two improvements in these directions: the tail of the pulse is now non-parametrically modeled via a Gaussian process, presented in Section 13.2, after a brief review of Gaussian processes in Section 13.1; and an initialization procedure for the location variables is presented in Section 13.3. Finally, in Section 13.4 some results are presented, followed by future works in Section 13.5.

13.1 A brief review of Gaussian processes

Recall that in Chapter 4 we defined a stochastic process as a collection of random variables $\{X_t\}$, for t in an arbitrary index set $\mathcal{T}$. A *Gaussian process* is a special kind of stochastic process, where every finite linear combination of the random variables in $\{X_t\}$ has a joint Gaussian multivariate distribution. Equivalently, we can say that for every finite set of indexes $t_1, \ldots, t_k \in \mathcal{T}$, the random vector $(X_{t_1}, \ldots, X_{t_k})$ is a

multivariate Gaussian random vector. The theory of Gaussian process is widely used in Statistics, in particular in Machine Learning and Linear Regression [69, 79, 105].

We can see the set $\{X_t\}$ as a function of variable t that we simply denote by X. The usual notation to indicate that it is a Gaussian process is $X \sim \mathcal{GP}(m, K)$, where $m_t = \mathbb{E}[X_t]$ is the mean function and $K(X_t, X_s) = \text{cov}(X_t, X_s)$ is the covariance kernel. Without loss of generality we can consider that m is identically zero, implying that every property of the process can be obtained from the covariance kernel only. Therefore, it is the most important part of the description of a Gaussian process, and some examples of covariance kernels widely used in the literature are presented below. An extensive discussion of their properties can be found in [79, 106]:

- Constant: $K(x, y) = C$
- Gaussian noise: $K(x, y) = \sigma^2 \delta_{x,y}$
- Squared-exponential: $K(x, y) = \sigma_f^2 \exp\left(-\dfrac{\|x - y\|^2}{2\sigma_l^2}\right)$
- Ornstein-Uhlenbeck: $K(x, y) = \exp\left(-\dfrac{\|x - y\|}{\ell}\right)$
- Matérn: $K(x, y) = \dfrac{2^{1-\nu}}{\Gamma(\nu)} \left(\dfrac{\sqrt{2\nu}\|x - y\|}{\ell}\right)^{\nu} K_\nu \left(\dfrac{\sqrt{2\nu}\|x - y\|}{\ell}\right)$, where K_ν denotes the modified Bessel function of order ν and Γ is the gamma function
- Periodic: $K(x, y) = \exp\left(-\dfrac{2\sin(\|x - y\|/2)}{\ell^2}\right)$
- Rational quadratic: $K(x, y) = (1 + \|x - y\|^2)^{\alpha}$, for $\alpha \geq 0$

Some of these kernel functions possess interesting interpretations and are specifically tailored for some special situations: the constant kernel ensures that all the random variables are identically correlated; the Gaussian noise kernel represents nothing more than independent and identically distributed Gaussian random variables; the squared exponential kernel is tailored to model smooth functions; the Matérn kernel possesses a simplified form with $\nu = n + 1/2$ for n integer, and is specially used in Machine Learning when $\nu = 3/2$ and $\nu = 5/2$; the periodic kernel is good to model periodicity within the process. Note that all of these kernels depend on some parameters that must be estimated or determined beforehand.

13.2 Pulse tail modeled by a Gaussian process

The idea of modeling the tail of the long pulse via a Gaussian process was motivated by the fact that the convergence of its respective variables in the parametric model

presented in the last chapter was quite complicated in some cases. One possible reason for this fact is that their posterior distribution is very complicated, implying the need to use a Metropolis step within the Gibbs sampler in order to estimate them. Moreover, the typical format of the tail is quite simple, so we would not expect to use so many (six) parameters to completely describe it. In this Section we present a framework to model the tail of the pulse non-parametrically via a Gaussian process, together with the respective estimation procedure.

To establish notation, denote the original signal by $\mathbf{x}$ and the corrupted signal by $\mathbf{y}$. As in the last chapter, we can split these signals in sub-signals denoted by $\mathbf{x}_0$, $\mathbf{y}_0$, $\mathbf{x}_1$, $\mathbf{y}_1$, $\mathbf{x}_2$ and $\mathbf{y}_2$ containing the region preceding the degradation (sub-index 0), the initial discontinuity (sub-index 1) and the pulse tail (sub-index 2) in the original and degraded signals respectively. We can then write

$$\mathbf{y}_0 = \mathbf{x}_0, \tag{13.1}$$

$$\mathbf{y}_1 = \mathbf{x}_1 + \mathbf{v}_d, \tag{13.2}$$

$$\mathbf{y}_2 = \mathbf{x}_2 + \mathbf{f}, \tag{13.3}$$

where $\mathbf{v}_d$ denotes the initial discontinuity and $\mathbf{f}$ is the pulse tail. Define auxiliary matrices $\mathbf{K}$, $\mathbf{U}_1$ and $\mathbf{U}_2$, sub-matrices of the identity matrix, such that

$$\mathbf{x} = \mathbf{K}\mathbf{x}_0 + \mathbf{U}_1\mathbf{x}_1 + \mathbf{U}_2\mathbf{x}_2. \tag{13.4}$$

Also define the matrices $\mathbf{A}_0 = \mathbf{A}\mathbf{K}$, $\mathbf{A}_1 = \mathbf{A}\mathbf{U}_1$ and $\mathbf{A}_2 = \mathbf{A}\mathbf{U}_2$, which will be useful in the estimation procedure.

The initial discontinuity is still modeled as white noise, beginning at time sample n_0 of the original signal and lasting for M time samples, with variance σ_d^2 fixed:

$$\mathbf{v}_d(n) = r(n)[u(n - n_0) - u(n - n_0 - M)], \tag{13.5}$$

where $r(n) \sim N(0, \sigma_d^2)$ and n_0, M and σ_d^2 are unknown a priori.

The pulse tail, before the sampling of the signal, can be seen as a generic function $f(t)$ that is superimposed to the original signal. It is known that this function is much smoother than the underlying signal, thus it is reasonable to model it via a Gaussian process with squared-exponential covariance kernel:

$$f \sim \mathcal{GP}(0, K), \tag{13.6}$$

where the covariance kernel computed at two time instants t_1 and t_2 is given by

$$K(t_1, t_2) = \sigma_f^2 \exp\left(-\frac{|t_1 - t_2|^2}{2\sigma_l^2}\right). \tag{13.7}$$

Parameters σ_f^2 and σ_l^2 control the amplitude of the pulse and the effective extent of the correlation kernel, respectively.

The choice of this covariance kernel to model the tail of a long pulse is questionable, since it models a stationary function f, which is clearly not the case of a typical tail of a long pulse. Nothing prevents us from defining a new covariance kernel specifically tailored to deal with long pulses, but it is possible that this kernel would also depend on a lot of parameters, similarly to the parametric model of the last chapter. But considering that we chose the non-parametric model because of its simplicity, this approach would make no sense.

When dealing with the time-sampled signal, we are considering time samples of function f, that we denote by vector $\mathbf{f}$. By the definition of a Gaussian process, we know that $\mathbf{f}$ has a joint multivariate Gaussian distribution with $\mathbf{0}$ mean and a covariance matrix that we will discuss later.

Since we are adopting the Gaussian process model in a Bayesian framework, it is possible to assign priors to the parameters of the covariance kernel and estimate them together with the remaining variables. However, in the preliminary tests performed until now, the values of these parameters were fixed in the beginning of the estimation procedure; more details about this are given in Section 13.4.

As previously done, assemble the parameters of the initial discontinuity and the AR model for the original signal in vectors $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_x$, respectively:

$$\boldsymbol{\theta}_d = [n_0 \quad M \quad \sigma_d^2]^T, \tag{13.8}$$

$$\boldsymbol{\theta}_x = [\mathbf{a}^T \quad \sigma_e^2]^T. \tag{13.9}$$

Recall that the vector $\boldsymbol{\theta}_t$ containing the parameters of the tail is not necessary anymore, since vector $\mathbf{f}$ contains the entire pulse.

As in the previous chapter, the parameters of the AR model for the underlying signal are not estimated together with the other unknowns, because this would imply an increase in the required computational time. Therefore, these quantities are estimated beforehand, by analyzing a region of the signal preceding the estimated beginning of the degradation.

13.2.1 Description of the algorithm

We wish to recover signal $\mathbf{x}$ from the observed degraded signal $\mathbf{y}$. The hypothesis that the original signal is well described by an AR model allows us to write a statistical description for it, and after some change of variables and long computations, for $\mathbf{y}$, $\boldsymbol{\theta}_d$ and $\mathbf{f}$. By using Bayes' Theorem, we can compute the posterior distribution for the quantities we wish to estimate, that is $p(\boldsymbol{\theta}_d, \mathbf{f}, \mathbf{x}|\boldsymbol{\theta}_x, \mathbf{y})$, which will be used to this end. Since the expression for this distribution is complicated and intractable

analytically, we sample from this distribution via a Gibbs sampler, eventually with some Metropolis steps, implemented in this way:

a) Initialize values $n_0^{(0)}$, $M^{(0)}$, $\sigma_d^{2^{(0)}}$, $\mathbf{f}^{(0)}$ and $\mathbf{x}^{(0)}$

b) For k from 1 to N_{iter}:

   i) Sample $n_0^{(k)}$ and $M^{(k)}$ from distribution

$$p(n_0, M|\sigma_d^{2^{(k-1)}}, \mathbf{f}^{(k-1)}, \mathbf{x}^{(k-1)}, \boldsymbol{\theta}_x, \mathbf{y})$$

   ii) Sample $\mathbf{f}^{(k)}$ from distribution

$$p(\mathbf{f}|n_0^{(k)}, M^{(k)}, \sigma_d^{2^{(k-1)}}, \mathbf{x}^{(k-1)}, \boldsymbol{\theta}_x, \mathbf{y})$$

   iii) Sample $\mathbf{x}^{(k)}$ from distribution

$$p(\mathbf{x}|n_0^{(k)}, M^{(k)}, \sigma_d^{2^{(k-1)}}, \mathbf{f}^{(k)}, \boldsymbol{\theta}_x, \mathbf{y})$$

   iv) Sample $\sigma_d^{2^{(k)}}$ from distribution

$$p(\sigma_d^2|n_0^{(k)}, M^{(k)}, \mathbf{f}_t^{(k)}, \mathbf{x}^{(k)}, \boldsymbol{\theta}_x, \mathbf{y}).$$

The mean of the posterior distribution is estimated by averaging the samples obtained after the burn-in time.

Note that the sampling procedures for variables n_0, M, σ_d^2, and $\mathbf{x}$ are identical to those presented in the previous chapter: there, these variables are conditioned on the entire vector $\boldsymbol{\theta}_t$, which uniquely determines the tail of the pulse; and here they are conditioned on $\mathbf{f}$, which obviously also uniquely determines the pulse tail.

We now describe the conditional distribution $p(\mathbf{f}|\boldsymbol{\theta}_d, \mathbf{x}, \boldsymbol{\theta}_x, \mathbf{y})$.

13.2.1.1 Computation of $p(\mathbf{f}|\boldsymbol{\theta}_d, \mathbf{x}, \boldsymbol{\theta}_x, \mathbf{y})$

By using Bayes' Theorem, we have that

$$p(\mathbf{f}|\boldsymbol{\theta}_d, \mathbf{x}, \boldsymbol{\theta}_x, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta}_d, \mathbf{x}, \boldsymbol{\theta}_x)p(\mathbf{f}). \tag{13.10}$$

Since we are supposing that $\mathbf{f}$ is well described by a Gaussian process with a quadratic-exponential covariance kernel, we have that $\mathbf{f} \sim N(\mathbf{0}, \mathbf{C}_\text{f})$, where matrix $\mathbf{C}_\text{f}$ is given by the Gram matrix of the covariance kernel function, computed from Equation 13.7. The likelihood $p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta}_d, \mathbf{x}, \boldsymbol{\theta}_x)$ was computed in the previous chapter

in Equation 12.24, and we recall its formula here, together with the necessary minor modifications:

$$p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta}_d,\mathbf{x},\boldsymbol{\theta}_x) = \frac{\lambda^M}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}\det(\boldsymbol{\Phi})}\exp\left\{-\frac{1}{2\sigma_e^2}E_{\min}\right\}, \tag{13.11}$$

where

$$E_{\min} = \lambda^2\mathbf{y}_1^T\mathbf{y}_1 + \mathbf{z}^T\begin{bmatrix}\mathbf{A}_0^T\\ \mathbf{A}_2^T\end{bmatrix}[\mathbf{A}_0\ \mathbf{A}_2]\mathbf{z} + (\mathbf{x}_1^{\text{MAP}})^T\boldsymbol{\Theta}, \tag{13.12}$$

$$\mathbf{z} = \begin{bmatrix}\mathbf{y}_0\\ \mathbf{y}_2 - \mathbf{f}\end{bmatrix}, \tag{13.13}$$

$$\mathbf{x}_1^{\text{MAP}} = \boldsymbol{\Phi}^{-1}\boldsymbol{\Theta}, \tag{13.14}$$

$$\boldsymbol{\Phi} = \lambda\mathbf{I}_M + \mathbf{A}_1^T\mathbf{A}_1, \tag{13.15}$$

$$\boldsymbol{\Theta} = \lambda\mathbf{y}_1 - \mathbf{A}_1^T[\mathbf{A}_0\ \mathbf{A}_2]\mathbf{z}. \tag{13.16}$$

We noticed that by considering $\lambda = 0$ this expression is much simplified, and this is a reasonable hypothesis, as previously discussed. We have then

$$p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta}_d,\mathbf{x},\boldsymbol{\theta}_x) \propto \exp\left(-\frac{1}{2}\mathbf{z}^T\mathbf{R}\mathbf{z}\right), \tag{13.17}$$

where

$$\mathbf{R} = \frac{1}{\sigma_e^2}\begin{bmatrix}\mathbf{A}_0^T\\ \mathbf{A}_2^T\end{bmatrix}\mathbf{S}[\mathbf{A}_0\ \mathbf{A}_2], \tag{13.18}$$

$$\mathbf{S} = \mathbf{I}_{N-P} - \mathbf{A}_1(\mathbf{A}^{-1}\mathbf{A})^{-1}\mathbf{A}_1^T. \tag{13.19}$$

Therefore, the conditional distribution for $\mathbf{f}$ is given by

$$p(\mathbf{f}|\boldsymbol{\theta}_d,\mathbf{x},\boldsymbol{\theta}_x,\mathbf{y}) \propto \exp\left(-\frac{1}{2}\mathbf{f}^T\mathbf{C}_{\text{f}}\mathbf{f}\right)\exp\left(-\frac{1}{2}\mathbf{z}^T\mathbf{R}\mathbf{z}\right) \tag{13.20}$$

$$= \exp\left\{-\frac{1}{2}\left(\mathbf{f}^T\mathbf{C}_{\text{f}}\mathbf{f} + \mathbf{z}^T\mathbf{R}\mathbf{z}\right)\right\}. \tag{13.21}$$

We must then compute the term $\mathbf{z}^T\mathbf{R}\mathbf{z}$ in order to explicit its dependence on $\mathbf{f}$. To this end, note that it can be rewritten as:

$$\mathbf{z}^T\mathbf{R}\mathbf{z} = [\mathbf{y}_0^T\ \ (\mathbf{y}_2-\mathbf{f})^T]\left[\begin{array}{c|c}\mathbf{R}_{11} & \mathbf{R}_{12}\\ \hline \mathbf{R}_{21} & \mathbf{R}_{22}\end{array}\right]\begin{bmatrix}\mathbf{y}_0\\ \mathbf{y}_2-\mathbf{f}\end{bmatrix} \tag{13.22}$$

$$= -\mathbf{y}_0^T\mathbf{R}_{12}\mathbf{f} - \mathbf{f}^T\mathbf{R}_{21}\mathbf{y}_0 - \mathbf{y}_2^T\mathbf{R}_{22}\mathbf{f} - \mathbf{f}^T\mathbf{R}_{22}\mathbf{y}_2 + \mathbf{f}^T\mathbf{R}_{22}\mathbf{f} + \text{terms not depending on } \mathbf{f}. \tag{13.23}$$

Therefore, we have that

$$\begin{aligned}\mathbf{f}^T\mathbf{C}_\mathrm{f}\mathbf{f} + \mathbf{z}^T\mathbf{R}\mathbf{z} = &-\mathbf{y}_0^T\mathbf{R}_{12}\mathbf{f} - \mathbf{f}^T\mathbf{R}_{21}\mathbf{y}_0 - \mathbf{y}_2^T\mathbf{R}_{22}\mathbf{f} - \\ &\mathbf{f}^T\mathbf{R}_{22}\mathbf{y}_2 + \mathbf{f}^T\mathbf{R}_{22}\mathbf{f} + \mathbf{f}^T\mathbf{C}_\mathrm{f}\mathbf{f} + \\ &\text{terms not depending on } \mathbf{f}.\end{aligned} \tag{13.24}$$

Since this expression is quadratic in $\mathbf{f}$, the conditional distribution for $\mathbf{f}$ is a Gaussian whose mean and covariance matrix can be easily computed by finding the critical point and the negative Hessian of the expression above, respectively, as indicated in Section 12.3.1.1.2. We then have that $\mathbf{f}|\boldsymbol{\theta}_d, \mathbf{x}, \boldsymbol{\theta}_x, \mathbf{y} \sim N(\bar{\mathbf{f}}, \boldsymbol{\Sigma}_\mathrm{f})$, where

$$\bar{\mathbf{f}} = [\mathbf{R}_{22} + \mathbf{C}_\mathrm{f}^{-1} + \mathbf{R}_{22}^T + \mathbf{C}_\mathrm{f}^{-T}]^{-1}[\mathbf{R}_{12}^T\mathbf{y}_0 + \mathbf{R}_{21}\mathbf{y}_0 + (\mathbf{R}_{22}^T + \mathbf{R}_{22}\mathbf{y}_2)], \tag{13.25}$$

$$\boldsymbol{\Sigma}_\mathrm{f} = \left[\frac{1}{2}(\mathbf{R}_{22} + \mathbf{C}_\mathrm{f}^{-1} + \mathbf{R}_{22}^T + \mathbf{C}_\mathrm{f}^{-T})\right]^{-1}. \tag{13.26}$$

We now impose two additional simplifications:

- Since the tail of the pulse varies much slower than the original underlying signal we can abandon its AR structure, at least in the region of the pulse tail. This corresponds to considering matrix $\mathbf{A}_2$ as an identity matrix, and then matrix $\mathbf{R}_{22}$ turns out to be diagonal with constant terms. Intuitively, we are considering that the underlying original signal is essentially white noise when compared to the tail of the pulse.

- Sampling from the distribution $N(\bar{\mathbf{f}}, \boldsymbol{\Sigma}_\mathrm{f})$ is very expensive, since the size of $\mathbf{f}$ can be of the order of thousands of time samples. Therefore, instead of sampling from it, at each step of the algorithm we simply compute $\bar{\mathbf{f}}$ and consider its value as the current value of $\mathbf{f}$.

Finally, to completely explain the algorithm, we must describe the computation of initial values $n_0^{(0)}$, $M^{(0)}$, $\sigma_d^{2^{(0)}}$, $\mathbf{f}^{(0)}$, and $\mathbf{x}^{(0)}$:

- The initialization of n_0 and M requires a long explanation, and is described in more detail in Section 13.3.

- The only unknown part in the variable $\mathbf{x}$ that is directly sampled in the algorithm is $\mathbf{x}_1$. It is initialized simply with zeros, meaning that no previous knowledge about the signal underlying the initial discontinuity is available.

- The variable σ_d^2 is initialized by computing the variance of the initial discontinuity, using the estimate provided by the initialization of n_0 and M.

- The initialization of $\mathbf{f}$ is performed by using function `fitrgp` of MATLAB$^{\mathrm{TM}}$, available in the last versions of the Statistics and Machine Learning

Toolbox$^{\text{TM}}$, where the beginning of the pulse is given by the initial estimate of n_0 and M. This function also estimates parameters σ_l^2 and σ_f^2, which are kept fixed during the entire procedure.

13.3 Initialization procedure for the location variables

As noticed in the last chapter, an accurate initialization of variables n_0 and M was important for the effective convergence of the Gibbs sampler. This initialization was performed by hand, and a method for automatically doing this task was required. In [107] a method for detection of long pulses in audio signals is presented, and we adapted this method to provide a quite accurate estimate of the desired variables.

The initialization procedure looks for unusual bursts of high-frequencies in a time-frequency domain, typical of the initial discontinuity of the pulse. More precisely, the corrupted signal is split in contiguous blocks of length L with an overlap of 50% between adjacent blocks, and the discrete Fourier Transform (DFT) [43] of each block is computed. Denote the DFT of block b by $\hat{\mathbf{y}}_b$, for $b = 1, \dots B$. Since there is significant audio content up to some frequency, it is more convenient to look for unusual information in vectors $\hat{\mathbf{y}}_b$ starting at some frequency defined by the user. Denote this cut-off frequency by f_{co} and the respective frequency bin by α_{co}. Define then function μ by

$$\mu(b) = \frac{1}{\beta - \alpha_{\text{co}} + 1} \sum_{k=\alpha_{\text{co}}}^{\beta} |\hat{\mathbf{y}}_b(k)|, \tag{13.27}$$

where β is the last frequency bin of the considered block. This function is an arithmetic mean of the high frequency content of each block, starting at frequency f_{co}, and we expect that a high value of it represents an initial discontinuity present in block b of the degraded signal. However, some problems may occur: as reported in [107], if the considered signal possesses a broad dynamic range with substantial high-frequency content (for example, brass or percussive instruments), the value of μ can be high but not necessarily corresponding to the presence of long pulses.

In order to circumvent this issue, a median filter is used in function μ. As it is known in the literature, the median filter has the property of removing local occurrences of unusual values within a sequence, and is widely used in Image Processing as a tool to remove impulsive noises while preserving edges [108][1]. The procedure is described below:

- Define a new function μ_m, resulting by median filtering μ with an odd-sized

[1]There is a folk Theorem stating that "median filtering outperforms linear filtering for supressing noise in image with edges", that recently was proven to be false, in general [109].

window of length c, that is, each value of μ is substituted by the median of the c values centered on it.

- In order for the functions μ_m and μ to have the same domain, we pad function μ, previously to the median filtering, with $\lfloor c/2 \rfloor$ zeros before and after its first and last values, respectively.

- Define function $\Delta\mu(b)$ as the difference between μ and μ_m normalized by its highest value, that is,

$$\Delta\mu(b) = \frac{\mu(b) - \mu_m(b)}{\max\limits_{b'}[\mu(b') - \mu_m(b')]}, \tag{13.28}$$

  for $b = 1, \ldots, B$. This ensures that the maximum absolute value of $\Delta\mu$ is one, thus allowing an easier definition of the threshold specified below.

- Define a threshold λ such that block b^* is considered corrupted by the initial discontinuity of a long pulse if $|\Delta\mu(b^*)| \geq \lambda$.

This procedure defines a set of blocks $b_1^*, \ldots, b_M^*$, possibly corrupted by initial discontinuities of long pulses, and assumes that these blocks are causally ordered. We need to use this information to generate initial estimates for n_0 and M. If the blocks above are contiguous, we define $n_0^{(0)}$ as the first time sample of block b_1^*, and $M^{(0)}$ as the gap size between the last time sample of block b_M^* and $n_0^{(0)}$. If the blocks are not contiguous, the signal is possibly corrupted by more than one pulse, and for each contiguous set of selected blocks (which we expect to represent each pulse within the signal) we perform this procedure, obtaining an initial estimate of n_0 and M for each pulse present in the signal.

In order to better estimate these variables, the value of L, representing the length of each block prior to the computation of its DFT, must be discussed. In [107] it was adopted $L = 2048$, but now this choice would lead to a very crude estimate of n_0 and M. Moreover, recall from Chapter 12 that the size of the analyzed signal around the pulse is at most 10,000 time samples, implying that this signal can be split in a very small number of blocks. Since we are not interested in a good frequency resolution but only in a good time resolution and in the frequency content, we use a small window of length L between 16 and 64, depending on the considered signal being noisy or not. The values of L, F_{co}, and λ can be chosen by the user, and we noticed that the aforementioned window size and λ around 0.4 lead to a good initial estimate of the desired variables, which the Gibbs sampler is able to fix and then arrive at its correct value. This point will be further discussed in the next section, together with some results.

13.4 Results

Since in this chapter two independent methods to perform two tasks were presented, first we show here some results using techniques from Chapter 12: in Section 13.4.1 we incorporate the initialization procedure here proposed to the estimation scheme previously presented, where the pulse was described parametrically then, and in Section 13.4.2 we show the results of the initialization procedure together with the nonparametric model of the pulse.

All the signals are available in [72] under the link *Companion website to my D.Sc. Thesis*.

13.4.1 Pulse described parametrically with the initialization procedure

Tests were performed with a real signal corrupted by an artificial pulse following the parametric model and with a real degraded signal. The results are split in the next two sections, to improve the organization. The main goal of this section is to provide results to compare with those presented in Section 12.4, since the main difference between both methods being tested is just the initialization procedure.

13.4.1.1 Real signal with artificial pulse

An arbitrary excerpt containing 10,000 time samples of the signal `classical.wav` was extracted and corrupted artificially by a pulse following the parametric model, as in the last chapter, with the following parameters: $n_0 = 500$, $M = 10$, $\sigma_d^2 = 0.5$, $V_t = 0.3$, $\tau_e = 0.07$, $\tau_f = 0.013$, $f_{\max} = 60$, $f_{\min} = 20$, and $\phi = 0$. Recall that these parameters were chosen by hand in order to roughly describe a typical pulse present in audio signals. The variables of the initialization procedure were fixed at $L = 16$, $f_{\text{co}} = 3,000$ Hz, $c = 5$, and $\lambda = 0.4$.

The algorithm was run for 200 iterations, with a burn-in time of 150 iterations. Therefore, the estimated parameters consists of the average of the last 50 samples. The time per iteration was also about 2.35 s, since the estimation procedure is similar to that in the last chapter. In Table 13.1 we can compare the original, initial and estimated values for $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_t$, and Figures 13.2, 13.3, and 13.4 show the convergence of the parameters in $\boldsymbol{\theta}_d$ and first and second halves of $\boldsymbol{\theta}_t$, respectively. Green squares are real values, and red circles represent the estimated values for each parameter. Finally, in Figure 13.5 we can compare the estimated pulse (in red) with the original one (in blue). The degraded signal is also shown, in green. In Figure 13.1 function $\Delta\mu$ is plotted, and we can see that only one pulse was detected within the degraded signal, as expected.

Note that the initialization procedure was able to estimate the correct values of n_0 and M quite accurately, and all the variables in vector $\boldsymbol{\theta}$ were estimated in a way such that the restored signal presents no audible difference with respect to the original one.

Table 13.1: Real signal with artificial pulse: comparison of real, initial and estimated values for parameters in $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_t$.

	Real values	Initial values	Estimated values
n_0	500	504	500
M	10	8	10
σ_d^2	0.5	0.2128	0.5897
V_t	0.3	0.5	0.3302
τ_e	0.07	0.09	0.0652
τ_f	0.013	0.010	0.0098
$f_{\max}$	60	55	80.5181
$f_{\min}$	20	25	20.0280
ϕ	0	0.3	0.0521

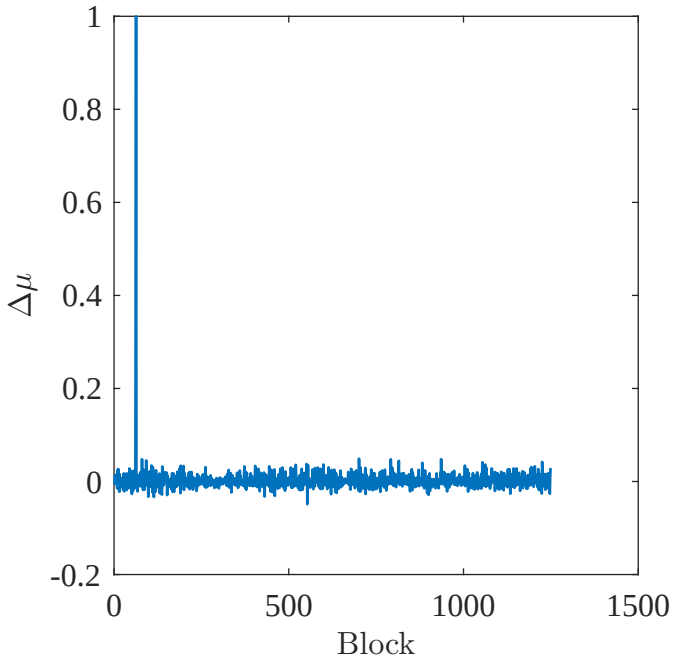

Figure 13.1: Function $\Delta\mu$ for real signal with artificial pulse.

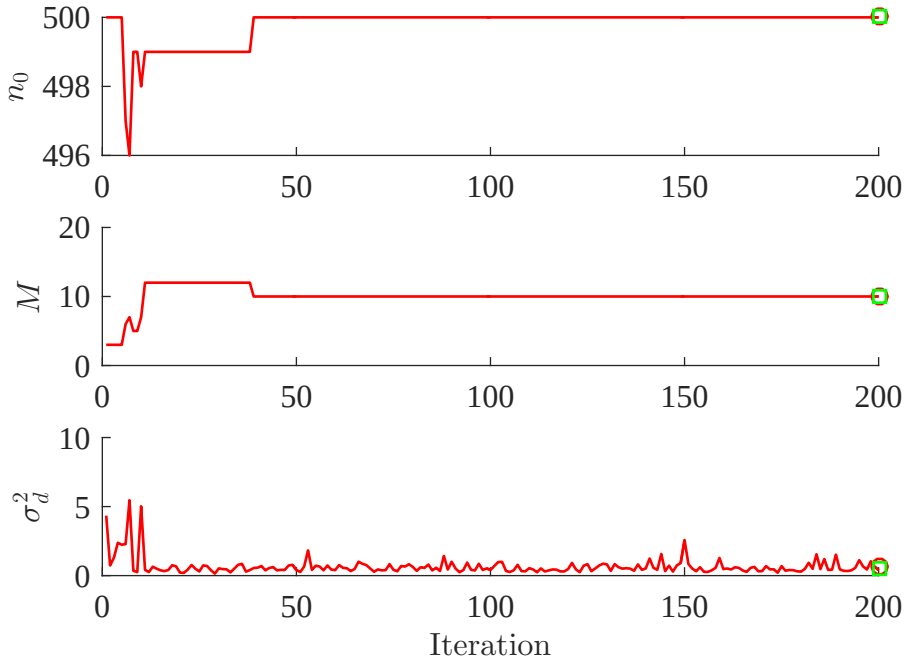

Figure 13.2: Real signal with artificial pulse and initialization procedure: Convergence of $\boldsymbol{\theta}_d$.

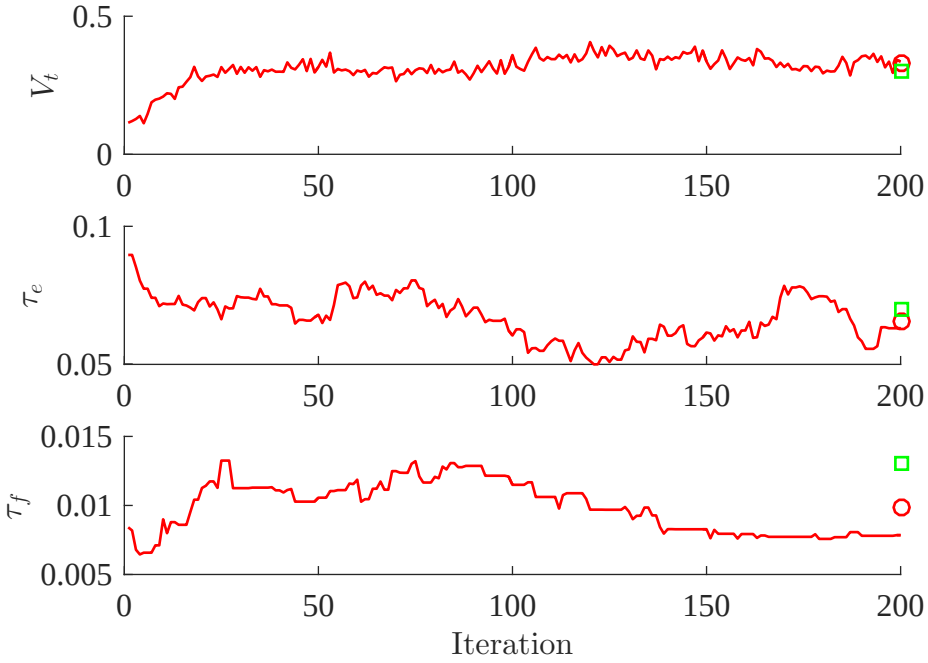

Figure 13.3: Real signal with artificial pulse and initialization procedure: Convergence of V_t, τ_e, and τ_f.

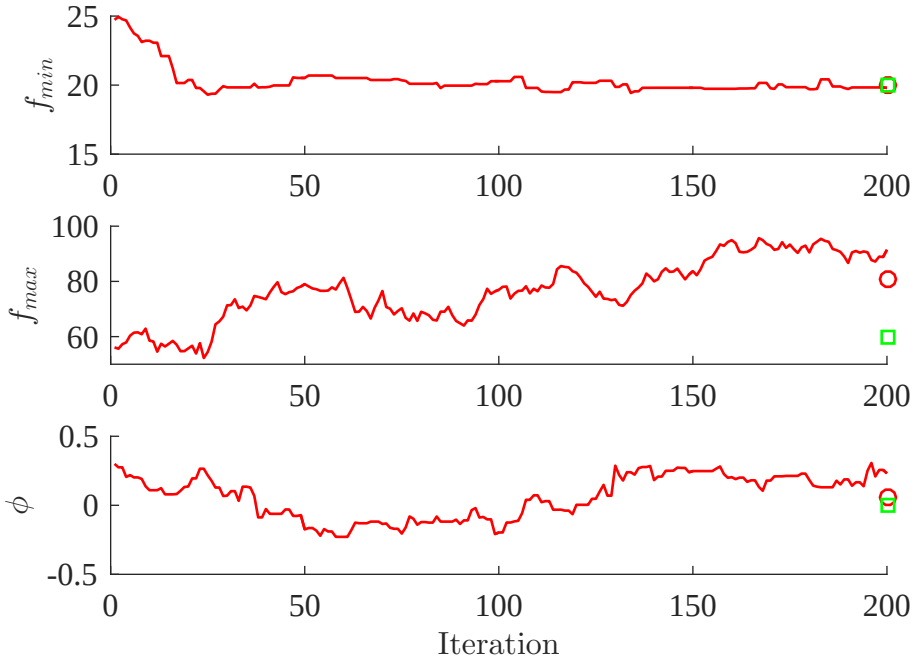

Figure 13.4: Real signal with artificial pulse and initialization procedure: Convergence of $f_{\min}$, $f_{\max}$, and ϕ.

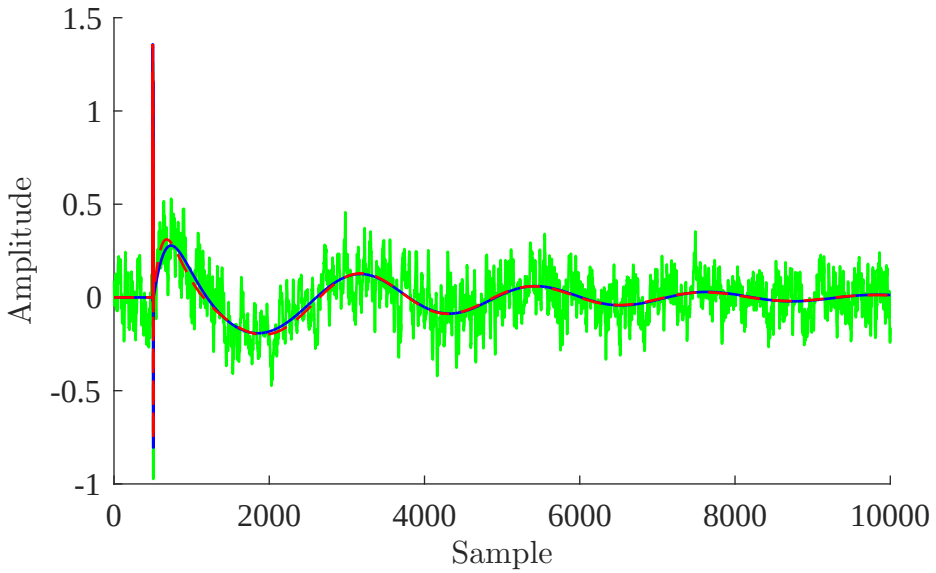

Figure 13.5: Real signal with artificial pulse and initialization procedure: Comparison of the estimated pulse (red) with the original pulse (blue). The degraded signal is shown in green.

13.4.1.2 Real degraded signal

As in Section 12.4, the recording from cylinder number 154 from [104], approximately from 1907 and corrupted with a long pulse, was analyzed. An excerpt of 7,000 time samples around the pulse was isolated, and the parameters of the AR model were estimated from the first 600 time samples. The algorithm was run for 500 iterations, with a burn-in time of 400 iterations. Therefore, the estimated parameters are given by the mean of the last 100 iterations.

Since the considered signal is also corrupted with background noise, the size of the window in the initialization procedure was chosen to be 64 time samples, and for the same reason the cut-off frequency was chosen to be $f_{\text{co}} = 10,000$ Hz. The threshold λ was chosen as 0.4 and the size of the median filter was 30 blocks. In Table 13.2 we can compare the initial and estimated values for $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_t$, and Figures 13.7, 13.8, and 13.9 show the convergence of the parameters in $\boldsymbol{\theta}_d$ and first and second halves of $\boldsymbol{\theta}_t$, respectively. Finally, in Figure 13.10 we can compare the estimated pulse (in red) with the original signal (in green). In the figures depicting convergence of variables the red circle denotes the respective estimated value.

We note again that the initialization procedure is capable of correctly locating the initial discontinuity, with some tuning of its parameters. This task is necessary because of the fact that the signal is also corrupted with background noise.

Table 13.2: Real signal with real pulse and initialization procedure: comparison of initial and estimated values for parameters in $\boldsymbol{\theta}_d$ and $\boldsymbol{\theta}_t$.

	Initial values	Estimated values
n_0	640	643
M	288	278
σ_d^2	0.2132	0.2351
V_t	0.5	0.2239
τ_e	0.09	0.0150
τ_f	0.010	0.0071
f_{max}	55	85.7401
f_{min}	25	26.0414
ϕ	0.3	0.1786

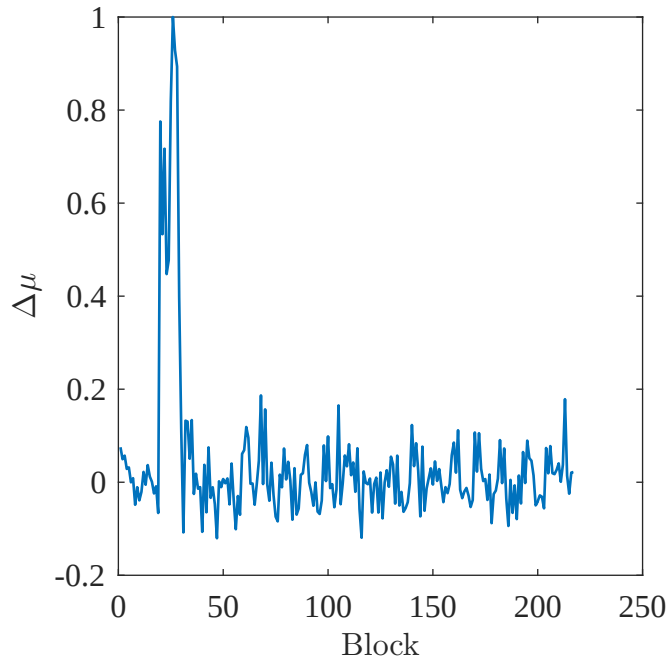

Figure 13.6: Function $\Delta\mu$ for real signal with artificial pulse.

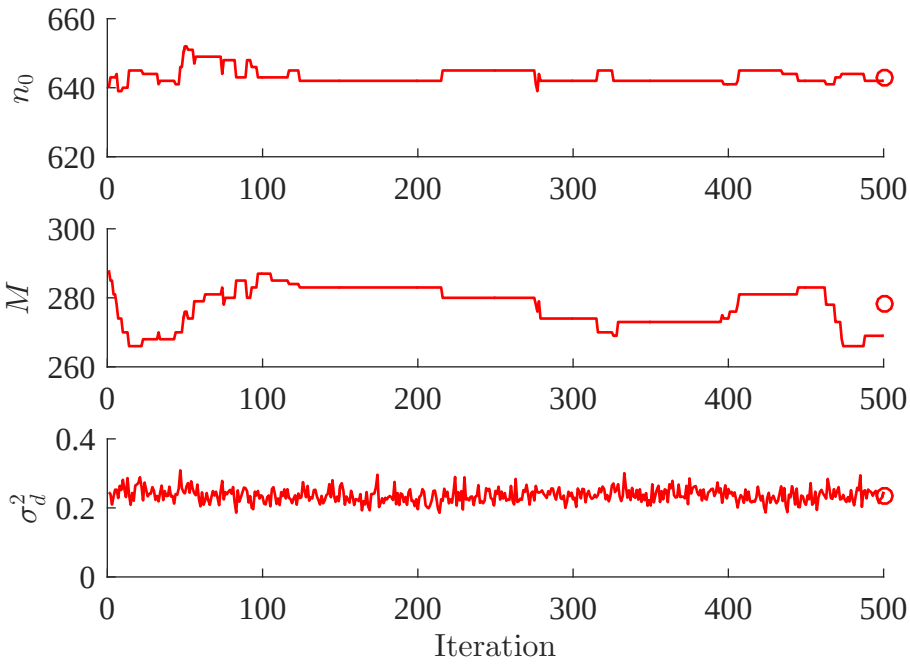

Figure 13.7: Real signal with real pulse and initialization procedure: Convergence of $\boldsymbol{\theta}_d$.

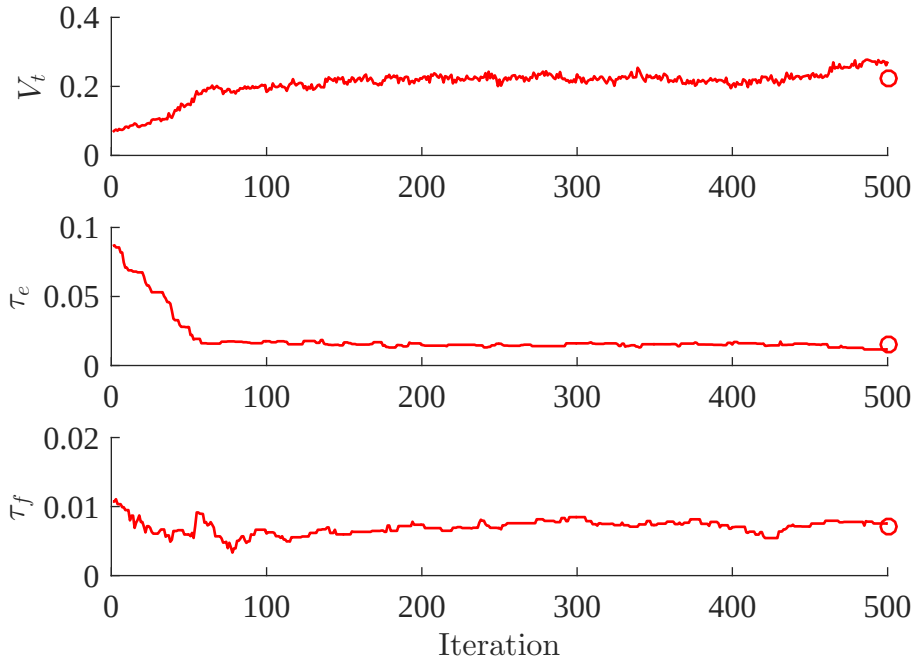

Figure 13.8: Real signal with real pulse and initialization procedure: Convergence of V_t, τ_e, and τ_f.

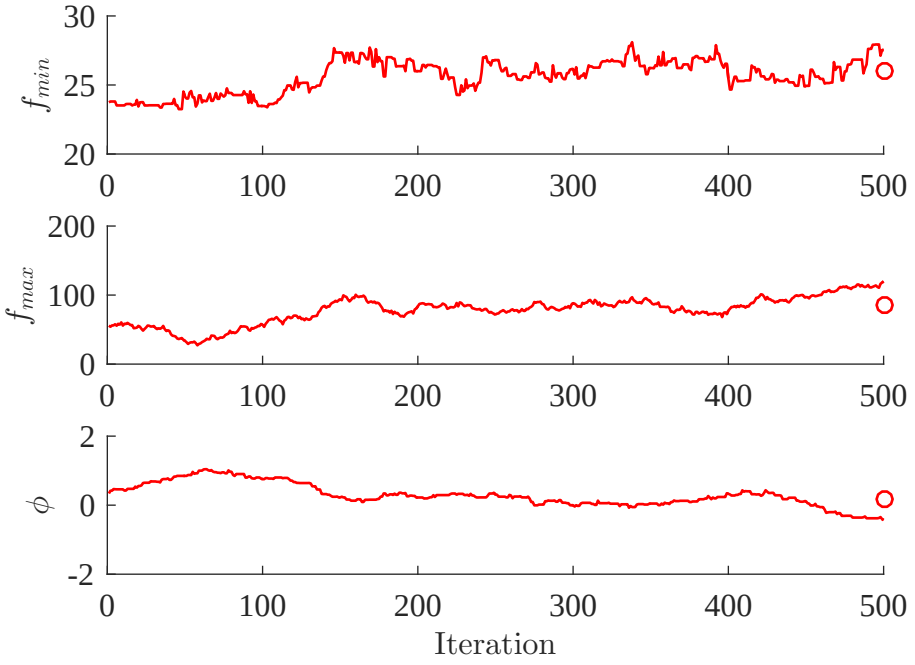

Figure 13.9: Real signal with real pulse and initialization procedure: Convergence of $f_{\min}$, $f_{\max}$, and ϕ.

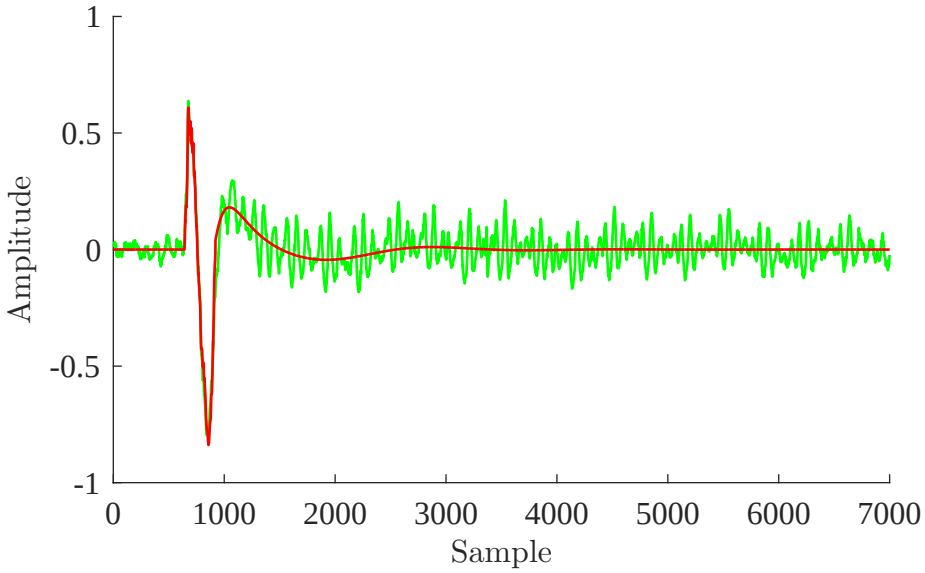

Figure 13.10: Real signal with real pulse and initialization procedure: Comparison of the estimated pulse (red) with the original signal (green).

13.4.2 Pulse described by a Gaussian process with the initialization procedure

We now merge the two improvements proposed in this chapter, and abandon the parametric model for the pulse. The same tests as before were performed, and the results are split in the next two subsections, for organization sake.

13.4.2.1 Real signal with artificial pulse

An excerpt consisting of 7,000 time samples of the signal `classical.wav` was extracted and corrupted artificially by a pulse following the parametric model with the same parameters as before. Recall that the excerpt previously considered consisted of 10,000 time samples, but now we decreased its size because of the computational cost required, further discussed below. The initialization procedure was performed with $L = 16$, $f_{\text{co}} = 3,000$ Hz, $c = 5$, and $\lambda = 0.4$.

Since there is no need to estimate the variables in $\boldsymbol{\theta}_t$ anymore, the algorithm was run for only 50 iterations, being the first 25 considered the burn-in time. The time per iteration is much larger than before, of about 14 s, the reason for this being the need of compute and operate with the covariance matrix of a multidimensional Gaussian of size approximately 6,000 $\times$ 6,000 at each iteration, in order to estimate the tail of the pulse. Also for this reason the considered excerpt is smaller than before, with no impairment on the quality of the restored signal[2].

In Table 13.3 we can compare the original, initial and estimated values for $\boldsymbol{\theta}_d$ and Figure 13.12 show the convergence of the parameters in $\boldsymbol{\theta}_d$, where green squares are real values, and red circles represent the estimated values for each parameter. Finally, in Figure 13.13 we can compare the estimated pulse (in red) with the original one (in blue). The degraded signal is also shown, in green. In Figure 13.11 function $\Delta\mu$ is plotted, and we can see that only one pulse was detected within the degraded signal, as expected.

We note from this set of results that the Gaussian process model for the tail of the pulse is able to estimate it correctly, despite the squared exponential covariance kernel not being the most adequate one to model its characteristics.

[2]When trying to run the algorithm with a block of size 10,000 we exceeded the capacity of the memory of my computer, 8 GB. By considering only 7,000 time samples, the memory use is at most 6,5 GB.

Table 13.3: Real signal with artificial pulse: comparison of real, initial and estimated values for parameters in $\boldsymbol{\theta}_d$.

	Real values	Initial values	Estimated values
n_0	500	504	499
M	10	8	11
σ_d^2	0.5	0.6953	0.9667

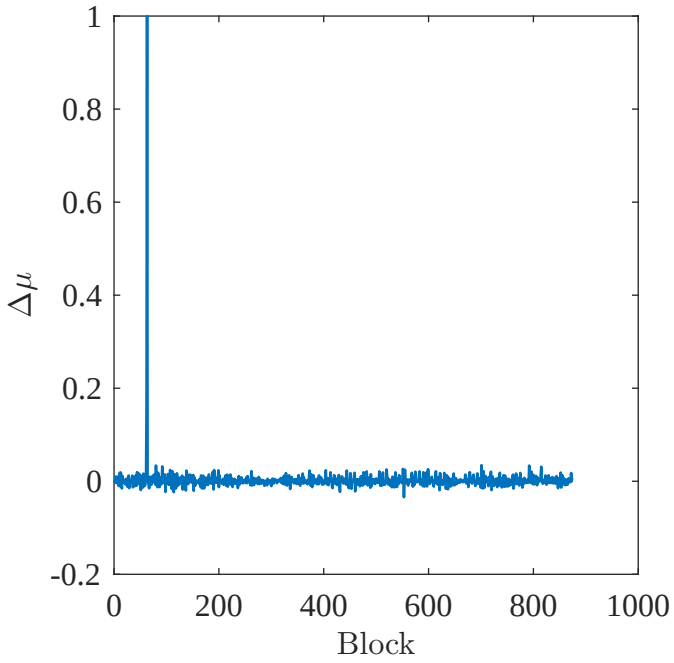

Figure 13.11: Function $\Delta\mu$ for real signal with artificial pulse, estimated by a Gaussian process.

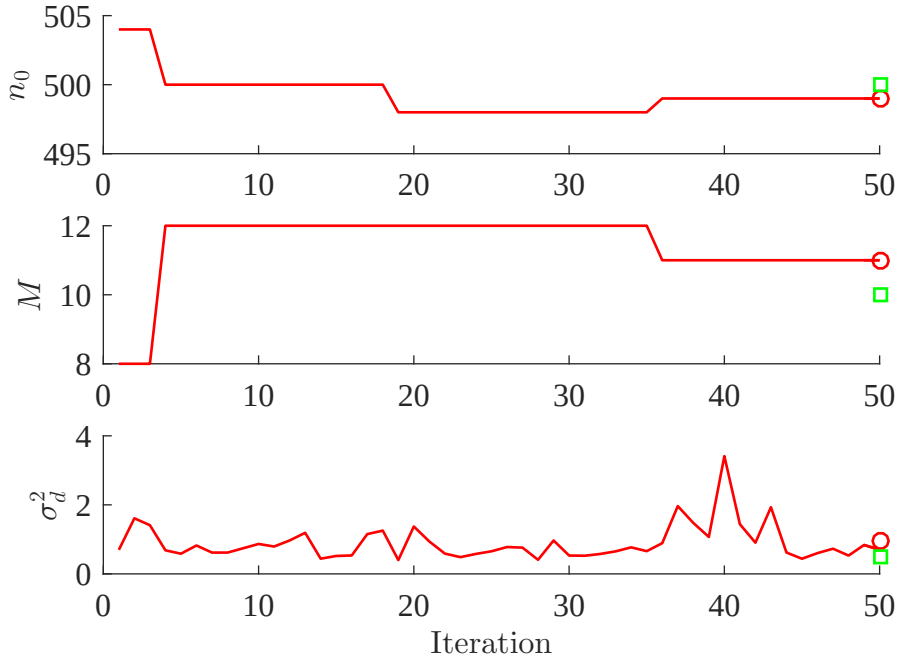

Figure 13.12: Real signal with artificial pulse estimated by a Gaussian process and initialization procedure: Convergence of $\boldsymbol{\theta}_d$.

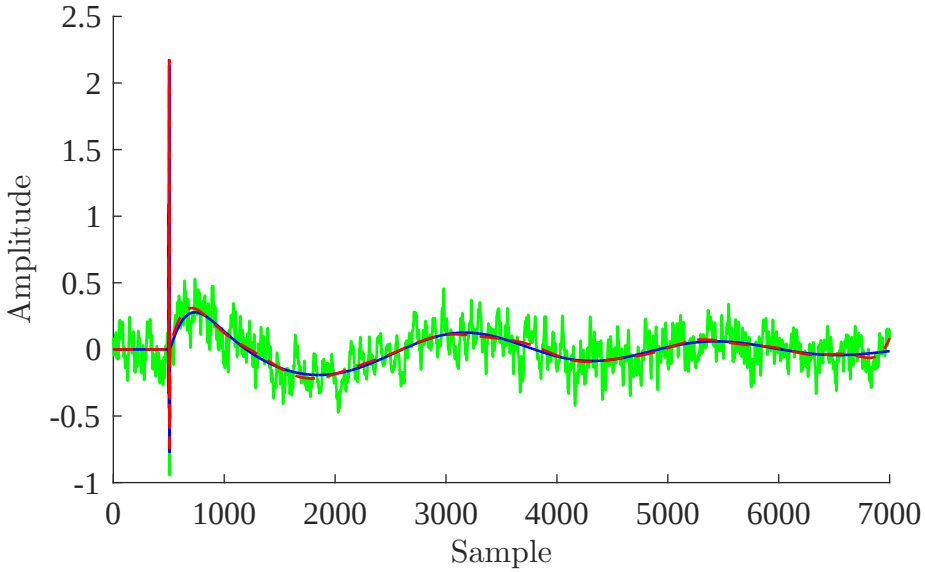

Figure 13.13: Real signal with artificial pulse estimated by a Gaussian process and initialization procedure: Comparison of the estimated pulse (red) with the original pulse (blue). The degraded signal is shown in green.

13.4.2.2 Real degraded signal

Finally, we present the last test, where the previously considered real degraded signal is restored using the Gaussian process model for the tail of the pulse. The same excerpt of 7,000 time samples around the pulse was considered, and the parameters of the AR model were also estimated from the first 600 time samples. The algorithm was run for 40 iterations, half of which are considered as the burn-in time. The parameters of the initialization procedure were the same as before: $L = 64$, $f_{\text{co}} = 10,000$ Hz, $\lambda = 0.4$ and $c = 30$.

In Table 13.4 we can compare the initial and estimated values for $\boldsymbol{\theta}_d$, and Figure 13.15 shows the convergence of the parameters in $\boldsymbol{\theta}_d$, where the red circles represent the estimated values. In Figure 13.16 we can compare the estimated pulse (in red) with the original signal (in green). In Figure 13.14 function $\Delta\mu$ is plotted.

Table 13.4: Real signal with real pulse and initialization procedure: comparison of initial and estimated values for parameters in $\boldsymbol{\theta}_d$.

	Initial values	Estimated values
n_0	640	638
M	288	268
σ_d^2	0.2132	0.3350

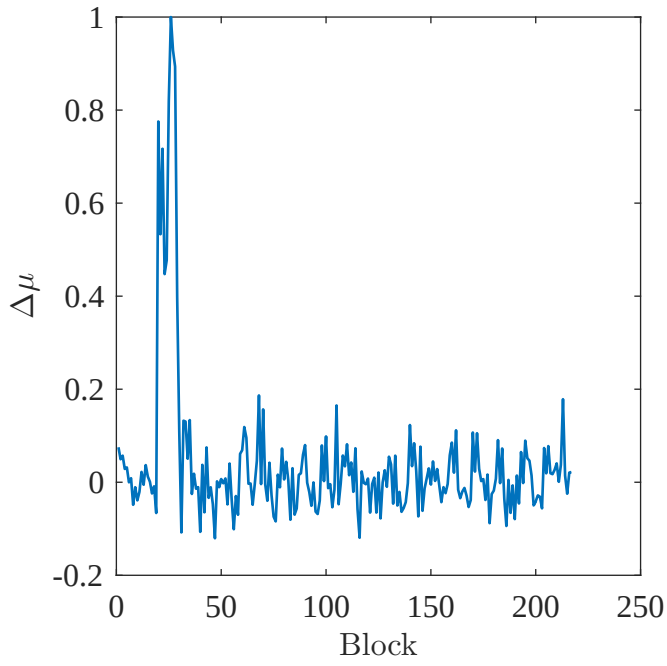

Figure 13.14: Function $\Delta\mu$ for real signal with real pulse estimated by a Gaussian process.

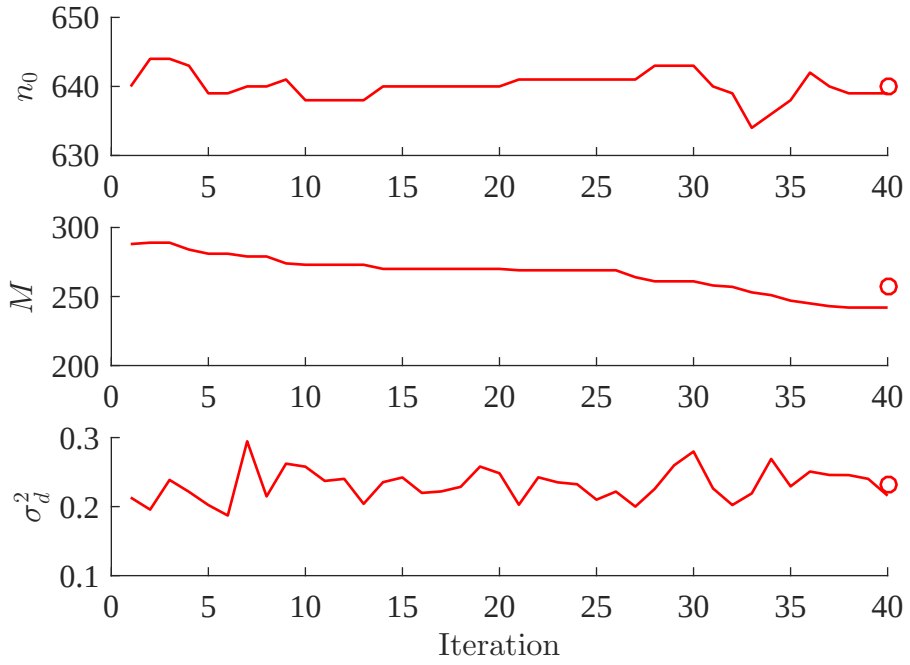

Figure 13.15: Real signal with real pulse estimated by a Gaussian process and initialization procedure: Convergence of $\boldsymbol{\theta}_d$.

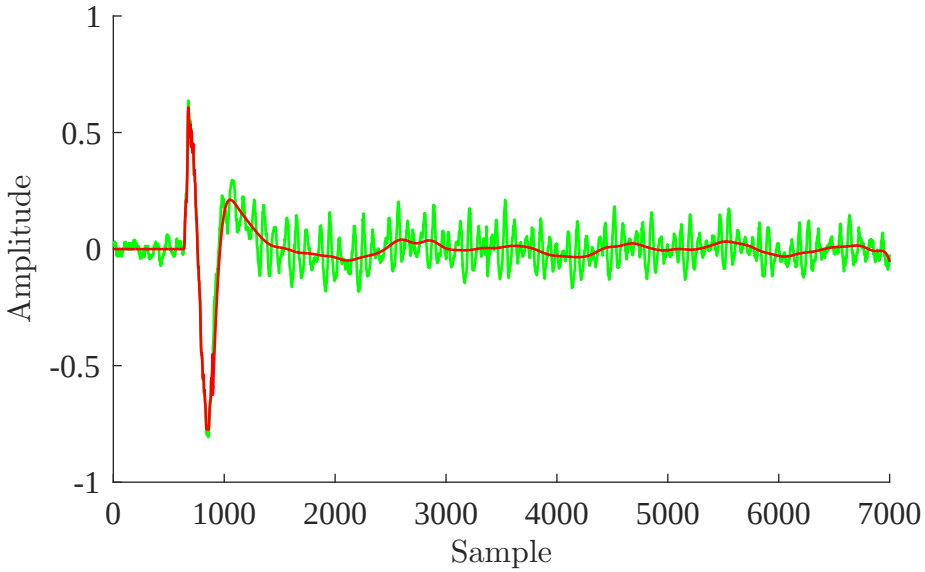

Figure 13.16: Real signal with real pulse estimated by a Gaussian process and initialization procedure: Comparison of the estimated pulse (red) with the original signal.

From this set of results we conclude that modeling the tail of the pulse by a Gaussian process is also effective when dealing with real degraded signals, with the advantage of its estimate being much simpler than the estimation procedure of the parameters in the parametric model.

13.5 Conclusion and future works

In this chapter we presented two improvements to the solution of the problem of long pulse suppression: an efficient initialization procedure for the variables in $\boldsymbol{\theta}_d$; and a nonparametric and more flexible model for the pulse tail. We conclude from the presented results that these two techniques, together with the sampling scheme for $\boldsymbol{\theta}_d$ presented in Chapter 12, provide a good framework for joint location and suppression of long pulses in audio signals. There is the need to tune some parameters, specially in the initialization procedure, but this task is much easier than tuning the parameters of the proposal distribution for $\boldsymbol{\theta}_t$ in Chapter 12. We note that the introduction of the Gaussian process model for the pulse tail implies an increase in the required computational cost, but not to a critical level: the increase in the time spent per iteration is compensated by the need of fewer iterations in order for the sampler to properly converge.

We believe that the content in this chapter and the previous one provides a solution to the considered problem that is good enough to be published, and a paper with this content is being prepared, where more extensive tests with real degraded signals will be performed, in addition to comparisons with previously proposed methods.

A possible future work is to consider the detection of initial discontinuities not by an arbitrarily fixed threshold value applied on function $\Delta\mu$, but in the context of Statistical Decision Theory: it is possible to create a statistical model for the frequency content of a block of an audio signal (e.g., in [83] a Markov chain model is proposed), and treat the problem of deciding if a block is corrupted or not by an initial discontinuity as hypothesis testing.

A drawback of the proposed method is the interpolation of the missing signal in the initial discontinuity: it is performed as a maximum a posteriori procedure, and as reported in [37] this is not very effective with long initial discontinuities. Indeed, when considering the real cylinder recording, even after the convergence of the algorithm it is possible to hear a small *click* in the beginning of the pulse. A possible way to improve this part of the algorithm is to substitute the sampling of the $\mathbf{x}_1$ variable in the Gibbs sampler for a more efficient interpolation scheme [82, 110]. This will not lead to the classical sampler, since it is not guaranteed that this interpolation will be equivalent to a sample of the conditional distribution of $\mathbf{x}_1$, but we believe that it will not impair the convergence properties of the sampler.

Another possibility is to estimate the parameters of the AR model along the procedure, which implies an increase in the required computational power. This could improve the interpolation of the missing signal in the initial discontinuity.

We also believe that this approach works when dealing with multiple superimposed pulses, a degradation which appears when a big scratch is close to another in

a disk recording, for example. In order to treat this degradation some adaptations must be made in the algorithm, and this will be performed very soon in order to be included in the paper being prepared with the content of this chapter.

Part V

CONCLUSION

Chapter 14

Conclusion and future works

> *"Let me tell you something you already know. The world ain't all sunshine and rainbows. It's a very mean and nasty place and I don't care how tough you are it will beat you to your knees and keep you there permanently if you let it. You, me, or nobody is gonna hit as hard as life. But it ain't about how hard ya hit. It's about how hard you can get it and keep moving forward. How much you can take and keep moving forward. That's how winning is done!"*
>
> – Rocky Balboa, *Rocky IV*

In this thesis we presented several methods to dealing with nonlinearities in audio signals, as well as some approaches to the problem of low frequency decaying pulses:

- In Chapter 7 two methods to deal with nonlinear distortions with memory were presented, one based on a polynomial approximation for the inverse of the nonlinearity, and other based on a piecewise linear approximation.
- Chapter 8 contains preliminary studies on a method to treat nonlinear distortions with memory, being modeled by a Hammerstein system, more adequate to deal with real distortions found in audio signals.
- Chapter 9 contains the results from the investigations presented in Chapters 7 and 8, and publications derived from this framework are [6, 7].
- The third part of the thesis contains a treatment of memoryless nonlinearities from a more recent viewpoint: we abandon the AR model for the underlying original signal and suppose that it is sparse in the DCT domain. In Chapter 10 this approach is tackled from a deterministic viewpoint, whereas in Chapter 11 the sparsity in the DCT domain is modeled in a Bayesian context.
- The thesis is closed with a detailed treatment of the low-frequency decaying pulse problem: in Chapter 12 the pulse is modeled in a parametric way, and

in order to estimate its tail it is necessary to estimate some parameters (this approach was published in [5]); and in Chapter 13 a more modern and elegant treatment is proposed, modeling the tail via a Gaussian process and considering an initialization procedure for some variables, critical to the convergence of the Gibbs sampler.

Some possibilities for future works were largely explained and motivated along these chapters, but we recall them briefly here and propose other possibilities beyond the subjects studied until now, as well as collaborations with researchers from other areas where statistical methods can be also employed.

14.1 Nonlinear distortions + AR model

- Parameterize the filters of the AR model and in the Hammerstein model via their poles, instead of their coefficients.
- Approximate the static memoryless nonlinearity via splines instead of polynomials or piecewise linear functions; another possibility is to model the static memoryless nonlinearity via a Gaussian process.
- Implement a sampling scheme able to deal with multi-modal distributions, for dealing with nonlinear distortions with memory.
- Adapt the idea presented in [82] to nonlinearly distorted audio signals.
- Turn the method robust to noise, correctly estimating the coefficients even in this scenario.
- Introduce psychoacoustical elements in the restoration procedure.
- Compare with previous works.

14.2 Nonlinear distortions + sparsity

The main future work of Chapter 10 is to generalize the proposed method to deal with nonlinear distortions with memory. We recall now some future works related to the content of Chapter 11:

- Explore other prior distributions tailored to model sparsity of the original undistorted signal.
- Reformulate the model in order to allow some frequency-depending penalty, since lower frequencies are more likely to be present within an audio signal than the higher ones.

- Incorporate noise in the model and thus allow for an estimation of the noisy signal, being the sparsity context more adequate for this task than the AR one.
- Increase computational efficiency by not computing some important quantities in a recursive way (recall the remark at the end of Section 11.5).
- Explore more deeply the information available in the posterior distribution.
- Incorporate memory within the model.

14.3 Low-frequency decaying pulse

- Treat the problem of detecting an initial discontinuity in the context of Statistical Decision Theory.
- Implement a more efficient interpolation scheme for the time samples in the initial discontinuity.
- Estimate the AR model parameters along the procedure, in order to make the interpolation more efficient.

14.4 Other audio restoration problems

During the year of 2015, I co-supervised the course completion assignment of Luís Felipe Velloso de Carvalho [111], where sinusoidal analysis methods were employed to restore audio signals degraded by speed variations during its playback. The main idea of the proposed algorithm is to detect and track the spectral peaks in order to find the main frequency lines of the signal and then obtain a global curve, representing the possible speed variations within the degraded signal. An algorithm which realizes time-varying resampling is then used to correct such deviations. Good results were reported even in real distorted signals, but the presence of background noise impairs the efficiency of the method, mainly because the frequency lines are misidentified in this context.

In [112] (also available online at the author's webpage `http://bayes.wustl.edu/`) is proposed a Bayesian technique to estimate the spectrum of a signal, and it is argued that this procedure is much more efficient than computing the FFT, if the signal is embedded in white noise. We expect that if the estimation of spectral peaks was done via this Bayesian method, the posterior restoration of audio signals degraded by speed variations will be much more effective. Preliminary tests indicated that this Bayesian technique is able

to correctly estimate the main frequencies present in audio signals, even when corrupted with additive white noise.

14.5 Applications of Statistics techniques to other fields

The amount of Statistics I learned during my D.Sc. certainly allows me to collaborate with researchers from other areas wanting to use statistical (mainly Bayesian) computational methods in their research. A possible future collaboration is with Leonardo Duarte from UNICAMP, employing MCMC methods in the estimation of nonlinearities in chemical sensor analysis. Another possibility is collaborate with some friends from the Institute of Physics of UFRJ working with quantum optics, being the use of statistical methods very important to validate the results of experiments. Currently I am also participating in a series of lectures with some friends and professors from the Institute of Mathematics of UFRJ, aiming at learning Gaussian processes in more detail.

Finally, although not exactly an application of Statistics, I wish to publish my M.Sc. monograph [1] as a book, as suggested by the examiners.

Bibliography

[1] CARVALHO, H. T. *Uma Introdução às Singularidades em Relatividade Geral*. M.Sc. dissertation, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil, 2013.

[2] STIGLER, S. M. "Who Discovered Bayes' Theorem?" *The American Statistician*, v. 37, n. 4, pp. 290–296, 1983.

[3] "Researchers Play Tune Recorded Before Edison". 2008. http://www.nytimes.com/2008/03/27/arts/27soun.html.

[4] TOMAZELI DUARTE, L., SUYAMA, R., ATTUX, R., et al. "A sparsity-based method for blind compensation of a memoryless nonlinear distortion: Application to ion-selective electrodes", *IEEE Sensors Journal*, v. 15, n. 4, pp. 2054–2061, 2014.

[5] CARVALHO, H. T., ÁVILA, F. R., BISCAINHO, L. W. P. "A Bayesian procedure for restoration of audio signals degraded by low-frequency pulses". In: *Anais do 12o. Congresso de Engenharia de Áudio da AES-Brasil*, pp. 47–54, São Paulo, Brazil, May 2014. AES-Brasil.

[6] CARVALHO, H. T., ÁVILA, F. R., BISCAINHO, L. W. P. "Bayesian suppression of memoryless nonlinear audio distortion". In: *Proceedings of the 23rd. European Signal Processing Conference (EUSIPCO)*, pp. 1063–1067, Nice, France, September 2015. EURASIP.

[7] ÁVILA, F. R., CARVALHO, H. T., BISCAINHO, L. W. P. "Bayesian blind identification of nonlinear distortion with memory for audio applications", *IEEE Signal Processing Letters*, v. 23, n. 4, pp. 414–418, 2016.

[8] HACKING, I. *The Emergence of Probability: A Philosophical Study of Early Ideas About Probability Induction and Statistical Inference*. 2 ed. Cambridge, UK, Cambridge University Press, 2006.

[9] STIGLER, S. M. *The History of Statistics: The Measurement of Uncertainty before 1900*. 1 ed. Massachusetts, USA, Belknap Press, 1990.

[10] DEGROOT, M. H., SCHERVISH, M. J. *Probability and Statistics*. 4 ed. Upper Saddle River, USA, Pearson, 2011.

[11] WASSERMAN, L. *All of Statistics: A Concise Course in Statistical Inference*. 1 ed. New York, USA, Springer, 2004.

[12] DEVLIN, K. *The Unfinished Game: Pascal, Fermat and the Seventeenth-Century Letter that Made the World Modern*. 1 ed. New York, USA, Basic Books, 2008.

[13] JAYNES, E. T., BRETTHORST, G. L. *Probability Theory: The Logic of Science*. 1 ed. Cambridge, UK, Cambridge University Press, 2003.

[14] KOLMOGOROV, A. N. *Foundations of the Theory of Probability*. 2 ed. New York, USA, Chelsea Publishing Company, 1956.

[15] BILLINGSLEY, P. *Probability and Measure*. 3 ed. New Jersey, USA, Wiley-Interscience, 1995.

[16] CHUNG, K. L. *A Course in Probability Theory*. 2 ed. Cambridge, USA, Academic Press, 2000.

[17] KAY, S. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. 1 ed. Upper Saddle River, USA, Prentice Hall, 1993.

[18] NOCEDAL, J., WRIGHT, S. *Numerical Optimization*. New York, USA, Springer, 2006.

[19] MCGRAYNE, S. B. *The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, and Emerged Triumphant from Two Centuries of Controversy*. 1 ed. New Haven, USA, Yale University Press, 2012.

[20] ECKHARDT, R. "Stan Ulam, John von Neumann, and the Monte Carlo method", *Los Alamos Science*, v. Special Issue: Stanislaw Ulam 1909–1984, n. 15, pp. 131–143, 1987.

[21] METROPOLIS, N. "The beginning of the Monte Carlo Method", *Los Alamos Science*, v. Special Issue: Stanislaw Ulam 1909–1984, n. 15, pp. 125–130, 1987.

[22] ROBERT, C., CASELLA, G. *Introducing Monte Carlo Methods with R*. 1 ed. New York, USA, Springer, 2009.

[23] KNUTH, D. E. *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. 3 ed. Boston, USA, Addison-Wesley Professional, 2014.

[24] RANDOM.ORG. "Random.org". http://www.random.org/.

[25] MARKOV, A. A. "Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga", *Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitet*, v. 2, n. 15, pp. 135–156, 1906.

[26] KINGMAN, J. F. C. "The first Erlang century – and the next", *Queuing Systems*, v. 63, n. 3, pp. 3–12, 2009.

[27] ROBERT, C., CASELLA, G. *Monte Carlo Statistical Methods*. 2 ed. New York, USA, Springer, 2005.

[28] METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., et al. "Equation of state calculations by fast computing machines", *The Journal of Chemical Physics*, v. 21, pp. 1087–1092, 1953.

[29] HASTINGS, W. K. "Monte Carlo sampling methods using Markov Chains and their applications", *Biometrika*, v. 57, n. 1, pp. 97–109, 1970.

[30] GIVENS, G. H., HOETING, J. A. *Computational Statistics*. 2 ed. Hoboken, USA, Wiley, 2012.

[31] GEMAN, S., GEMAN, D. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 6, n. 6, pp. 721–741, 1984.

[32] GELFAND, A. E., SMITH, A. F. M. "Sampling-Based Approaches to Calculating Marginal Densities", *Journal of the American Statistical Association*, v. 85, n. 410, pp. 398–409, 1990.

[33] DIACONIS, P., SALOFF-COSTE, L. "What do we know about the Metropolis algorithm?" *Journal of Computer and System Sciences*, v. 57, n. 1, pp. 20–36, 1998.

[34] COWLES, M. K., CARLIN, B. P. "Markov Chain Monte Carlo convergence diagnostics: A comparative review", *Journal of the American Statistical Association*, v. 91, n. 434, pp. 883–904, 1996.

[35] COVER, T. M., THOMAS, J. A. *Elements of Information Theory*. 2 ed. New Jersey, USA, Wiley-Interscience, 2006.

[36] BOX, G. E. P., JENKINS, G. M., REINSEL, G. C. *Time Series Analysis: Forecasting and Control*. 3 ed. Upper Saddle River, USA, Prentice-Hall, 1994.

[37] GODSILL, S. J., RAYNER, P. J. W. *Digital Audio Restoration - A Statistical Model Based Approach*. 1 ed. Cambridge, UK, Cambridge University Press, 1998.

[38] RAJAN, J., RAYNER, P. J. W., GODSILL, S. J. "Bayesian approach to parameter estimation and interpolation of time-varying autoregressive processes using the Gibbs sampler", *IEEE Proceedings on Vision, Image and Signal Processing*, v. 144, n. 4, pp. 249–256, 1997.

[39] HANS, M., SCHAFER, R. W. "Lossless compression of digital audio", *IEEE Signal Processing Magazine*, v. 18, n. 3, pp. 21–32, 2001.

[40] PRIESTLEY, M. B. *Spectral Analysis and Time Series, Vols. I and II*. 1 ed. San Diego, USA, Academic Press, 1981.

[41] PEEBLES, P. Z. *Probability, Random Variables and Random Signal Principles*. 2 ed. New York, USA, McGraw Hill, 1987.

[42] WIKIPEDIA. "Wiener-Khinchin Theorem". . http://en.wikipedia.org/wiki/Wiener-Khinchin_theorem.

[43] OPPENHEIM, A. V., SCHAFER, R. W. *Discrete-Time Signal Processing*. 3 ed. Upper Saddle River, USA, Prentice Hall, 2009.

[44] YULE, G. U. "On a method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot numbers", *Philosophical Transactions of the Royal Society of London, series A*, v. 226, pp. 267–298, 1927.

[45] ÁVILA, F. R. *Métodos Bayesianos para Restauração de Sinais de Áudio com Distorções Não-lineares*. D.Sc. thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil, 2012.

[46] WIKIPEDIA. "Distortion". . http://en.wikipedia.org/wiki/Distortion/.

[47] MOORE, B. C. J., TAN, C. T., ZACHAROV, N., et al. "Measuring and predicting the perceived quality of music and speech subjected to combined linear and nonlinear distortion", *Journal of the Audio Engineering Society*, v. 52, n. 12, pp. 1228–1244, 2004.

[48] ZÖLZER, U. *DAFX: Digital Audio Effects*. 2 ed. Hoboken, USA, Wiley, 2011.

[49] WIKIPEDIA. "In the Court of the Crimson King". . http://en.wikipedia.org/wiki/In_the_Court_of_the_Crimson_King/.

[50] DOYLE III, F. J., PEARSON, R. K., OGUNNAIKE, B. A. *Identification and Control Using Volterra Models*. 1 ed. New York, USA, Springer, 2002.

[51] VOLTERRA, V. *Theory of Functionals and of Integral and Integro-Differential Equations*. 1 ed. Mineola, USA, Dover Publications, 2005.

[52] ARFKEN, G. *Mathematical Methods for Physicists*. 1 ed. San Diego, USA, Academic Press, Inc., 1985.

[53] CODDINGTON, E. A., LEVINSON, N. *Theory of Ordinary Differential Equations*. 1 ed. Malabar, USA, Krieger Publishing Company, 1984.

[54] TRICOMI, F. G. *Integral Equations*. 1 ed. Mineola, USA, Dover Publications, 1985.

[55] WIENER, N. *Response of a Non-Linear Device to Noise*. In: Report R-129, Massachusetts Institute of Technology, Radiation Laboratory, Cambridge, USA, 1942.

[56] IKEHARA, S. *A Method of Wiener in a Nonlinear Circuit*. In: Report 217, Massachusetts Institute of Technology, Research Laboratory of Eletronics, Cambridge, USA, 1951.

[57] WHITE, S. A. "Restoration of nonlinearly distorted audio by histogram equalization", *Journal of the Audio Engineering Society*, v. 30, n. 11, pp. 828–832, 1982.

[58] PREIS, D., H, P. "Restoration of nonlinearly distorted magnetic recordings", *Journal of the Audio Engineering Society*, v. 32, n. 1/2, pp. 26–30, 1984.

[59] KLIPPEL, W. "Compensation for Nonlinear Distortion of Horn Loudspeakers by Digital Signal Processing", *Journal of the Audio Engineering Society*, v. 44, n. 11, pp. 964–972, 1996.

[60] BAKÓ, T. B., BANK, B., DABÓCZI, T. "Restoration of nonlinearly distorted audio with the application to old motion-pictures". In: *Proceedings of the AES 20th International Conference*, pp. 191–198, Budapest, Hungary, October 2001. AES.

[61] TROUGHTON, P. T. *Simulation Methods for Linear and Nonlinear Time Series Models with Application to Distorted Audio Signals*. Ph.D. thesis, University of Cambridge, Cambridge, UK, 1999.

[62] TROUGHTON, P. T., GODSILL, S. J. "MCMC methods for restoration of nonlinearly distorted autoregressive signals", *Signal Processing*, v. 81, n. 1, pp. 83–97, 2001.

[63] TROUGHTON, P. T., GODSILL, S. J. "MCMC methods for restoration of nonlinearly distorted autoregressive signals". In: *Proceedings of the 9th European Signal Processing Conference (EUSIPCO 1998)*, pp. 1–4, Rhodes, Greece, September 1998. EURASIP.

[64] YEH, D. T., BANK, B., KARJALAINEN, M. "Nonlinear modeling of a guitar loudspeaker cabinet". In: *Proceedings of the 11th. International Conference on Digital Audio Effects (DAFx-08)*, Espoo, Finland, September 2008.

[65] KAIZER, A. J. M. "Modeling of the nonlinear response of an electrodynamic loudspeaker by a Volterra series expansion", *Journal of the Audio Engineering Society*, v. 35, n. 6, pp. 421–433, 1987.

[66] MALEK, J. "Blind compensation of memoryless nonlinear distortions in sparse signals". In: *Proceedings of the 21st European Signal Processing Conference (EUSIPCO 2013)*, pp. 1–5, Marrakech, Morroco, September 2013. EURASIP.

[67] ÁVILA, F. R., BISCAINHO, L. W. P. "ML estimation of memoryless nonlinear distortions in audio signals". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4493–4497, Florence, Italy, May 2014. IEEE.

[68] LAPLACE, P.-S. "Memoir on the probability of the causes of events", *Statistical Science*, v. 1, n. 3, pp. 364–378, 1986.

[69] BISHOP, C. M. *Pattern Recognition and Machine Learning*. New York, USA, Springer, 2007.

[70] PICARD, G., CAPPÉ, O. "Blind identification of Hammerstein nonlinear distortion models". In: *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2003)*, pp. 17–20, New Paltz, USA, October 2003. IEEE.

[71] HAYES, M. H. *Statistical Digital Signal Processing and Modeling*. 1 ed. New Jersey, EUA, Wiley, 1996.

[72] CARVALHO, H. T. "My personal webpage". http://www.smt.ufrj.br/∼hugo.carvalho.

[73] SPIEGELHALTER, D. J., BEST, N. G., CARLIN, B. P., et al. "Bayesian measures of model complexity and fit", *Journal of the Royal Statistical Society: Series B*, v. 64, n. 2, pp. 583–639, 2002.

[74] VAN DER LINDE, A. "DIC in variable selection", *Statistica Neerlandica*, v. 59, n. 1, pp. 45–56, 2005.

[75] GAMERMAN, D., LOPES, H. F. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. 2 ed. Boca Raton, USA, Chapman & Hall/CRC, 2006.

[76] TREFETHEN, L. N., BAU III, D. *Numerical Linear Algebra*. 1 ed. Philadelphia, USA, SIAM: Society for Industrial and Applied Mathematics, 1997.

[77] HUERTA, G., WEST, M. "Priors and component structures in autoregressive time series models", *Journal of the Royal Statistical Society: Series B*, v. 61, n. 4, pp. 881–899, 1999.

[78] PRADO, R., LOPES, H. F. "Sequential parameter learning and filtering in structured autoregressive state-space models", *Statistics and Computing*, v. 32, n. 1, pp. 43–57, 1999.

[79] RASMUSSEN, C. E., WILLIAMS, C. K. I. *Gaussian Processes for Machine Learning*. 1 ed. Cambridge, USA, MIT Press, 2006.

[80] LINDSTEN, F., SCHÖN, T. B., JORDAN, M. I. "Bayesian semiparametric Wiener system identification", *Automatica*, v. 49, n. 7, pp. 2053–2063, 2013.

[81] OLIVER, D. S. "Metropolized randomized maximum likelihood for sampling from multimodal distributions", *arXiv.org/abs/1507.08563*, 2015.

[82] JANSSEN, A. J. E. M., VELDHUIS, R. N. J., VRIES, L. B. "Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 34, n. 2, pp. 317–330, 1986.

[83] WOLFE, P. J., GODSILL, S. J., NG, W.-J. "Bayesian variable selection and regularization for time–frequency surface estimation", *Journal of the Royal Statistical Society: Series B*, v. 66, n. 3, pp. 575–589, 2004.

[84] WOLFE, P. J., GODSILL, S. J. "Perceptually motivated approaches to music restoration", *Journal of New Music Research*, v. 30, n. 1, pp. 83–92, 2001.

[85] DEFRAENE, B., VAN WATERSCHOOT, T., FERREAU, H., et al. "Real-time perception-based clipping of audio signals using convex optimization", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 20, n. 10, pp. 2657–2671, 2012.

[86] DEFRAENE, B., ANSOUR, N., DE HERTOGH, S., et al. "Declipping of audio signals using perceptual compressed sensing", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 21, n. 12, pp. 2627–2637, 2013.

[87] DEFRAENE, B., VAN WATERSCHOOT, T., DIEHL, M., et al. "Embedded-optimization-based loudspeaker precompensation Using a Hammerstein loudspeaker model", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. 22, n. 11, pp. 1648–1659, 2014.

[88] AHMED, N., NATARAJAN, T., RAO, K. R. "Discrete cosine transform", *IEEE Transactions on Computers*, v. C-23, n. 1, pp. 90–93, 1974.

[89] RAO, K. R., YIP, P. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. 1 ed. San Diego, USA, Academic Press, 1990.

[90] FOUCART, S., RAUHUT, H. *A Mathematical Introduction to Compressive Sensing*. 1 ed. Basel, Switzerland, Birkhäuser, 2013.

[91] NATARAJAN, B. K. "Sparse approximate solutions to linear systems", *SIAM Journal on Computing*, v. 24, n. 2, pp. 227–234, 1995.

[92] MOHIMANI, H., BABAIE-ZADEH, M., JUTTEN, C. "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm", *IEEE Transactions on Signal Processing*, v. 57, n. 1, pp. 289–301, 2009.

[93] BABACAN, S. D., MOLINA, R., KATSAGGELOS, A. K. "Bayesian compressive sensing using Laplace priors", *IEEE Transactions on Image Processing*, v. 19, n. 1, pp. 53–63, 2010.

[94] TIBSHIRANI, R. "Regression Shrinkage and Selection via the Lasso", *Journal of the Royal Statistical Society, Series B*, v. 58, n. 1, pp. 267–288, 1996.

[95] HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2 ed. New York, USA, Springer, 2009.

[96] BARNDORFF-NIELSEN, O. E. "Normal Inverse Gaussian Distributions and Stochastic Volatility Modelling", *Scandinavian Journal of Statistics*, v. 24, n. 1, pp. 1–13, 1997.

[97] MOHAMMAD-DJAFARI, A. "Bayesian approach with prior models which enforce sparsity in signal and image processing", *EURASIP Journal on Advances in Signal Processing*, v. 52, 2012.

[98] ÁVILA, F. R. *Algoritmos Baseados em Modelos Bayesianos para Restauração de Sinais de Áudio*. M.Sc. dissertation, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil, 2008.

[99] VASEGHI, S. V. *Algorithms for Restoration of Archived Gramophone Recordings*. Ph.D. thesis, University of Cambridge, Cambridge, UK, 1988.

[100] VASEGHI, S. V., FRAYLING-CORK, R. "Restoration of old gramophone recordings", *Journal of the Audio Engineering Society*, v. 40, n. 10, pp. 791–801, 1991.

[101] GODSILL, S. J., TAN, C. H. "Removal of low frequency transient noise from old recordings using model-based signal separation techniques". In: *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 1997)*, New Paltz, USA, October 1997. IEEE.

[102] ESQUEF, P. A. A., BISCAINHO, L. W. P., VÄLIMÄKI, V. "An efficient algorithm for the restoration of audio signals corrupted with low-frequency pulses", *Journal of the Audio Engineering Society*, v. 51, n. 6, pp. 502–517, 2003.

[103] ESQUEF, P. A. A., WELTER, G. S. "Audio se-thumping using Huang's Empirical Mode Decomposition". In: *Proceedings of the 14th. International Conference on Digital Audio Effects (DAFx-11)*, Paris, France, September 2011.

[104] DE CATALUNYA, B. "Uns incunables del sonor - La col·lecció Regordosa-Turull de cilindres de cera". http://www.bnc.cat/Exposicions/Uns-incunables-del-sonor/Continguts-de-l-exposicio.

[105] MACKAY, D. J. C. *Information Theory, Inference and Learning Algorithms*. 1 ed. Cambridge, UK, Cambridge University Press, 2003.

[106] WILSON, A. G. *Covariance Kernels for Fast Automatic Pattern Discovery and Extrapolation with Gaussian Processes*. D.Sc. thesis, University of Cambridge, Cambridge, UK, 2014.

[107] ESQUEF, P. A. A., BISCAINHO, L. W. P., DINIZ, P. S. R. "Detecção de pulsos longos em sinais de áudio". In: *Proceedings of the XVII Brazilian Symposium on Telecommunications (SBrT)*, pp. 191–196, Vila Velha, Brazli, September 1999. SBrT.

[108] JAIN, A. K. *Fundamentals of Digital Image Processing*. 1 ed. Upper Saddle River, USA, Pearson, 1988.

[109] ARIAS-CASTRO, E., DONOHO, D. L. "Does median filtering truly preserve edges better than linear filtering?" *The Annals of Statistics*, v. 37, n. 3, pp. 1172–1206, 2009.

[110] ESQUEF, P. A. A., BISCAINHO, L. W. P. "An efficient model-based multirate method for reconstruction of audio signals across long gaps", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 14, n. 4, pp. 1391–1400, 2006.

[111] DE CARVALHO, L. F. V. *Correção de Desvios na Velocidade de Reprodução de Gravações de Música*. B.Sc. monography, POLI/UFRJ, Rio de Janeiro, Brazil, 2015.

[112] BRETTHORST, G. L. *Bayesian Spectrum Analysis and Parameter Estimation*. 1 ed. New York, USA, Springer, 1988.