Nonlinear System Modeling and Identification of Loudspeakers

A Dissertation Proposal Presented

by

Pascal Brunet

to

The Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

for the Degree of

Doctor of Philosophy

in

Electrical Engineering

Northeastern University

Boston, Massachusetts

April 2014

To my wife Lois and my children Olivia, Gabriel, Muriel, Lauren

Nonlinear System Modeling and Identification of Loudspeakers

by

Pascal Brunet

ABSTRACT OF DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate School of Engineering
Northeastern University, April 2014

Abstract

This dissertation considers modeling and identification of nonlinear systems pertinent to loudspeakers with nonlinear distortion effects. It is well known that when loudspeakers are driven at high playback levels the nonlinear characteristics of these speakers become a major source of sound degradations. Nonlinear distortion not only diminishes listening pleasure but more importantly impairs speech intelligibility. Consequently it is essential to find a good model that matches to the loudspeaker response for the purpose of predicting and preventing the nonlinear distortion. This becomes particularly important for the purpose of improving sound quality of mobile phones. This report analyzes the loudspeaker operation and provides nonlinear modeling techniques that can reliably be used for its identification process. Frequency domain and state-space modeling are considered and emphasis is given towards model that mix polynomial nonlinear state-space models and fractional order state-space models.

Glossary

Notations

- x: state vector $\in \mathbb{R}^{n_a}$
- y: system output $\in \mathbb{R}$
- u: system input $\in \mathbb{R}$
- n: noise (or index, depending on context)
- t: time variable (or index, depending on context)
- A: state matrix $\in \mathbb{R}^{n_a \times n_a}$
- B: state-input matrix $\in \mathbb{R}^{n_a \times 1}$
- C: state-input matrix $\in \mathbb{R}^{1 \times n_a}$
- D: output-input matrix $\in \mathbb{R}$
- E: polynomial weight matrix (state equation) $\in \mathbb{R}^{n_a \times n_p}$
- F: polynomial weight matrix (output equation) $\in \mathbb{R}^{1 \times n_q}$
- p: vector of monomials (PNLSS equations) $\in \mathbb{R}^{n_p \times 1}$
- d: nonlinearity order $\in \mathbb{N}$

- $\zeta_{(d)}$: vector of ζ monomials of degree d
- $\zeta_{\{d\}}$: vector of ζ monomials of degree 2 to d
- $diag(M)$: diagonal matrix formed from the diagonal elements of M
- $vec(M)$: column vector obtained by stacking all the columns of M on top of each other, from left to right
- M^{+}: Moore-Penrose pseudo-inverse of matrix M
- z^{*}: complex conjugate of z

Acronyms

- BLA: Best Linear Approximation
- CT: Continuous Time
- DT: Discrete Time
- DFT: Discrete Fourier Transform
- FRF: Frequency Response Function
- LS: Least-squares
- NL: NonLinear
- ODE: Ordinary Differential Equation
- PNLSS: Polynomial Nonlinear State-Space
- SVD: Singular Value Decomposition
- RMSE: Relative Mean Square Error

Acknowledgments

I wish to express my gratitude and indebtedness to Professor Bahram Shafai for his helpful guidance, encouragement and stimulating discussions during the preparation of this work and for his fruitful comments and continuous follow up during its development. His positive attitude, kindness and cheerfulness were of great support to me.
I thank Professors Dana Brooks and Deniz Erdogmus for serving in my dissertation committee, expressing interest in my work and for their pertinent suggestions.
I want also to thank Dr. Alexander Voishvillo for his help and Dr. Wolfgang Klippel for his kind encouragements.

Pascal Brunet

Northeastern University

April 2014

Contents

Chapter 1

Introduction

Acoustic transducers are part of our everyday life, and we use them intensively throughout the day using our cellphones, listening to the radio in our car, looking at the TV or playing games on computer at night. In all cases, sound distortion is present and has negative impact on the sound quality, diminishing listening pleasure and, worse, speech intelligibility. In some cases, texting is the only way to get our message across. In particular, cellphones, teleconference systems, PC systems use small loudspeakers driven at high-amplitude to get enough sound level greatly increasing nonlinear distortion. It is particularly critical when it comes to hands-free or speaker-phone situations. So, nonlinear distortion becomes increasingly prevalent, and yet there is still no satisfactory model for this phenomenon.

The study of loudspeaker and its characterization based on sine response remained common approach for many years. Sine sweep, step by step or continuous, have been used to measure frequency response and distortion. For non-linear behavior characterization single tone is used to measure harmonic distortion and two tones are used for intermodulation and difference distortion. Many different and sophisticated variations of these basic measurements are used, but sine response doesn't predict reliably the music or speech quality. Multitone and random noise excitations, along

with coherence analysis have been introduced [1] but have not gained in popularity. The different distortion measurements (harmonic distortion, intermodulation, multitone distortion, non-coherent power) are not related to each other by an underlying model, and remain purely symptomatic.

It is natural to think that the loudspeaker industry could benefit from the modern techniques of nonlinear system identification to obtain a comprehensive and accurate model for diagnosis, quality control, simulation, prediction and ultimately, linearization. Following the advancements in nonlinear system theory, during the last 30 years, many attempts have been made in the identification and linearization of loudspeaker [2–12]. However due to the wide range of audio frequencies (20 Hz to 20kHz), the complexity of the device and high human ear sensitivity, the loudspeaker identification and linearization remain an elusive goal.

This study reviews recent developments in the domain of loudspeaker identification and explores new possibilities to improve modeling that is better match to the loudspeaker response. First we present the loudspeaker operation principles and the major causes of distortion, then we explore the successive modeling approaches that have been investigated in the last decades. Finally we provide new directions of research in the frequency domain and propose two techniques based on state-space for modeling of loudspeaker which can effectively be used in identification process.

The first one is a polynomial extension of the state-space model that provides a generic and comprehensive approach to the representation of Lipschitz nonlinear systems. A two steps identification method is proposed. Simulation and experimental results are presented and discussed.

The second one is a fractional order state-space modeling. We show that fractional order differential equations lead to a simpler and more accurate modeling of the loudspeaker dynamics. The theory of fractional calculus pertinent to our study is

explained. An identification technique adapted from the previous one is described. Experimental results are shown that confirm the validity of our approach.

As the identification process can produce models with excessive order and unstable poles a section describes different methods for model reduction and stabilization.

Finally a section shows how the two approaches (polynomial and fractional order) can be combined to obtain a comprehensive (linear and nonlinear) modeling of the loudspeaker.

Chapter 2

Problem Description

In this chapter, we present the operation principle of common loudspeaker and a first approach of state-space modeling. An analysis section shows the limit of that approach in presence of nonlinearities.

2.1 Loudspeaker Mechanism

The most common type of driver is electro-dynamic. The driving part, the motor, is a moving coil into a static magnetic field. The audio signal goes through the coil and creates a variable magnetic field that interact with fixed magnets and generate a mechanical force that is roughly proportional to the electrical current. The acoustic radiation is insured by a lightweight cone (diaphragm) attached to the coil. An elastic suspension maintains the coil and the attached cone in place into the frame ("basket"). The cone is also mechanically connected to the basket by an elastic suspension called surround (see fig. 2.1). Designing a driver combines acoustic, mechanical, electrical and material science. A simplified linear model based on lumped parameters describes the loudspeaker mechanism at low frequencies and small amplitudes. It is composed

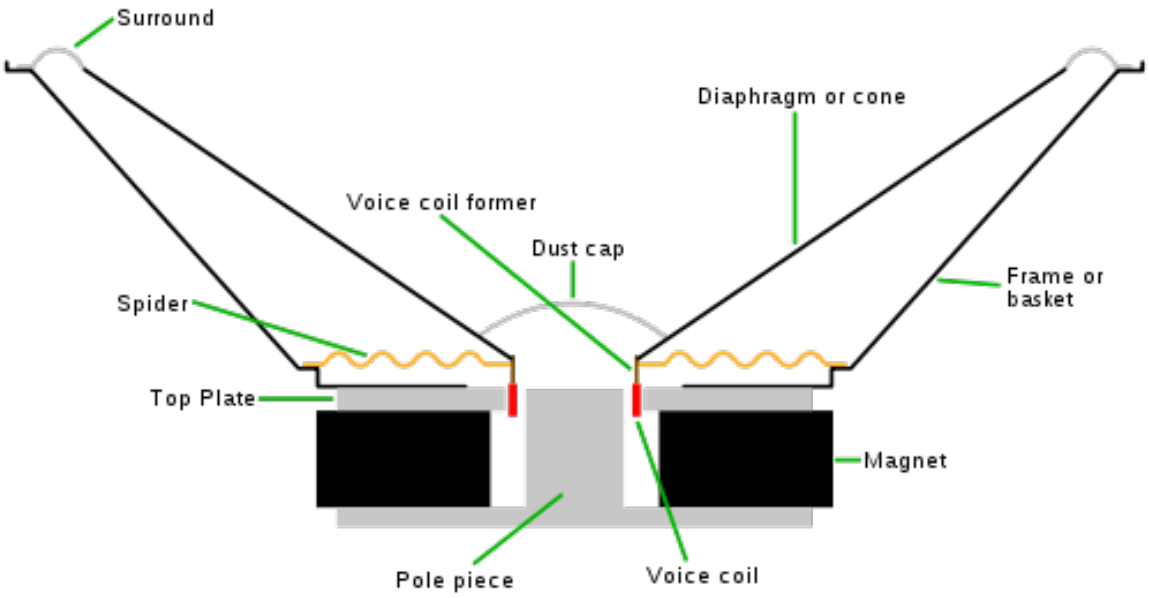

Figure 2.1: Loudspeaker mechanism

of two differential equations.

$$u(t) = Ri(t) + L\frac{di}{dt} + Bl\frac{dx}{dt} \tag{2.1}$$

$$Bli(t) = m\frac{d^2x}{dt^2} + r\frac{dx}{dt} + kx(t) \tag{2.2}$$

where u(t) is the input voltage, i(t) the current, x(t) the cone displacement and R, L, Bl, m, r, k are electromechanical parameters of the loudspeaker. It is important to note that the force factor Bl, the voice coil inductance L and the stiffness k are nonlinear function of the displacement x. Therefore non-linearity is intrinsic to the driver's principle of operation. Beside the changing parameters just mentioned, there is a wide variety of non-linear behaviors [13]. For example, at high frequencies the cone and dome no longer behave as rigid bodies. They exhibit breakup modes and eventually the vibrations become nonlinear. Another distortion inherent to the fundamental principle of operation is the Doppler effect due to the fact that the sound is emitted from the diaphragm which is a moving source.

2.2 State-Space Modeling

In this subsection we introduce the state-space representation of the loudspeaker. By inspection of equations (2.1) and (2.2), the following state vector can be defined:

$$z(t) \triangleq (x(t), \frac{dx}{dt}, i(t))^T = (z_1(t), z_2(t), z_3(t))$$

Hence (2.1) and (2.2) can equivalently be written as:

$$\dot{z_1} = z_2$$

$$u = Blz_2 + L\dot{z_3} + Rz_3$$

$$Blz_3 = m\dot{z_2} + rz_2 + kz_1$$

where:

$$\dot{z} \triangleq \frac{dz}{dt}$$

Simple algebraic manipulations leads to:

$$\begin{aligned}
\dot{z_1} &= z_2 \\
\dot{z_2} &= -\frac{k}{m}z_1 - \frac{r}{m}z_2 + \frac{Bl}{m}z_3 \\
\dot{z_3} &= -\frac{Bl}{L}z_2 - \frac{R}{L}z_3 + \frac{1}{L}u
\end{aligned}$$

which can be written in state-space notation as follows:

$$\dot{z} = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{k}{m} & -\frac{r}{m} & \frac{Bl}{m} \\ 0 & -\frac{Bl}{L} & -\frac{R}{L} \end{pmatrix} z + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{L} \end{pmatrix} u \triangleq Az + Bu \tag{2.3}$$

The output variable can be either the cone displacement $x(t)$ or the cone velocity $\frac{dx}{dt}$. If the cone velocity is chosen, the output equation is given by:

$$y(t) = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} z \triangleq Cz \tag{2.4}$$

The pair of equations (2.3) and (2.4) form the State-Space equations of the model described by (2.1) and (2.2). However the acoustic pressure produced at some distance of the loudspeaker is proportional to the cone acceleration and that has to be taken into account when considering a loudspeaker model identification based on acoustic measurements.

The Frequency Response Function (FRF) is obtained in a standard manner from the State-Space equations:

$$G(j\omega) = Y(j\omega)/U(j\omega) = C(j\omega I - A)^{-1}B$$

$j\omega G(j\omega)$ will then yield the acoustic frequency response with a scale factor . As an example, using the following parameters from a real loudspeaker [11]:

- m= 14.35 g
- r= 0.786 kg/s
- k= 1852 N/m
- Bl= 4.95 N/A
- L= 266 μH
- R= 3.3 Ohms

the acoustic frequency response shown in Fig. 2.2 is obtained. The FRF exhibits the overall characteristics typical of a loudspeaker which incorporates band-pass response with audio range 60-2kHz and +40dB/decade in the low range. However this is clearly a very simplified model as compared to the reality (Fig. 2.3) which shows the limitations of physical modeling of actual loudspeaker.

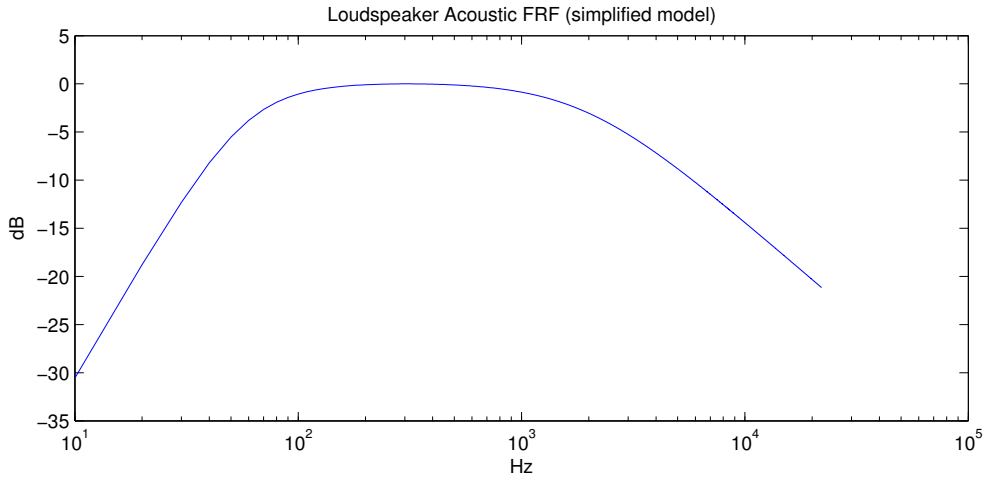

Figure 2.2: Loudspeaker Acoustic FRF (simple model).

2.3 Analysis

As we pointed out in the introduction, we are dealing with loudspeaker, not only for music and entertainment but also for communication. In the case of complex signals like speech or music, distortion sounds generally like a modulation noise that degrades the clarity of the signal. To demonstrate this effect, music has been played through a loudspeaker and the added non-linear distortion has been measured as the non-coherent power present in the acoustical signal [1]. Fig. 2.4 shows both the spectrum of the musical signal and the spectrum of the added distortion noise. To further analyze various modeling methodologies, we start with the simple state-space model introduced in the previous section and study the effect of its parameter variations. Due to the fact that parameters such as the force-factor Bl is not constant, rather it has the following nonlinear characteristic (Fig. 2.5), we perform a robustness analysis on the simplified model. Fig. 2.6 shows the response of simplified model of Fig. 2.2 with interval uncertainty on the parameter Bl. It can be shown that the envelope of the uncertainty captures the nonlinear uncertainty within the specified interval. At the same time we add a nonlinear term using a polynomial approximation to the linear

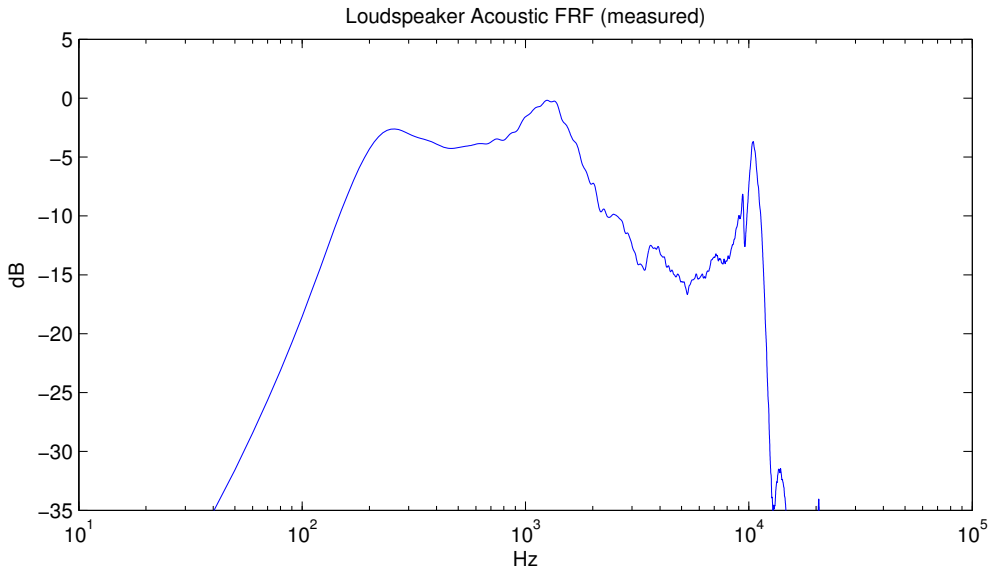

Figure 2.3: Loudspeaker Acoustic FRF (measured).

model (2.3) in order to show its effects on the simplified model. Comparing the exact response of the loudspeaker (Fig. 2.7) suggest that one needs a more sophisticated nonlinear state-space model to be used for system identification purposes. This will be discussed in more details in further section.

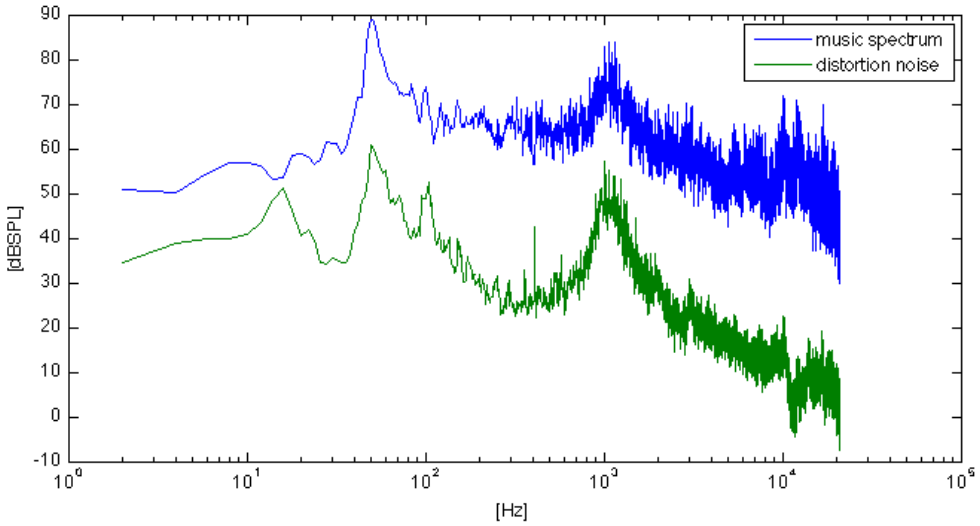

Figure 2.4: Music spectrum and Distortion Noise from a Loudspeaker.

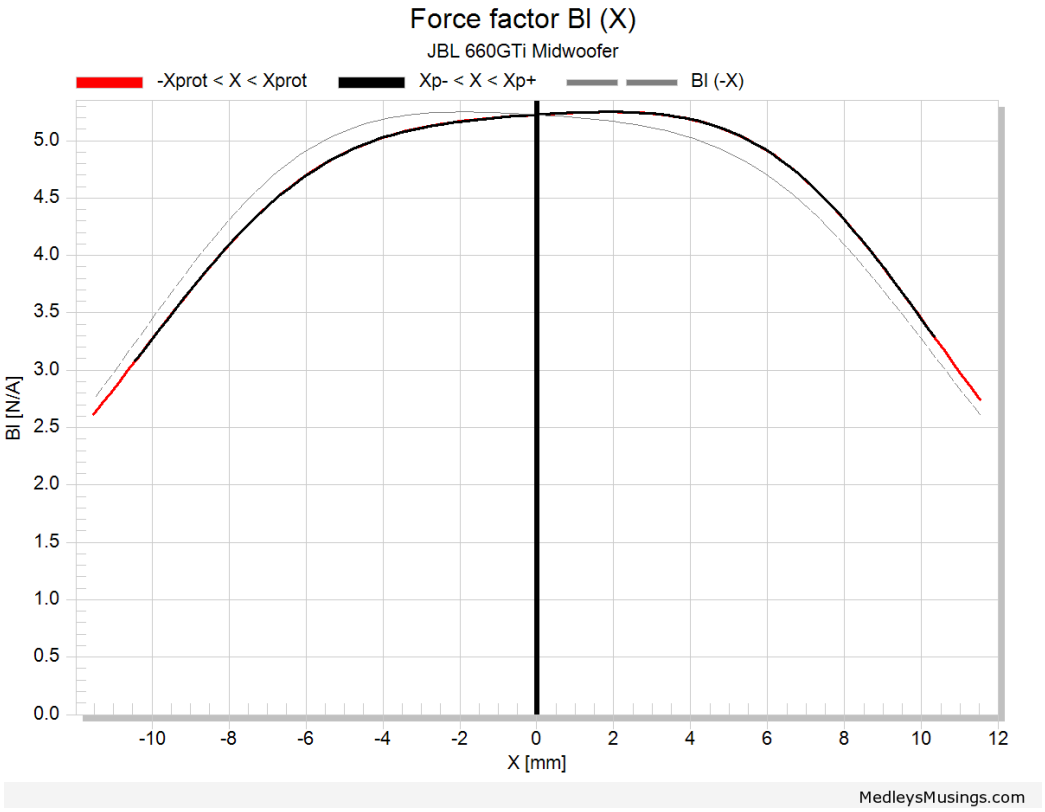

Figure 2.5: Force-factor Bl as a function of displacement x.

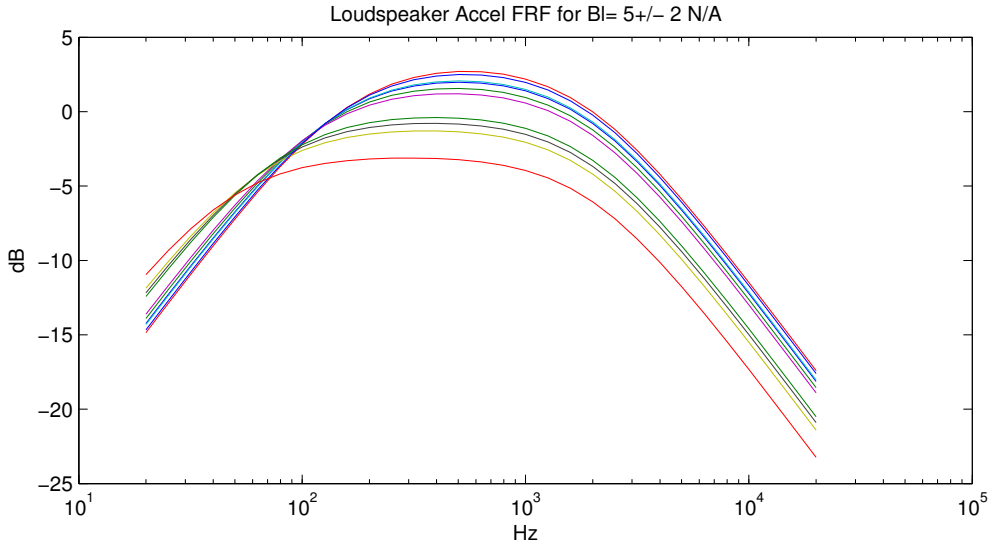

Figure 2.6: Loudspeaker Acoustic FRF (simple model) with Bl perturbation.

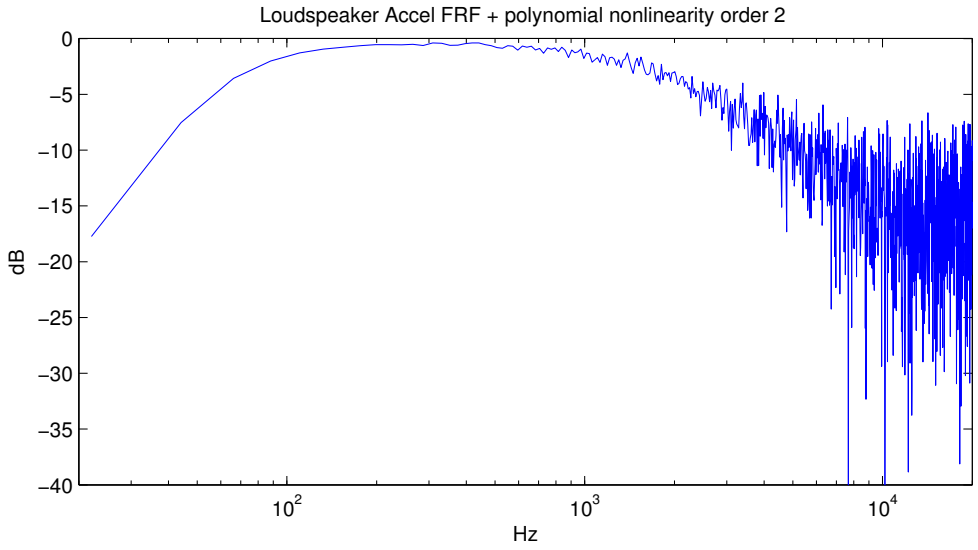

Figure 2.7: Loudspeaker Acoustic FRF (simple model) with added nonlinearity.

Chapter 3

Previous Approaches

This chapter reviews some classical methods of nonlinear modeling in a historical perspective.

3.1 White Box (1980's)

The first attempts of system identification applied to loudspeaker were based on the lumped model described by equations (2.1) and (2.2). That simple system identification delivers a first prediction of the mechanical behavior of the loudspeaker for low frequencies, and small signals. It was applicable up to the cone breakup frequency where the cone still behaves as a rigid piston. The measurement method is based on sine excitation and proceeds in two successive parts, involving added mass or loudspeaker enclosure [14–17].

In a seminal paper, A. J. M Kaizer, in 1987 [2], approximates the most prominent nonlinearities (force factor Bl(x), self-inductance L(x), stiffness k(x)) by polynomials, then expresses the response of the system to exponentials in term of Volterra series [18].

The output of the system is written as:

$$y_Q(t) = \sum_{n=1}^{Q} y_n(t)$$

with:

$$y_n(t) = \int \dots \int h_n(\tau_1, \dots \tau_n) u(t-\tau_1) \dots u(t-\tau_n) d\tau_1 \dots d\tau_n$$

Where u(t) is the input signal and $h_n(\tau_1, \dots \tau_n)$ is the Volterra kernel of order n

In the same paper, a inverse Volterra filter is derived and used for loudspeaker linearization. The Volterra expansion and the inverse filter are limited to low order (Q= 3).

That paper has proven to be very influential and different approaches have been proposed in the same direction [3, 5]

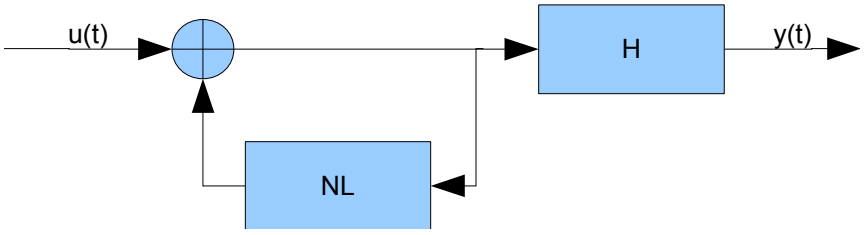

Figure 3.1: non-linear feedback followed by linear block diagram from Klippel

The white box approach is still very alive but is limited to low frequencies and low order nonlinearities (typically two or three). The linearization based on the resulting model is not exact and tend to introduces higher-order distortion [9].

It is worth noting that all linearization are based on feedforward controller because a feedback control is not practical and too costly for most of loudspeaker usages (e.g. cellphone).

3.2 Black-box (1990's)

While the white box approach continued to be researched, in the 90's, some put aside the physical model as too simplistic and explored input/output model with no physical insight. First one to be studied was NARMAX (Nonlinear AutoRegressive Moving Average with eXogenous input), in the time domain [4]. With a NARMAX model, we have a time-domain input-output mathematical relationship of the following type:

$$y_t = f(y_{t-1}, y_{t-2}, ...y_{t-n}, u_t, u_{t-1}, ...u_{t-m}, e_{t-1}, e_{t-2}, ...e_{t-d}) + e_t$$

y being the output, u the input, e the output noise and f(.) a nonlinear function (e.g. polynomial).

Other attempts were made in the frequency domain, using a general Volterra model [7]. Volterra models are interesting because of their standard and general approach. They relate immediately to the frequency domain and provide generalized frequency responses, but their complexity is such that the order is limited practically to three.

3.3 Block Model (2000's)

This last decade, the research has focused on finding a practical and simple approximation instead of a comprehensive model.

In 2000, Angelo Farina issued a paper that had a strong influence on the electroacoustic community [19]. He showed that by using a exponentially swept sine excitation it was possible to measure the free field response of a transducer in an ordinary room subject to echos and reverberation. Moreover that input signal made possible to have separate access to the linear response and all the individual harmonic nonlinear responses. Although that study stop short of system identification the underlying model is a simplified Volterra model with diagonal kernels $h_n(t, t...t) \equiv h_n(t)$

The output of the system is then:

$$y(t) = \sum_{n=1}^{Q} u^n(t) * h_n(t)$$

This is the equation of a parallel Hammerstein system (see 3.2).

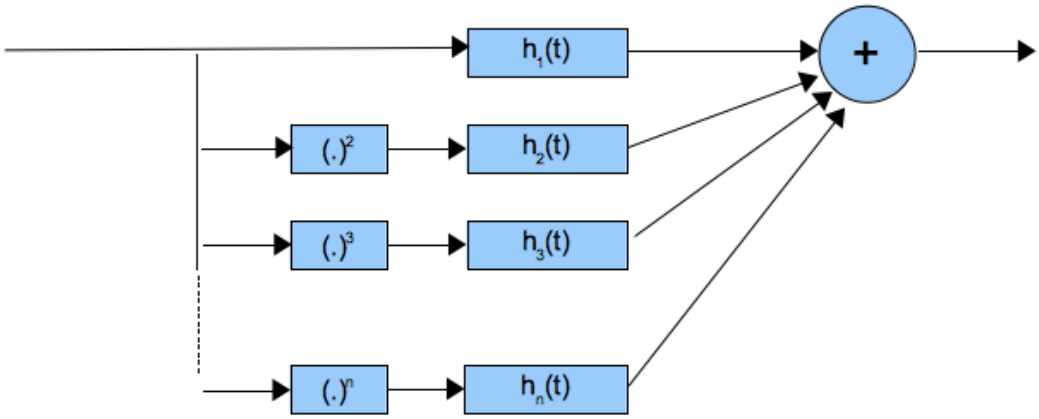

Figure 3.2: Parallel-Hammerstein System

The above work has been pursued in [12] to obtain a proper identification of each path, more exactly of each transfer function h_n.

This model is appealing by its simplicity (one FRF by order of nonlinearity) and its scalability (branches are added for increasing order), but it seems pretty remote of the physical model and somehow artificial. In particular, it does not model the nonlinear feedback mechanism of the physical model [13].

Independently, [9] proposes a modified Wiener-Volterra model (see fig.3.3), that has the property of having an exact inverse. That makes it suitable for loudspeaker linearization, by derivation of a predistortion filter.

It is interesting to note that the parallel-Hammerstein and the Wiener-Volterra models cover the full acoustic frequency range contrary to previous models.

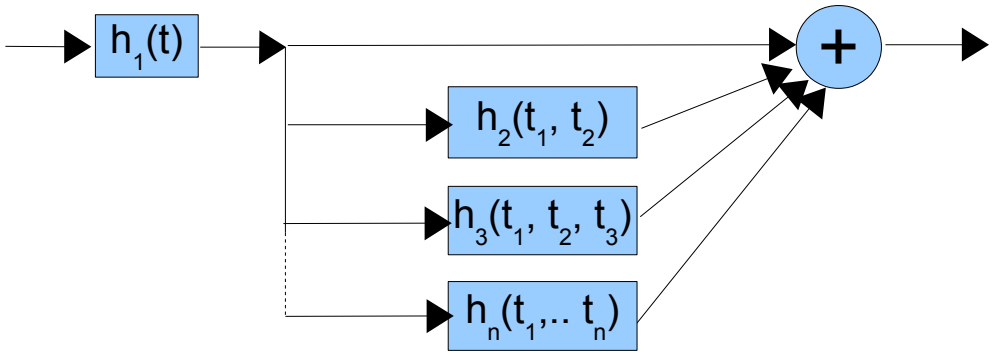

Figure 3.3: Wiener-Volterra System

Chapter 4

Polynomial Nonlinear State-Space Model

In this chapter the Polynomial Nonlinear State-Space (PNLSS) modeling is introduced [20, 21], with application to nonlinear modeling of loudspeaker through some illustrative examples. An identification procedure is proposed consisting of two parts: frequency domain subspace identification for the linear parameters of the PNLSS model and time domain nonlinear optimization for the nonlinear parameters. Simulation and experimental results are presented and discussed.

4.1 Nonlinear State-Space Modeling

4.1.1 Frequency Domain Block Model

In the last decade many papers have been published on the frequency domain approach (see [22], [23] and the references therein). The general approach is the following:

1. Find the Best Linear Approximation (BLA)

2. Identify the added nonlinearity,

which is practical and well-suited for weakly nonlinear systems.

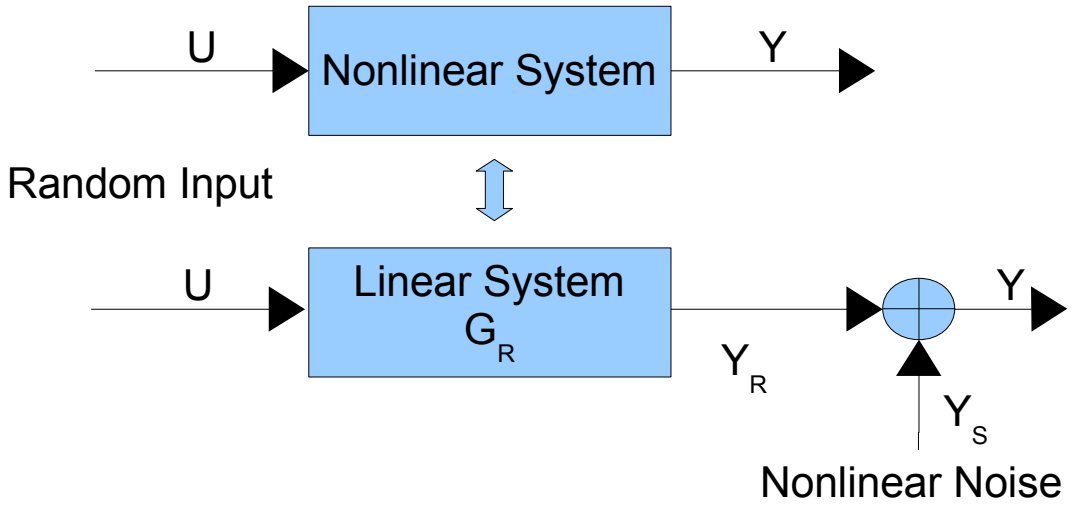

Figure 4.1: Nonlinear system with random input and its equivalent linear system + nonlinear noise source.

This approach is justified by the following fact that a Volterra system subjected to Gaussian random input is equivalent to a linear system with an added noise source at the output [18, 23]. That concept is illustrated by fig. 4.1. The linear part Y_R contains all the contributions coherent with the excitation and the nonlinear part Y_S gathers all the non coherent contributions. For example, for each frequency $\omega, Y_S(\omega)$ is the sum of contributions like:

$$H_3(\omega_1, \omega_2, \omega - \omega_1 - \omega_2)U(\omega_1)U(\omega_2)U(\omega - \omega_1 - \omega_2) \tag{4.1}$$

where H_3 is the Volterra generalized frequency response of order 3 and $\omega_1 + \omega_2 \neq 0$. This contribution is a combination of inputs at other frequencies and has a net random phase unrelated to the phase of the input $U(\omega)$. $Y_S(\omega)$ being the sum of

these random contributions cannot be distinguished from a noise. Even and odd order contributions of different orders add up to $Y_S(\omega)$ as long as the sum of the different frequencies involved is equal to ω. We can have a coherent contribution in the special case of odd orders with frequencies that cancel each other in pairs. In the example considered, if $\omega_1 + \omega_2 = 0$ we have: $H_3(\omega_1, -\omega_1, \omega)U(\omega_1)U(-\omega_1)U(\omega) = H_3(\omega_1, -\omega_1, \omega)|U(\omega_1)|^2U(\omega)$ which has a fixed phase in regard with the input $U(\omega)$. This coherent signal cannot be distinguished from a linear contribution and introduces a bias on the linear response $H_1(\omega)$. Finally it is worth noting that even if the nonlinear contributions appear like an added noise, they are deterministic for a given realization of the input signal.

Using these considerations a general purpose and flexible block model is proposed as shown in Fig. 4.2. It is a parallel structure with each branch representing a typical situation. The 1^{st} branch is simply the linear case (c is a pure real gain). In the following branches NL_i are static polynomials systems. The 2^{nd} branch is Hammerstein system. The 3^{rd} branch is a Wiener system. The 4^{th} branch is a cascade approximation of a nonlinear feedback. Note that linear block G is the same in all branches. That model is identified in successive steps. First the best linear approximation of the overall system is identified and inserted as G in all branches. Then the active branches are selected based on their power contributions. Finally the polynomial NL_i of the selected branches are identified. This approach seems well suited to loudspeaker identification. In particular, the nonlinear feedback that is part of electrodynamic loudspeaker mechanism can be identified.

4.1.2 Theory

The analysis of previous section encourages one to look into approximation of nonlinear state-space representations to obtain a proper model for loudspeakers.

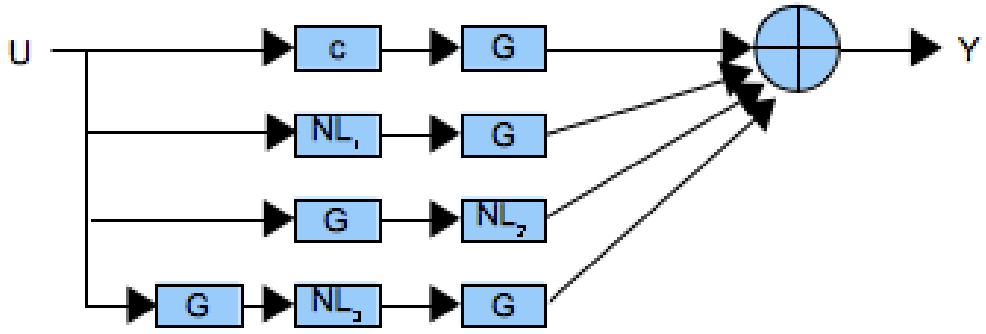

Figure 4.2: Generic nonlinear model for frequency domain identification.

The most general representation of nonlinear system in state space notation can be expressed as:

$$\begin{aligned} \dot{x}(t) &= f(x,u,t) \\ y(t) &= h(x,u,t) \end{aligned} \tag{4.2}$$

for continuous-time system, where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^l$. The analysis and design of nonlinear system (4.2) is not a trivial task.

PNLSS

Recently the following class of Lipschitz nonlinear systems has attracted a considerable attention [24]:

$$\begin{aligned} \dot{x}(t) &= Ax + Bu + Ep(x,u) \\ y(t) &= Cx + Du + Fq(x,u) \end{aligned} \tag{4.3}$$

where $p(t) \equiv p(x,u)$ and $q(t) \equiv q(x,u)$ satisfy the Lipschitz condition. Note that in this case the nonlinearity is in additive format. It can be shown that applying functional expansion of the function $\mathbf{f}$ and $\mathbf{h}$ in (4.2) with various kinds of basis functions, one can arrive at (4.3). In this paper, a set of polynomial basis functions

is chosen due to computational simplicity and its advantage in our application. The PNLSS model is defined by (4.3) consisting of the linear terms in x(t) and u(t) with constant coefficient matrices A, B, C, D, E, F and the vectors $p(t) \in \mathbb{R}^{n_p}$ and $q(t) \in \mathbb{R}^{n_q}$ containing nonlinear monomials in x(t) and u(t) of degree two up to a chosen degree r, where the coefficient matrices E and F contain the coefficients associated with those monomials. Note that the monomials of degree one are included in the linear part of the PNLSS model structure. When a full polynomial expansion is carried out, all monomials up to degree r must be taken into account. First, a vector z is defined as the concentration of the state vector and the input vector as

$$z(t) = [x_1(t) \dots x_n(t) u_1(t) \dots u_m(t)]^T \tag{4.4}$$

As a consequence, the dimension of the vector z(t) is given by $n_z = n+m$. Then, using the conventional index notation for monomials we define:

$$p(t) = q(t) = z(t)_{\{r\}} \tag{4.5}$$

Note that the vector $z(t)_{\{r\}}$ as defined in (4.5) should contain all monomials with a degree between two and r. For instance, the vector $z_{\{3\}}$ with $n_z = 2$ denotes

$$z_{\{3\}} = [z_{(2)} z_{(3)}]^T = [z_1^2, z_1 z_2, z_2^2, z_1^3, z_1^2 z_2, z_1 z_2^2, z_2^3]^T \tag{4.6}$$

where we define $z_{(r)}$ as the vector of all the distinct monomials of degree r composed from the elements of vector z. The number of elements in vector $z_{(r)}$ is given by the following binomial coefficient

$$N_r = \binom{n_z + r - 1}{r} \tag{4.7}$$

Thus, the vector $z_{\{r\}}$ has the length

$$L_r = \binom{n_z + r}{r} - 1 - n_z \tag{4.8}$$

and corresponds to considering all the distinct nonlinear combinations of degree r, which is the default choice for the PNLSS model structure. The total number of parameters required by the model in (4.3) , is given by

$$N = \left[\binom{n+m+r}{r} - 1 \right] (n+l) \tag{4.9}$$

4.1.3 Illustrative Examples

Duffing Oscillator

Duffing Equation: damped mass-spring system with hardening spring.

$$m\ddot{x} + r\dot{x} + k(1 + q^2x^2)x = u \tag{4.10}$$

State-space Equation in continuous time:

$$z_1 \triangleq x, z_2 \triangleq \dot{x}, z \triangleq (z_1, z_2)^T$$

$$\dot{z_1} = z_2$$
$$\dot{z_2} = -\frac{k}{m}z_1 - \frac{r}{m}z_2 + \frac{1}{m}u - \frac{k}{m}q^2z_2^3$$
$$y = z_1$$

State-space Equation in discrete time:
$\dot{x(t)} \approx (x(t + Ts) - x(t))/T_s$, Euler approximation with sampling period T_s

$$\zeta_1 \triangleq x(nT_s), \zeta_2 \triangleq \dot{x}(nT_s), \zeta(n) \triangleq (\zeta_1(n), \zeta_2(n))^T$$

$$\zeta_1(n+1) = \zeta_1(n) + T_s\zeta_2(n)$$
$$\zeta_2(n+1) = -\frac{kT_s}{m}\zeta_1(n) + (1 - \frac{rT_s}{m})\zeta_2(n) + \frac{T_s}{m}u(n) - \frac{kT_s}{m}q^2\zeta_2^3(n)$$
$$y(n) = \zeta_1(n)$$

Numerical example:

- m= 14.35 g

- r= 0.786 N.s/m
- k= 1852 N/m
- $q^2 = 3.33 \times 10^5$
- $T_s = 44100$ Hz

Resonance frequency is given by: $f_0 = \frac{1}{2\pi}\sqrt{\frac{k}{m} - \frac{r^2}{2m^2}} = 56.84$ Hz

The sinusoidal excitation frequency is set at 56 Hz for an FFT size of 48000 (1 Hz resolution). Two experiments are made : one with an amplitude of 9 N, the second with 100 N. At 9 N strong harmonic distortion appears (fig. 4.4, fig. 4.5). At 100 N the response shows chaotic behavior (fig. 4.6, fig. 4.7).

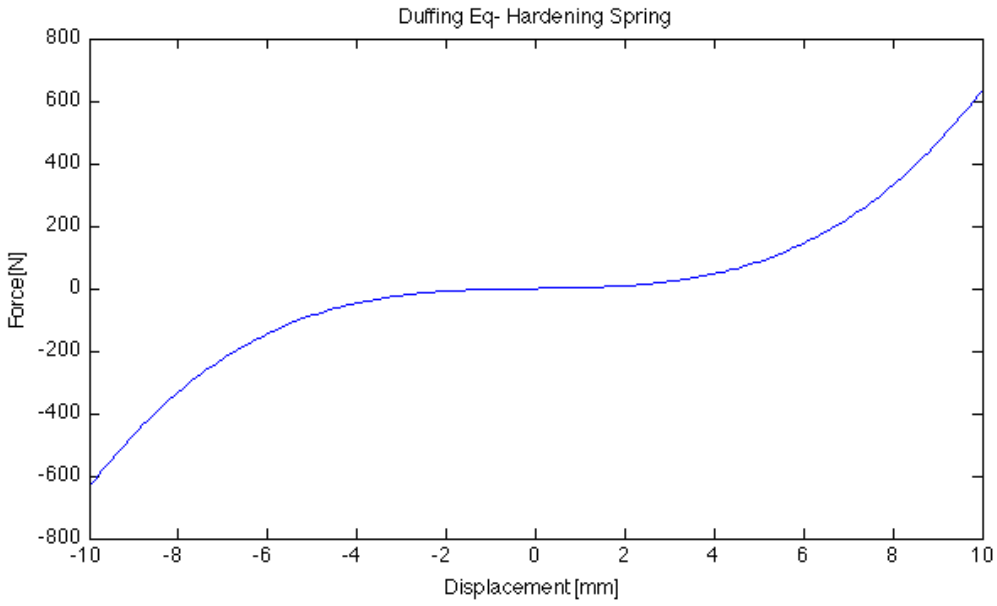

Figure 4.3: Stiffness $k(1 + q^2x^2)$ vs. displacement x

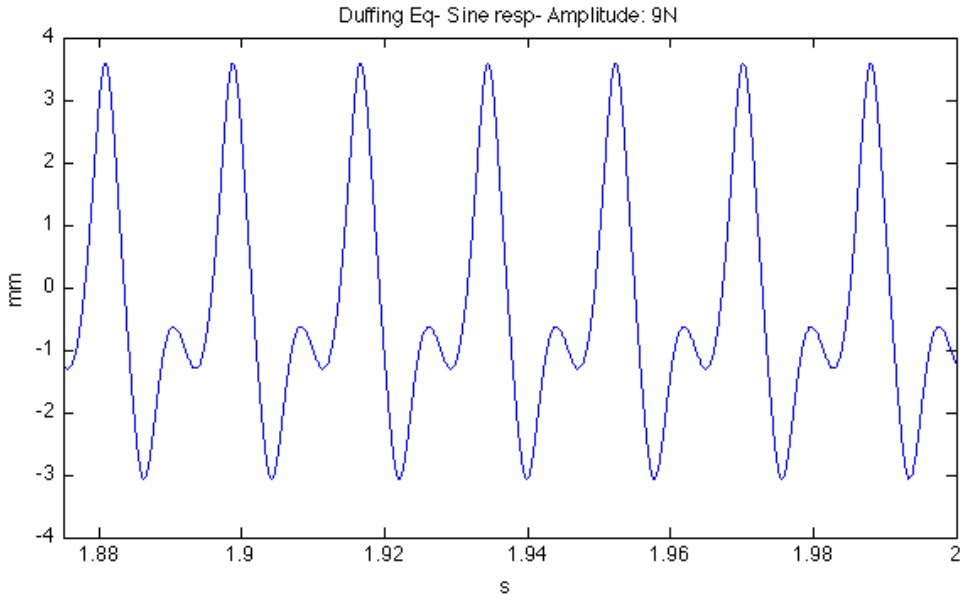

Figure 4.4: Time response to sinusoidal excitation 56 Hz, 9N

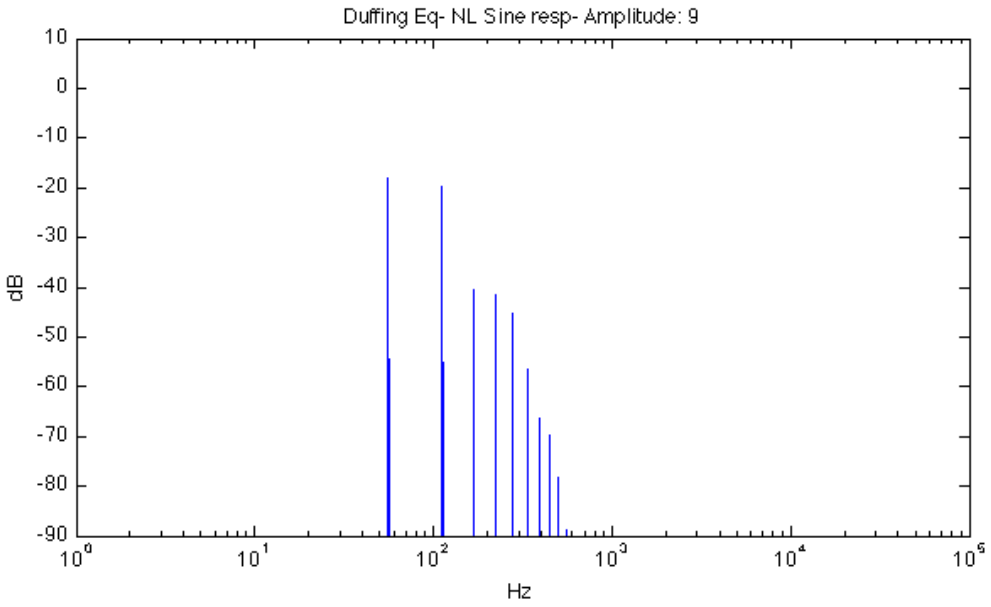

Figure 4.5: Frequency response to sinusoidal excitation 56 Hz, 9N

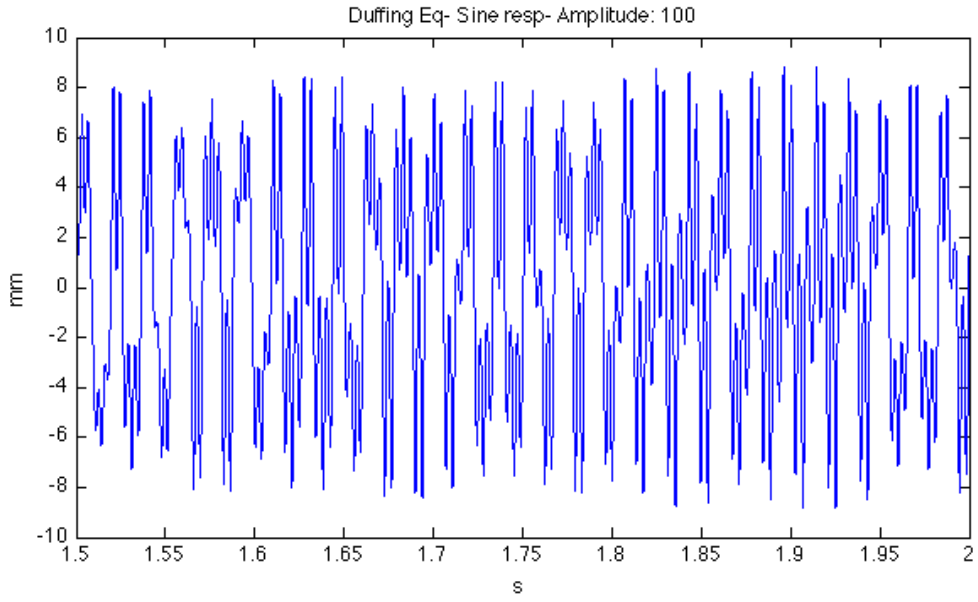

Figure 4.6: Time response to sinusoidal excitation 56 Hz, 100N

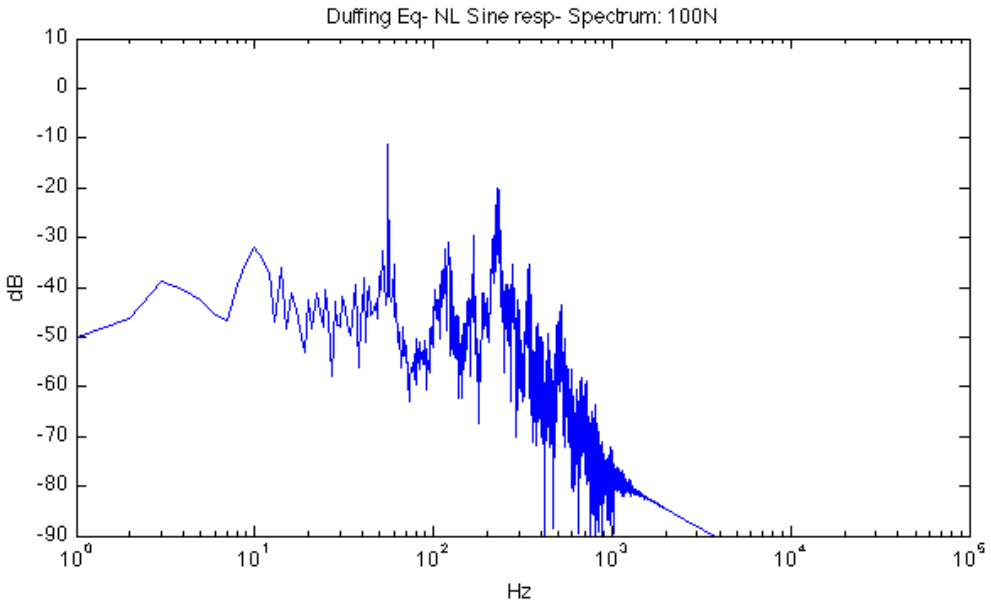

Figure 4.7: Frequency response to sinusoidal excitation 56 Hz, 9N

Simulation with Simulink (see fig. 4.8), fixed step= 1/44100, Euler (ode1) gives same results.

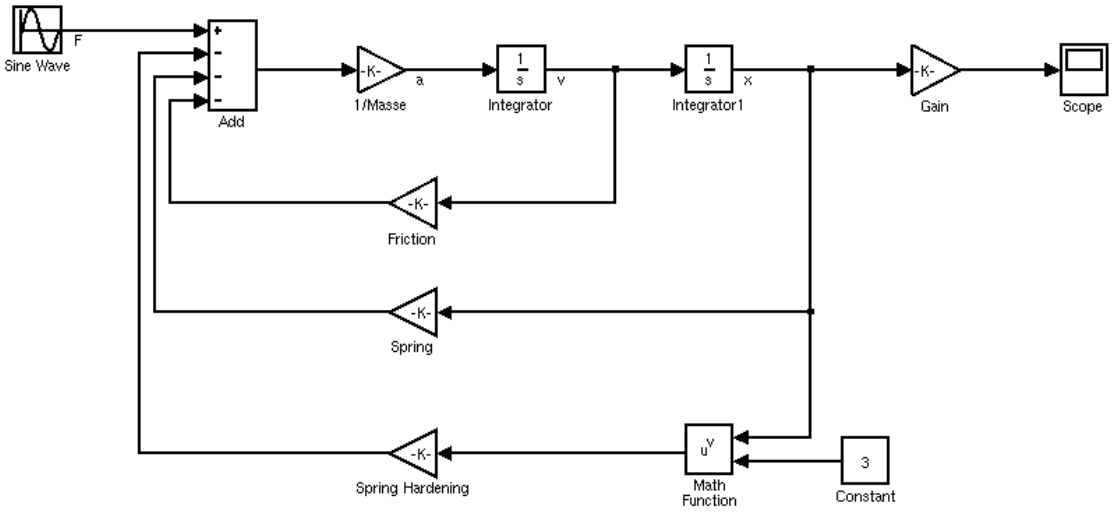

Figure 4.8: Simulink diagram of Duffing oscillator

Van Der Pol Oscillator

Van Der Pol Equation:

$$m\ddot{v} - \epsilon(1 - v^2)\dot{v} + v = u \tag{4.11}$$

State-space Equation in continuous time:

$$z \triangleq (v, \dot{v})^T = (z_1, z_2)$$

$$\dot{z_1} = z_2$$

$$\dot{z_2} = -z_1 + \epsilon z_2 + u - \epsilon z_1^2 z_2$$

$$y = z_1$$

Euler approximation: $\dot{v}(t) \approx (v(t + Ts) - v(t)/T_s$, with sampling period T_s

$$\zeta_1 \triangleq x(nT_s), \zeta_2 \triangleq \dot{x}(nT_s), \zeta(n) \triangleq (\zeta_1(n), \zeta_2(n))^T$$

$$\zeta_1(n+1) = \zeta_1(n) + T_s\zeta_2(n)$$

$$\zeta_2(n+1) = -T_s\zeta_1(n) + (1 + \epsilon T_s)\zeta_2(n) + T_s u(n) - \epsilon T_s \zeta_1^2(n)\zeta_2(n)$$

$$y(n) = \zeta_1(n)$$

Numerical example:

Sinusoidal excitation: $u = Usin(\omega n T_s)$ with:

- $\epsilon = 8.53$
- U= 1.2
- $\omega = 2\pi/10$
- $T_s = 1/100$

With that excitation, the oscillator response is chaotic (see fig. 4.9, fig. 4.10).

Simulation with Simulink, fixed step= 1/100, Euler (ode1) gives same results.

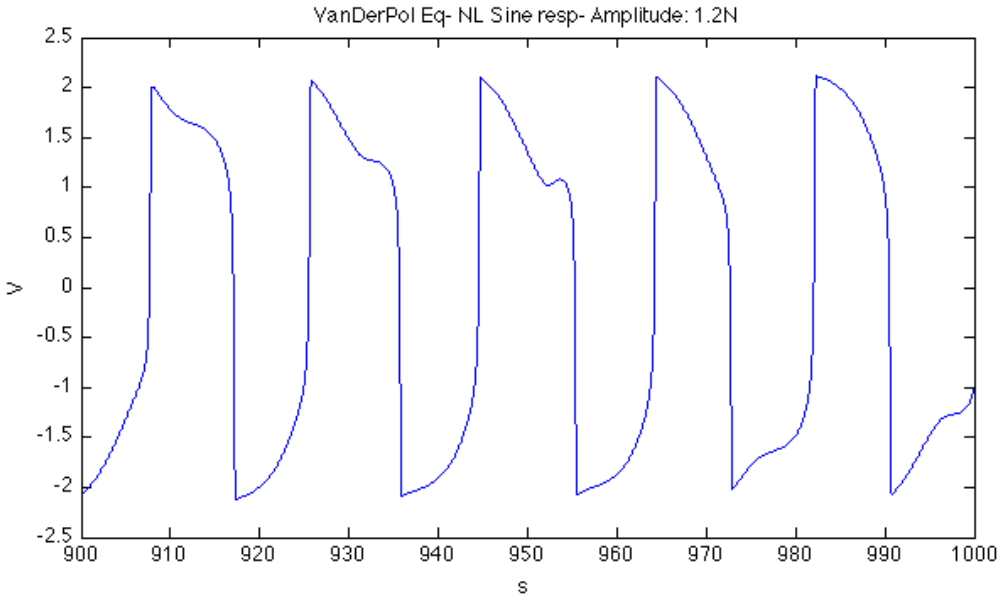

Figure 4.9: Time response to sinusoidal excitation 0.1 Hz, 1.2N

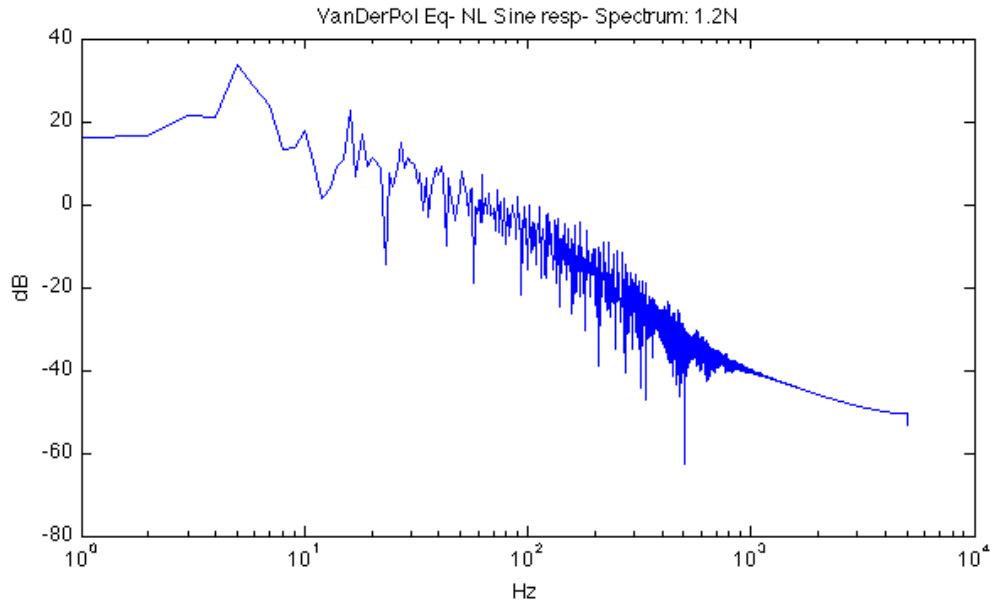

Figure 4.10: Frequency response to sinusoidal excitation 0.1 Hz, 1.2N

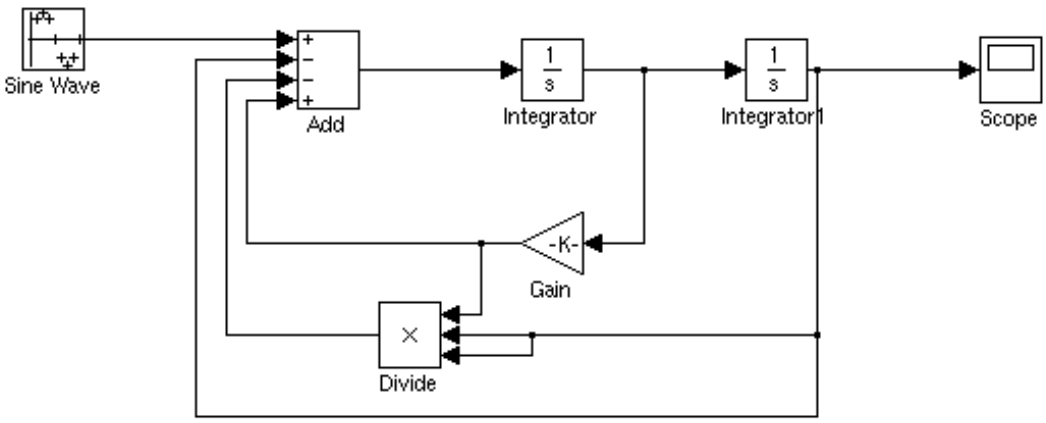

Figure 4.11: Simulink diagram of Van Der Pol oscillator

4.1.4 Loudspeaker PNLSS Modeling

In this section we focus on the loudspeaker electromechanical part and we show that a Taylor expansion of its parameters leads to a PNLSS model. The SS model (4.23) with current output is described by:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{k}{m} & -\frac{r}{m} & \frac{Bl}{m} \\ 0 & -\frac{Bl}{L} & -\frac{R}{L} \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 & \frac{1}{L} \end{pmatrix}^T$$

$$C = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$$

With state vector:

$$z = \begin{pmatrix} x & \dot{x} & i \end{pmatrix}$$

Observability

The observabiltity matrix is:

$$O = \begin{pmatrix} C \\ CA \\ CA^2 \end{pmatrix}$$

The determinant is: $|O| = (\frac{Bl}{L})^2 \frac{k}{m} > 0$ for all real loudspeakers. Therefore the system is observable from the current output, meaning that position and velocity can be recovered from the voltage and current. That can be achieved for example by a state observer like the Luenberger observer.

Polynomial Approximation of Nonlinear Parameters

The voice-coil inductance L is function of excursion x because the coil moves relative to the pole pieces.

The force factor Bl is the product of the magnetic flux density by the coil length. Bl change with coil position in regard to the magnet and is then a function of x.

The stiffness k of suspension is also function of x as it becomes harder as the magnitude of excursion increases.

Therefore we can express the SS matrices as functions of the excursion x:

$$A(x) = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{k(x)}{m} & -\frac{r}{m} & \frac{Bl(x)}{m} \\ 0 & -\frac{Bl(x)}{L(x)} & -\frac{R}{L(x)} \end{pmatrix}$$

$$B(x) = \begin{pmatrix} 0 & 0 & \frac{1}{L(x)} \end{pmatrix}^T$$

Using the same approach as in [2], L, Bl, and k are approximated by second order polynomials. That approach is justified by the fact that L, Bl, and k are usually

smooth functions of the excursion [8].

$$\begin{aligned}
-\frac{k(x)}{m} &\triangleq \alpha_0 + \alpha_1 x + \alpha_2 x^2 \\
-\frac{r}{m} &\triangleq \beta_0 \\
\frac{Bl(x)}{m} &\triangleq \gamma_0 + \gamma_1 x + \gamma_2 x^2 \\
-\frac{Bl(x)}{L}(x) &\triangleq \delta_0 + \delta_1 x + \delta_2 x^2 \\
-\frac{R}{L(x)} &\triangleq \epsilon_0 + \epsilon_1 x + \epsilon_2 x^2 \\
-\frac{1}{L(x)} &\triangleq \mu_0 + \mu_1 x + \mu_2 x^2
\end{aligned}$$

Assumption: the electrical inductance L(x) doesn't vanish in the excursion range of the loudspeaker. In other words, the voice coil doesn't escape from the magnet.

Then the state matrix can be expressed as:

$$\begin{aligned}
A(x) &= \begin{pmatrix} 0 & 1 & 0 \\ \alpha_0 & \beta_0 & \gamma_0 \\ 0 & \delta_0 & \epsilon_0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ \alpha_1 & 0 & \gamma_1 \\ 0 & \delta_1 & \epsilon_1 \end{pmatrix} x + \begin{pmatrix} 0 & 0 & 0 \\ \alpha_2 & 0 & \gamma_2 \\ 0 & \delta_2 & \epsilon_2 \end{pmatrix} x^2 \\
&\triangleq A_0 + A_1 x + A_2 x^2
\end{aligned}$$

And the input matrix:

$$\begin{aligned}
B(x) &= \begin{pmatrix} 0 & 0 & \mu_0 \end{pmatrix}^T + \begin{pmatrix} 0 & 0 & \mu_1 \end{pmatrix}^T x + \begin{pmatrix} 0 & 0 & \mu_2 \end{pmatrix}^T x^2 \\
&\triangleq B_0 + B_1 x + B_2 x^2
\end{aligned}$$

Reminding that the excursion x is the first entry of the state vector $z = \begin{pmatrix} x & \dot{x} & i \end{pmatrix}$, and after re-arranging the terms the state equation can be rewritten as:

$$\dot{z} = (A_0 z + B_0 u) + (A_1 z_1 z + A_2 z_1^2 z + B_1 z_1 u + B_2 z_1^2 u)$$

$\dot{z}$ is the sum of a linear part and nonlinear part. The nonlinear part can be expressed in terms of monomials of order 2 and 3 of z_1, z_2, z_3, u.

$$A_1 z_1 z + A_2 z_1^2 z + B_1 z_1 u + B_2 z_1^2 u = EP(z^T, u) \tag{4.12}$$

where E is a constant matrix and $P(z^T, u)$ is the vector of all monomials of degree 2 and 3 of z_1, z_2, z_3, u.

$P(z^T, u)$ has $\binom{4+2-1}{2} + \binom{4+3-1}{3} = 10 + 20$ terms and $E \in \mathbb{R}^{3\times 30}$.

$$\begin{aligned} P(z^T, u) = \big(&z_1^2, z_1z_2, z_1z_3, z_1u, z_2^2, z_2z_3, z_2u, z_3^2, z_3u, u^2, z_1^3, \\ &z_1^2z_2, z_1^2z_3, z_1^2u, z_1z_2^2, z_1z_2z_3, z_1z_2u, z_1z_3^2, z_1z_3u, z_1u^2, z_2^3, \\ &z_2^2z_3, z_2^2u, z_2z_3^2, z_2z_3u, z_2u^2, z_3^3, z_3^2u, z_3u^2, u^3\big)^T \end{aligned}$$

Expanding (4.12) yields:

$$EP(z^T, u) = \begin{pmatrix} 0 \\ \alpha_1 z_1^2 + \delta_1 z_1 z_3 + \alpha_2 z_1^3 + \delta_2 z_1^2 z_3 \\ \delta_1 z_1 z_2 + \epsilon_1 z_1 z_3 + \mu_1 z_1 u + \delta_2 z_1^2 z_2 + \epsilon_2 z_1^2 z_3 + \mu_2 z_1^2 u \end{pmatrix}$$

By identification, the coefficients of E follow:

i	2	2	2	2	3	3	3	3	3	3
j	1	3	11	13	2	3	4	12	13	14
E(i,j)	α_1	δ_1	α_2	δ_2	δ_1	ϵ_1	μ_1	δ_2	ϵ_2	μ_2

E(i,j) = 0, otherwise.

E is sparse. Only 10 coefficients are non zero amongst 90.

Altogether, in this section, we have shown that by taking in account the nonlinear variability of the loudspeaker parameters and using fair approximations, one can obtain quite straightforwardly a PNLSS modeling.

4.2 Identification Procedure

In this section, we present an overview of the PNLSS identification procedure in three steps. We follow up with a detailed description of each of them.

4.2.1 Procedure Overview

The identification procedure for PNLSS model consists of three major steps. First, the BLA of the system under test is determined non-parametrically in the mean square sense. Then, a parametric linear model is estimated from the BLA using frequency domain subspace identification method [25]. The last step consist of estimating the full nonlinear model by using a nonlinear search algorithm that minimizes the model output error in regard to the measured output. Procedure:

1. Frequency Response Function (FRF) estimation with periodic random phase multitone and frame-averaging.

2. Frequency Domain Subspace Identification:

   (a) Get FRF values and their sample variance, for some appropriate subset of frequencies

   (b) Estimate an extended observability matrix of $\hat{O}_r$ with sufficient rank r

      i. Construction of matrix Z as a frequency weighted input-output measurements compound matrix

      ii. Elimination of input term by QR factorization of Z

      iii. Reduction of noise influence by SVD of the factored term weighted by inverse of the variance and estimation of $\hat{O}_r$

   (c) Estimation the system matrices $\hat{A}, \hat{C}$ by least-squares solution of $\hat{O}_r$ recursive equations

(d) Remaining system matrices $\hat{B}, \hat{D}$ estimation by weighted linear least-square regression w.r.t measured FRF

3. Nonlinear Least Square Time Domain Identification:

(a) Time Domain Error Minimization by Iterative Procedure

(b) $\theta = [Vec(E), Vec(F)]$ vector of polynomial coefficients

(c) Parameter update uses Levenberg-Marquard algorithm

(d) Final parameter $\hat{\theta}$ minimize the residual error $\|r(t)\|$.

(e) This final step estimates the polynomial coefficients $\hat{E}, \hat{F}$

It should be pointed out that a parallel treatment of previous section for discrete-time nonlinear system can be established. This is convenient for system identification process. In this case, without loss of generality, we see similar state space notation as follows:

$$\begin{aligned} x(t+1) &= Ax + Bu + Ep(t) \\ y(t) &= Cx + Du + Fq(t) \end{aligned} \tag{4.13}$$

4.2.2 Best Linear Approximation

In this section we show discuss the method used to estimate the Best Linear Approximation of the system under test.

The method described below is inspired from [26]. As an input signal, we use a multisine with flat spectrum and random phases:

$$u(t) = U_0 \sqrt{\frac{2}{N}} \sum_{k=1}^{N} \cos\left(\omega_k t + \phi_k\right)$$

The resulting signal has an rms value of U_0.

The use of harmonic frequencies ensures a periodic signal: $\omega_k = 2\pi \frac{k}{T}$.

The phases are random, uniformly distributed on $[0, 2\pi)$.That phase pattern ensures a signal with random values and an amplitude distribution that tends asymptotically to a Gaussian law when $N \to \infty$.

A general input sequence is composed of $M \times (P + 1)$ periods of multisine. Each sequence of $P+1$ periods uses a particular realization of the random phases sequence, and M successive realizations of the phases leads to $M \times (P+1)$ periods of multisine. Periodic signal allows to average out the output noise and different realizations of the random phases pattern allow to average out the nonlinear amplitude distortion. For each realization, the first of the $P+1$ periods is disregarded as it contains the settling transient of the system.

For each period p of realization m, the DFT of input period is denoted by $U^{[m,p]}(k)$ and DFT of output period by $Y^{[m,p]}(k)$.

Averaging is done first along the P periods of each realization m:

$$\hat{U}^{[m]}(k) = U^{[m,p]}(k), \forall p, \text{ assuming no input noise}$$

$$\hat{Y}^{[m]}(k) = \frac{1}{P} \sum_{p=1}^{P} Y^{[m,p]}(k)$$

$$\hat{G}^{[m]}(k) = \frac{\hat{Y}^{[m]}(k)}{\hat{U}^{[m]}(k)}$$

Finally the BLA of the system FRF is obtained by averaging over M realizations:

$$\hat{G}(k) = \frac{1}{M} \sum_{m=1}^{M} \hat{G}^{[m]}(k)$$

The total sample variance of the BLA is obtained by:

$$\hat{\sigma}_G^2(k) = \frac{1}{M-1} \sum_{m=1}^{M} |\hat{G}^{[m]}(k) - \hat{G}(k)|^2$$

The variance due to the measurement noise can be estimated separately by averaging over the P periods of each realization. The variance due to nonlinearities can then be deduced by subtracting the noise variance from the total variance.

The nonlinear contributions can be minimized by using only odd order harmonics in the multitone stimulus [23, 26]. Indeed, even order distortion produces only components at even frequencies that do not fall on the excitation lines. Over types of multisine (odd odd, quasi-log,...) are described in the aforementioned references with their specific advantages and case of use.

4.2.3 Frequency Domain Subspace Identification

In this section, we expose a method to find a linear subspace model that fits the measured FRF $\hat{G}(k)$ in the LS sense.

Model Equation

We have:

$$\begin{aligned} x(n+1) &= Ax(n) + Bu(n) \\ y(n) &= Cx(n) + Du(n) \end{aligned} \tag{4.14}$$

In case of periodic input, when the system is the steady state, the DFT of (4.14) yields:

$$\begin{aligned} z_k X(k) &= AX(k) + BU(k) \\ Y(k) &= CX(k) + DU(k) \end{aligned} \tag{4.15}$$

Where:

$$\begin{aligned} k &= 1 \dots F, \text{frequency index} \\ F(k) &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n) e^{-j\omega_k n}, \text{DFT of f= u, x, y} \\ z_k &= e^{-j\omega_k} \end{aligned}$$

We assume an integer number of period on each DFT block. Considering a flat input spectrum $U(k) = 1$, we derive the FRF form of the equation (4.15):

$$\begin{aligned} z_k X(k) &= AX(k) + B \\ G(k) &= CX(k) + D \end{aligned} \tag{4.16}$$

Recursive use of these equations gives:

$$\begin{aligned} G(k) &= CX(k) + D \\ z_k G(k) &= CAX(k) + CB + z_k D \\ z_k^2 G(k) &= CA^2 X(k) + CAB + z_k CB + z_k^2 D \\ &\vdots \\ z_k^{r-1} G(k) &= CA^{r-1} X(k) + CA^{r-2} B + z_k CA^{r-3} B + \dots z_k^{r-2} CB + z_k^{r-1} D \end{aligned}$$

Written in a matrix form:

$$W_r(k) G(k) = O_r X(k) + S_r W_r(k) \tag{4.17}$$

with:

$$\begin{aligned} W_r(k) &\triangleq (1, z_k, \dots z_k^{r-1})^T \in \mathbb{C}^r \\ O_r &\triangleq (C, CA, \dots CA^{r-1})^T \in \mathbb{R}^{r \times n_a} \\ S_r &\triangleq \begin{pmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ CA^{r-2}B & CA^{r-3}B & \cdots & CB & D \end{pmatrix} \in \mathbb{R}^{r \times r} \end{aligned}$$

Note that O_r is the extended observability matrix ($r > n_a$) of the studied system and S_r is the impulse response matrix.

Collecting (4.17) for all frequencies $k = 1 \dots F$ gives:

$$\tilde{G} = O_r \tilde{X} + S_r \tilde{W} \tag{4.18}$$

with:

$$\tilde{G} \triangleq (W_r(1)G(1), \dots W_r(F)G(F)) \in \mathbb{C}^{r \times F}$$

$$\tilde{X} \triangleq (X(1), \dots X(F)) \in \mathbb{R}^{n_a \times F}$$

$$\tilde{W}_r \triangleq (W_r(1), \dots W_r(F)) \in \mathbb{C}^{r \times F}$$

Finally the real and imaginary parts of each matrix are written side by side using the following convention: $(.)^{re} = (Re(.), Im(.))$. That last step result in the real coefficients equation model:

$$\tilde{G}^{re} = O_r \tilde{X}^{re} + S_r \tilde{W}^{re} \tag{4.19}$$

Noise Model

Let the model (4.19) be observed with noise at the output leading to errors on the estimated FRF:

$$G(k) = G_0(k) + N_G(k)$$

where $G_0(k)$ is the real FRF and $N_G(k)$ represents a zero-mean circular complex random error.

Defining:

$$N_G \triangleq (W_r(1)N_G(1), \dots W_r(F)N_G(F)) \in \mathbb{C}^{r \times F}$$

We write the equation of the model with added noise:

$$\tilde{G}^{re} = O_r \tilde{X}^{re} + S_r \tilde{W}^{re} + \tilde{N}_G^{re} \tag{4.20}$$

Algorithm

Subspace algorithm with non-uniform frequency data and sample noise covariance from algorithm 2 in [25, 27].

1. Get FRF BLA $\hat{G}(j\omega_k)$ and its sample variance $\hat{\sigma}^2_G(j\omega_k)$, for the set of radial frequencies $\{\omega_k; k = 1, \dots F\}$. In the following we denote $\hat{G}(j\omega_k) \triangleq \hat{G}(k)$.

2. Extended observability matrix given $r > n_a$, with r chosen in the range $[1.5n_a, 6n_a]$

   (a) Construct the following matrices:

$$
\begin{aligned}
z_k &= e^{j\omega_k}, k = 1 \dots F \in \mathbb{C}^F \\
W_r(k) &= (1, z_k \dots z_k^{r-1})^T \in \mathbb{C}^r \\
\tilde{W} &= (W_r(1), \dots W_r(F)) \in \mathbb{C}^{r\times F} \\
\tilde{G} &= (W_r(1)\hat{G}(1) \dots W_r(F)\hat{G}(F)) \in \mathbb{C}^{r\times F} \\
\tilde{W}^{re} &= (Re(\tilde{W}), Im(\tilde{W})) \in \mathbb{R}^{r\times 2F} \\
\tilde{G}^{re} &= (Re(\tilde{G}), Im(\tilde{G})) \in \mathbb{R}^{r\times 2F} \\
Z &= (\tilde{W}^{re}, \tilde{G}^{re})^T \in \mathbb{R}^{2r\times 2F} \\
C_{in} &= (W_r(1)\hat{\sigma}_G(1) \dots W_r(F)\hat{\sigma}_G(F)) \in \mathbb{C}^{r\times F} \\
C_G &= Re(C_{in}C_{in}^H) \in \mathbb{R}^{r\times r}
\end{aligned}
$$

   (b) Elimination of input term by QR factorization

$$
\begin{aligned}
Z^T = QR &\iff Z = R^T Q^T \\
Z &= \begin{pmatrix} R_{11}^T & 0 \\ R_{12}^T & R_{22}^T \end{pmatrix} \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} = \begin{pmatrix} \tilde{W}^{re} \\ \tilde{G}^{re} \end{pmatrix} \\
R_{ij} &\in \mathbb{R}^{r\times r}
\end{aligned}
$$

R_{22} is the projection of $\hat{G}^{re}$ on the subspace orthogonal to the input $\tilde{W}^{re}$.

$$R_{11}Q_1^T = \tilde{W}_r e$$

$$R_{12}Q_1^T + R_{22}Q_2^T = \tilde{G}^{re}$$

From (4.20), it follows that:

$\tilde{G}^{re} = S_r\tilde{W}^{re} + O_r\tilde{X}^{re} + \tilde{N}_G^{re} \implies R_{22}Q_2^T = O_r\tilde{X}^{re} + \tilde{N}_G^{re}$

(c) Noise reduction by SVD and estimation of the extended Observability Matrix.

$$C_G^{-1/2}R_{22}^T = \Delta\Sigma\Gamma^T$$

$$\hat{O}_r = C_G^{1/2}\Delta[:, 1 : n_a]$$

where:

Δ is the matrix of left singular vectors

Σ is the matrix of singular values

Γ is the matrix of right singular vectors

$\hat{O}_r$ is the matrix of first n_a left singular vectors.

3. Estimation of A and C

$\hat{A} = \hat{O}_r^+[1 : r-1, :]\hat{O}_r[2 : r, :]$, LS estimate of A

$\hat{C} = \hat{O}_r[1, :]$

4. Estimation of B and D by weighted LS

$$\hat{B}, \hat{D} = \underset{B,D}{Argmin} \sum_{k=1}^{F} \frac{1}{\hat{\sigma}^2_{G(k)}} |G(k) - \hat{C}(z_k I_{n_a} - \hat{A})^{-1}B - D|^2$$

The cost function to minimize can be written in the form:

$$J = (\hat{G} - H\beta)^H W (\hat{G} - H\beta)$$

with:

$$H = \begin{pmatrix} \hat{C}(z_1 I_{n_a} - \hat{A})^{-1} & 1 \\ \vdots & \vdots \\ \hat{C}(z_F I_{n_a} - \hat{A})^{-1} & 1 \end{pmatrix} \in \mathbb{C}^{F \times (n_a+1)}$$

$$\beta = (B^T, D)^T \in \mathbb{R}^{(n_a+1)}$$

$$W = diag(\hat{\sigma}_G^{-2}(1), \ldots, \hat{\sigma}_G^{-2}(F)) \in \mathbb{R}^{F \times F}$$

The minimization is achieved for ([27]):

$$\frac{\partial J}{\partial \beta}|_{\beta=\hat{\beta}} = 0 \implies \hat{\beta} = Re(H^H W H)^{-1} Re(H^H W G)$$

4.3 Nonlinear Least-Squares Identification

The final step following the Frequency Domain Subspace Identification is the estimation of the nonlinear parameter matrices E, F by minimization of model output error in the time domain. The model output error is the difference between the system's measured output and the model output for the same given input (see fig. 4.12). As a reminder, the model we want to identify is the following PNLSS model:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + Ep(t) \\ y(t) &= Cx(t) + Du(t) + Fp(t) \end{aligned} \tag{4.21}$$

A, B, C, D are supposed to be known from the previous step. The vector p contains all monomials of degree 2 to d from states and input. For simplification, without lack of generality, the same monomial vector is used for both equations.

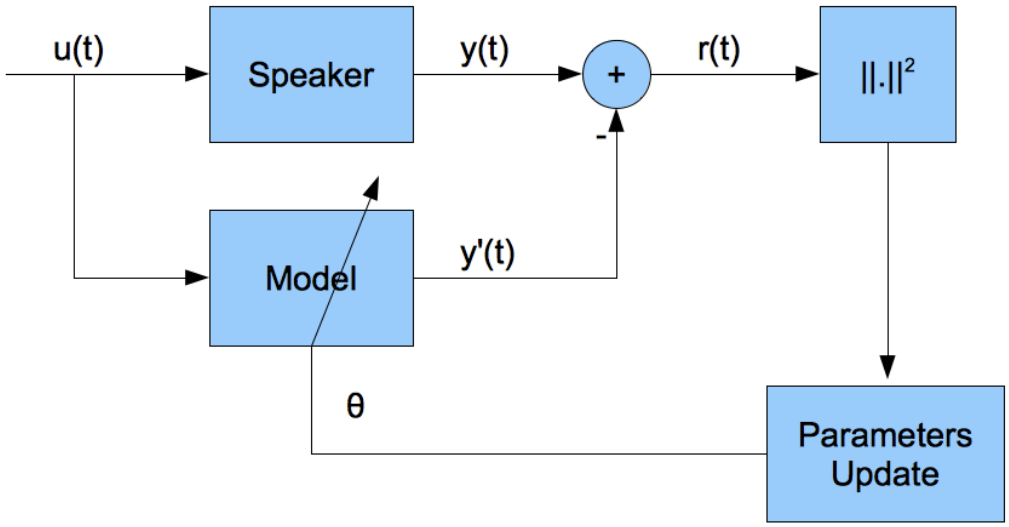

Figure 4.12: Nonlinear Least-Square Optimization Block Diagram

4.3.1 Levenberg-Marquard Algorithm

Definitions

$$y \in \mathbb{R}^m, \text{ vector of measured output samples}$$

$$\tilde{y} \in \mathbb{R}^m, \text{ vector of model output samples}$$

$$\theta \triangleq vec(E^T, F^T) \in \mathbb{R}^{n_\theta}, \text{ vector of nonlinear parameters to estimate}$$

where:

$$E \in \mathbb{R}^{n_a \times n_p}, F \in \mathbb{R}^{1 \times n_p}$$

$$n_p = \binom{n_a + 1 + d}{d} - n_a - 2$$

$$n_\theta = (n_a + 1)n_p$$

$$r(\theta) \triangleq \tilde{y}(\theta) - y \in \mathbb{R}^m, \text{ vector of residual errors}(m > n_\theta).$$

$$f(\theta) \triangleq \frac{1}{2}\|r(\theta)\|^2, \in \mathbb{R}, \text{ cost function to minimize}$$

Jacobian of residual vector r:

$$J(\theta) \triangleq \frac{\partial r}{\partial \theta} \in \mathbb{R}^{m \times n_\theta}$$

Gradient of cost-function f:

$$\nabla f(\theta) \triangleq \frac{\partial f}{\partial \theta} = \left(\frac{\partial r}{\partial \theta}^T\right) r = J^T r \in \mathbb{R}^{n_\theta \times m}$$

Hessian of cost-function f:

$$H(\theta) \triangleq \frac{\partial^2 f}{\partial \theta^2} \approx J^T J \in \mathbb{R}^{n_\theta \times n_\theta}$$

Calculus of Jacobian

Considering that: $r = \tilde{y} - y$ and $\frac{\partial y}{\partial \theta} = 0$ then:

$$J(\theta) = \frac{\partial r}{\partial \theta} = \frac{\partial \tilde{y}}{\partial \theta}$$

Each row of J is the gradient of the corresponding model output sample in regard to the parameters E, F.

As shown in [28], using (4.21) and considering $\frac{\partial u(t)}{\partial E_{i,j}} = 0$, $J(\theta)$ can be derived as :

$$\begin{aligned} \frac{\partial x(t+1)}{\partial E_{i,j}} &= A\frac{\partial x(t)}{\partial E_{i,j}} + E\frac{\partial p}{\partial x}\frac{\partial x(t)}{\partial E_{i,j}} + \frac{\partial E}{\partial E_{i,j}}p(t) \\ \frac{\partial \tilde{y}(t)}{\partial E_{i,j}} &= C\frac{\partial x(t)}{\partial E_{i,j}} + F\frac{\partial p}{\partial x}\frac{\partial x(t)}{\partial E_{i,j}} \\ \frac{\partial \tilde{y}(t)}{\partial F_j} &= \frac{\partial F}{\partial F_j}p(t) \end{aligned} \tag{4.22}$$

This set of equations allows to calculate recursively each element of the gradient $J(\theta)$, one row at a time. As seen in section 4.2.3, the Jacobian is the output of a dynamic system and must be calculated recursively for all output samples of the PNLSS model.

Expansion of $\frac{\partial p}{\partial x}$:

Let's define:

$$\xi(t) \triangleq (x^T(t)u(t))^T \in \mathbb{R}^{n_a+1}$$

concatenation of state vector and input scalar.

Then:

$$p(t) = \xi_{\{d\}}(t) \in \mathbb{R}^{n_p}, \text{ vector all monomials of degree 2 to d}$$

The i^{th} entry of $p(t)$ is the following monomial:

$$p_i = \xi^{\alpha_i} = x_1^{\alpha_i(1)} \dots x_{n_a}^{\alpha_i(n_a)} u^{\alpha_i(n_a+1)}$$

with:

$$\alpha_i \triangleq [\alpha_i(1), \dots \alpha_i(n_a+1)], \text{vector of monomial exponents}$$

Each monomial is of degree 2 to d, meaning $\sum_{j=1}^{n_a+1} \alpha_i(j) \in [2, d]$.

We can finally write:

$$\begin{aligned}\frac{\partial p}{\partial x} &= \left[\frac{\partial p_i}{\partial x_j}\right], i = 1 \dots n_p, j = 1 \dots n_a \\ \frac{\partial p_i}{\partial x_j} &= \alpha_i(j) p_i x_j^{-1}, \text{ if } p_i \text{ contains } x_j \\ &= 0, \text{ otherwise}\end{aligned}$$

Expansion of $\frac{\partial E}{\partial E_{i,j}} p$*:*

$$\frac{\partial E}{\partial E_{i,j}} = \begin{pmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{pmatrix}$$

is a sparse matrix of n_a rows by n_p columns, with 1 at row i, column j and 0 elsewhere.

$$\frac{\partial E}{\partial E_{i,j}} p = (0 \dots 0, p(j), 0 \dots 0)^T.$$

is then a vector of n_a elements with p(j) at position i and zero elsewhere.

Expansion of $\frac{\partial F}{\partial F_j} p$*:*

Similarly as above:

$$\frac{\partial F}{\partial F_j} = (0 \dots 0, 1, 0 \dots 0)$$

is a sparse row vector of n_p elements, with 1 at column j and 0 elsewhere.

Multiplication of that vector to p extracts the j^{th} entry:

$$\frac{\partial F}{\partial F_j} = p_j$$

Algorithm

In this algorithm, we choose to minimize the relative rms error: $\rho(\theta) \triangleq \frac{\sqrt{2f(\theta)}}{\|y\|} = \frac{\|r\|}{\|y\|}$ instead of $f(\theta)$ because it is easier to set limits on it. begin

θ= 0, initial parameters;

calculate residual r(θ), relative rms error $\rho(\theta) = \|r\|/\|\tilde{y}\|$;

calculate Jacobian J, gradient $\nabla f(\theta) = J^T r$, Hessian $H = J^T J$;

$\lambda = 1, i = 1, \alpha = 10$;

$\delta = 10^{-3}$; %to be adjusted

imax= 10; %to be adjusted

$\theta' = \theta - (H + \lambda.Diag(H))^{-1}\nabla f(\theta)$; % parameters update;

calculate new residual r(θ'), new error $\rho(\theta')$;

while ($\rho(\theta') > \delta$) and ($i < i_{max}$)

if $\rho(\theta') \geq \rho(\theta)$;

$\lambda = \lambda.\alpha$; % increase λ;

% and do nothing else: new parameters are rejected;

else;

$\lambda = \lambda/\alpha$; % decrease λ;

θ= θ'; % accept new parameters;

calculate residual r(θ), error $\rho(\theta)$;

calculate Jacobian J, gradient $\nabla f(\theta)$, Hessian H;

end ;

$\theta' = \theta - (H + \lambda.Diag(H))^{-1}\nabla f(\theta)$; %parameters update;

calculate new residual r(θ'), new error $\rho(\theta')$;

i= i+1;

end ;

θ= θ' %final parameters estimate ;

end.

The algorithm presented is inspired from [29,30]. As shown in [29], the parameters update equation: $\theta' = \theta - (H + \lambda.Diag(H))^{-1}\nabla f(\theta)$ is a combination of Gradient descent and Gauss-Newton method.

Validation

For validation of the Levenberg-Marquardt algorithm, we use a simulated non-linear model based on a state-space representation of an existing loudspeaker.

$$\dot{z} = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{k}{m} & -\frac{r}{m} & \frac{Bl}{m} \\ 0 & -\frac{Bl}{L} & -\frac{R}{L} \end{pmatrix} z + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{L} \end{pmatrix} u \triangleq Az + Bu \tag{4.23}$$

- m= 14.35 g
- r= 0.786 N.s/m
- k= 1852 N/m
- Bl= 4.95 N/A
- L= 266 μH
- R= 3.3 Ohms

To get the cone velocity, we use the output equation given by (2.4). The transfer function is multiplied by s in the Laplace domain to get the acceleration which is proportional to the acoustic pressure. That yields a new set of state-space matrices for acoustic simulation of loudspeaker. That set is translated in the digital domain using bilinear transform (Tustin method). For the non-linearity, we use an order 2 monomial vector: $p(t) = x_{\{2\}}(t)$. The non-linear parameter matrices E, F are set as

uniformly equal to 10^{-5}.

The stimulus used is a full multi tone of unity spectrum magnitude and random phases uniform on $[-\pi, \pi[$.

4.3.2 Identification of Continuous-Time Model

The methods described above are aimed at Discrete-Time (DT) models identification. Special care must be taken to apply them to Continuous-Time (CT) model. First, the assumption is made that the output signal power is band-limited. It is a perfectly reasonable assumption for loudspeaker study. Then the Bilinear Transform (BT) is chosen to relate DT to CT. It is known that he use of BT allows to identify continuous-time system in the discrete-time domain without loss of precision. Definition of BT:

$$s = \frac{2}{T}\frac{z-1}{z+1} \iff z = \frac{2+Ts}{2-Ts}$$

where:

s= complex frequency (CT)

z= complex frequency (DT)

T= sampling period

Then the relation between CT and DT transfer functions is the following:

$$G(s) = G(\frac{2}{T}\frac{z-1}{z+1}) = G_d(z)$$

The BT relates the CT frequency to the DT frequency in the following manner:

$$z = e^{j\omega^d} \implies s = j\omega = \frac{2}{T}\frac{e^{j\omega^d}-1}{e^{j\omega^d}+1} = j\frac{2}{T}\tan(\frac{\omega^d}{2})$$
$$\omega = \frac{2}{T}\tan(\frac{\omega^d}{2}) \iff \omega^d = 2\arctan(\frac{T}{2}\omega)$$

where the superscript d denotes the discrete domain. CT and DT frequencies are related by a frequency warping that maps the range $[-\infty, +\infty]$ to $[-\pi, +\pi]$.

Moreover, the CT and DT state-space matrices are related by [31]:

$$\begin{aligned} A &= \frac{2}{T}(A^d + I)^{-1}(A^d - I) \\ B &= \frac{2}{\sqrt{T}}(A^d + I)^{-1}B^d \\ C &= \frac{2}{\sqrt{T}}C^d(A^d + I)^{-1} \\ D &= D^d - C^d(A^d + I)^{-1}B^d \end{aligned} \tag{4.24}$$

Then the identification procedure for CT system is the following:

1. Measure M samples of the FRF from DC to Nyquist.

   $G_k \triangleq G(j\omega_k), \quad k = 0 \dots M-1$

2. Apply BT to get the DT equivalent of the FRF samples:

   $G_k^d \triangleq G^d(e^{j\omega_k^d}) = G(j\omega_k), \quad k = 0 \dots M-1$

   with:

   $\omega_k^d = 2\arctan(\frac{T}{2}\omega_k)$

3. Resample G^d at equidistant intervals

4. Extend G^d on full circle $[-\pi, +\pi]$

5. Estimate the DT state-space matrices A^d, B^d, C^d, D^d by Frequency Subspace Identification

6. Recover the CT state-space matrices A, B, C, D using the relations (4.24)

4.4 Simulations and Experimental Results

This section produces results that illustrate the identification procedure.

4.4.1 Simulations

First, the FRF of the loudspeaker simple model is identified using BLA followed by frequency subspace identification technique. The continuous time model is transferred to the digital domain by bilinear transformation. The BLA measurement is simulated by applying a input excitation into the state-space model of the loudspeaker, and estimating the transfer function between input and output. The input signal is a full multi-sine with flat spectrum and random phase:

$$u(t) = \frac{1}{\sqrt{N}} \sum_{n=-N/2}^{N/2} e^{j(\frac{2\pi}{N} tn + \phi_n)}$$

with:

$$\phi_n = \text{random variable uniform over} [0, 2\pi]$$

$$\phi_{-n} = \phi_n^*$$

$$\phi_0 = \phi_{N/2} = 0$$

Although the power spectrum of a multi-sine is perfectly flat, the amplitude distribution is Gaussian, due to the random phases. That makes it a suitable signal to test nonlinear systems. Fig. 4.13, 4.14, 4.15 show respectively the time domain, histogram and spectrum for an example of stimulus.

In our simulation, the FRF estimate is obtained by repeating the multi-sine twice and using only the second block, as to avoid the transient state and measure the system in its steady state. The duration of each block is verified to be much greater

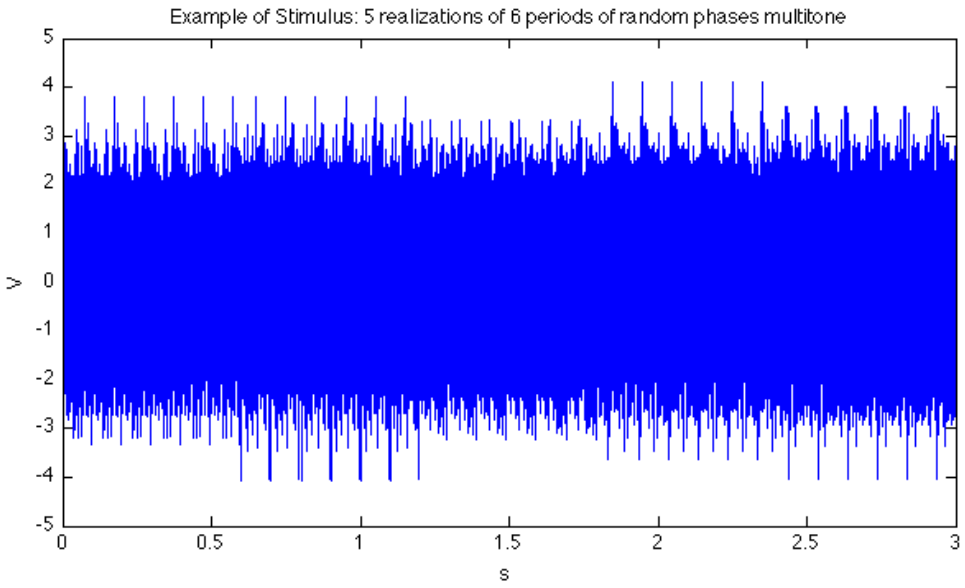

Figure 4.13: Random Phases Multisine- Time domain

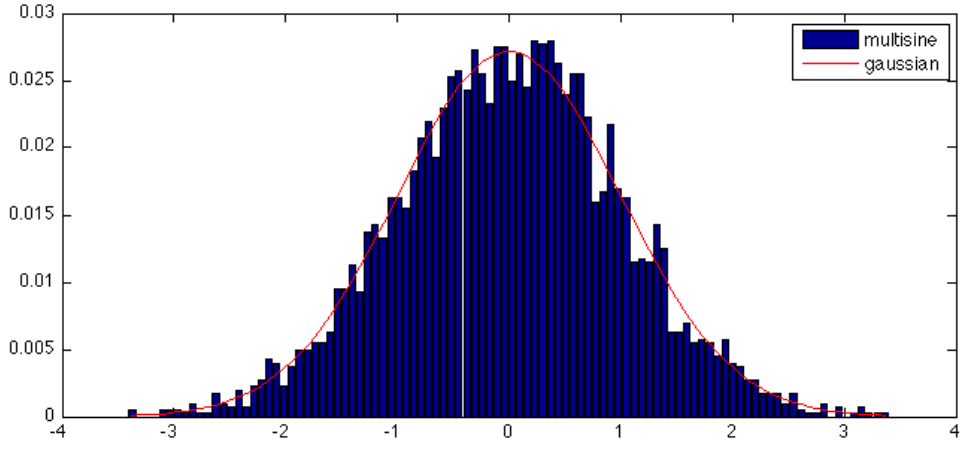

Figure 4.14: Random Phases Multisine- Histogram

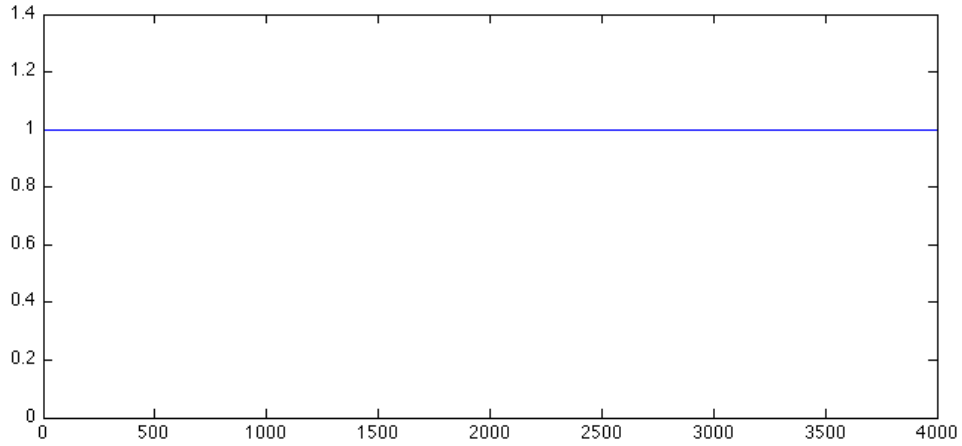

Figure 4.15: Random Phases Multisine- Spectrum

than the impulse response. The FRF is then simply obtained by dividing the DFT of the output by the DFT of the input. The FRF obtained is then fed into the Frequency Subspace Identification algorithm to obtain a state-space estimate. The state-space model is then used to obtain an FRF estimate which is compared with the original one. The results are presented in Fig. 4.16. The real FRF, the measured and the identification results are superimposed. The measurement error is about 300 dB down and the identification error is about 200 dB down. The measurement error curve is defined as:

$$err(\omega) = 20Log_{10}\frac{|\hat{G}(j\omega) - G(j\omega)|}{|G(j\omega)|} \tag{4.25}$$

where G is the measured FRF and $\hat{G}$ is the estimated one.

These results validate the Frequency Subspace Identification procedure. The next step is to examine how the frequency identification behaves in presence of nonlinear perturbations. For this purpose a PNLSS model (4.21) is constructed by adding a state monomial of degree two weighted by E and F where all the coefficients are set to -10^{-5}. The input signal is made of several consecutive blocks of multi-sine,

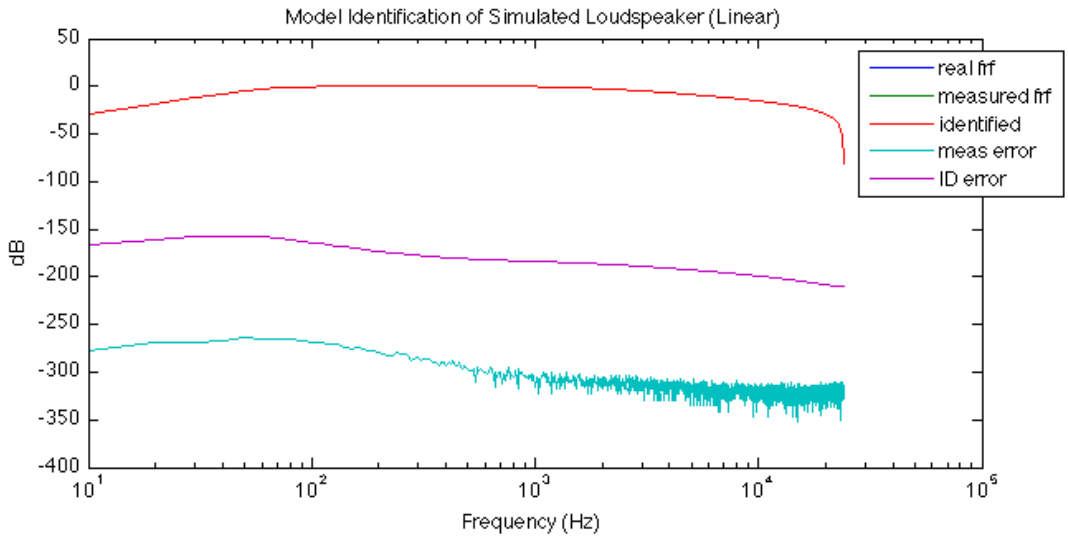

Figure 4.16: Model Identification of Simulated Loudspeaker (Linear)

each of which with a different phase pattern so as to excite the nonlinearity in a random fashion. The BLA of the FRF is obtained by applying a Welch algorithm on the input-output pair. The DFT of each pair of input-output blocks are calculated without overlap and with a flat windowing. The FRF is estimated by taking the quotient of the input-output cross-spectrum over the input auto-spectrum:

$$G_{uy}(j\omega) = \frac{\overline{U^*(j\omega)Y(j\omega)}}{\overline{U^*(j\omega)U(j\omega)}}$$

That method is known to give the BLA of the FRF in presence of output noise. That choice is justified by the fact that the random phase multi-sine has a Gaussian amplitude distribution. The nonlinearity acts then as an added noise source at the output. The FRF is then feed into the Frequency Identification algorithm as before and the FRF of the identified system is compared to the real (linear model) FRF. The results are shown in fig. 4.17 for an input rms level of 10^{-3}.

Due to the nonlinearity , the measured FRF and the linear FRF are now different. The measurement error, which is the difference between the two, shows the added nonlinear noise. The noise spectrum has a level between -30 and -60 dB. The identi-

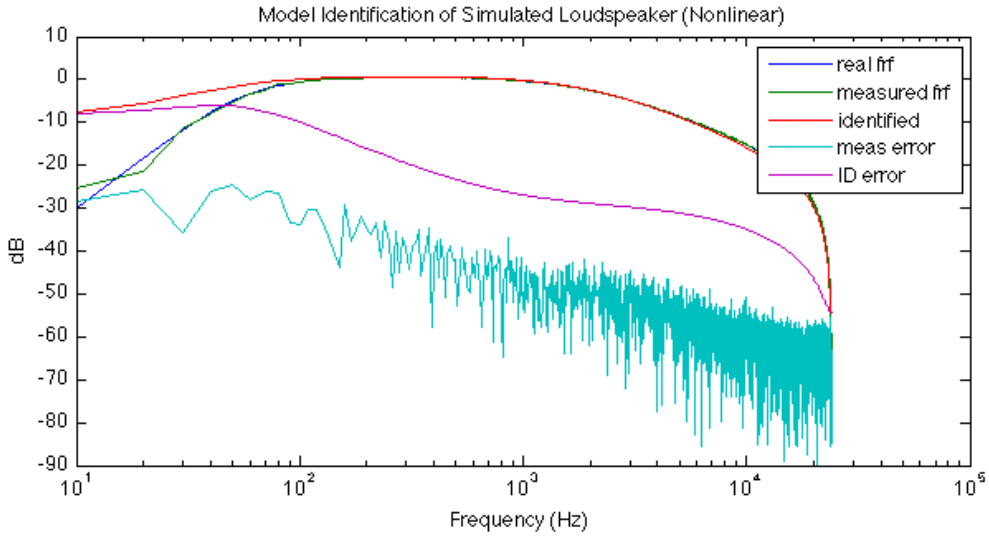

Figure 4.17: Model Identification of Simulated Loudspeaker (Nonlinear)

fication error is now much higher than in the previous experiment, and is especially pronounced in the low frequencies (below 100 Hz). That error follows the spectral distribution of the nonlinear noise.

Finally, to illustrate the last part of our three steps identification procedure the BLA estimated from the nonlinear simulation is used as a basis for nonlinear optimization to estimate the matrices of nonlinear coefficients E and F of the full PNLSS identification model. For this purpose we apply a multi-sine signal to the simulated nonlinear loudspeaker. The output response is then used in the nonlinear Least-Squares algorithm along with the input signal and the estimated linear state-space model parameters. The Levenberg-Marquard algorithm estimates the matrices E and F by successive iterations such that it minimizes the difference between the observed response and the PNLSS model output in the time domain. For the purpose of model validation, a new test signal is applied to the simulated loudspeaker and the PNLSS model to reconfirm the identification process. These results are shown in Fig. 4.18.

The initial relative error between the observed response and the output of the BLA of the loudspeaker is about -22 dB. The estimation relative error is -80 dB, and

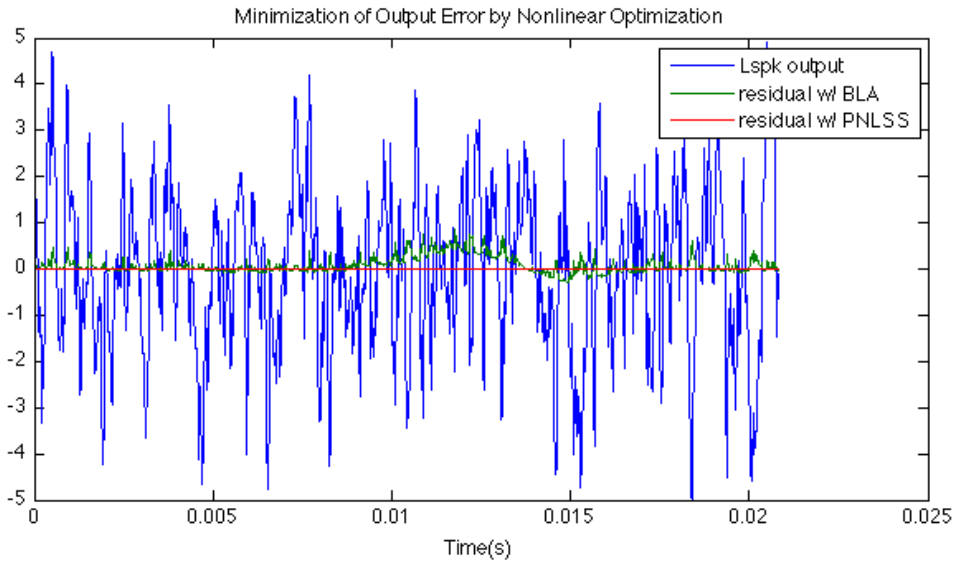

Figure 4.18: Minimization of Output Error by Nonlinear Optimization

the validation relative error is -69 dB. The estimated PNLSS model output matches extremely well with the simulated nonlinear loudspeaker response. This shows the validity of our approach which has potential to be employed in loudspeaker technology.

4.4.2 Experimental Results

As an experiment, a random phase multitone has been fed into a loudspeaker of 2" diameter. The stimulus contained 5 random phase realizations of 6 period each (see fig. 4.13). The excitation level was 1 Vrms to ensure moderate nonlinearities and avoid clipping. After averaging the resulting BLA has been submitted to the frequency subspace identification algorithm (see section 4.2.3) several times with increasing model order. The relative mean square error (RMSE) is calculated for each model order as:

$$rmse = 20Log_{10}\frac{\|\hat{G} - G\|}{\|G\|} \tag{4.26}$$

where G is the measured FRF and $\hat{G}$ is the estimated one. RMSE qualify the quality of the fit and is plotted vs. model order in fig. The complex LS results is shown along

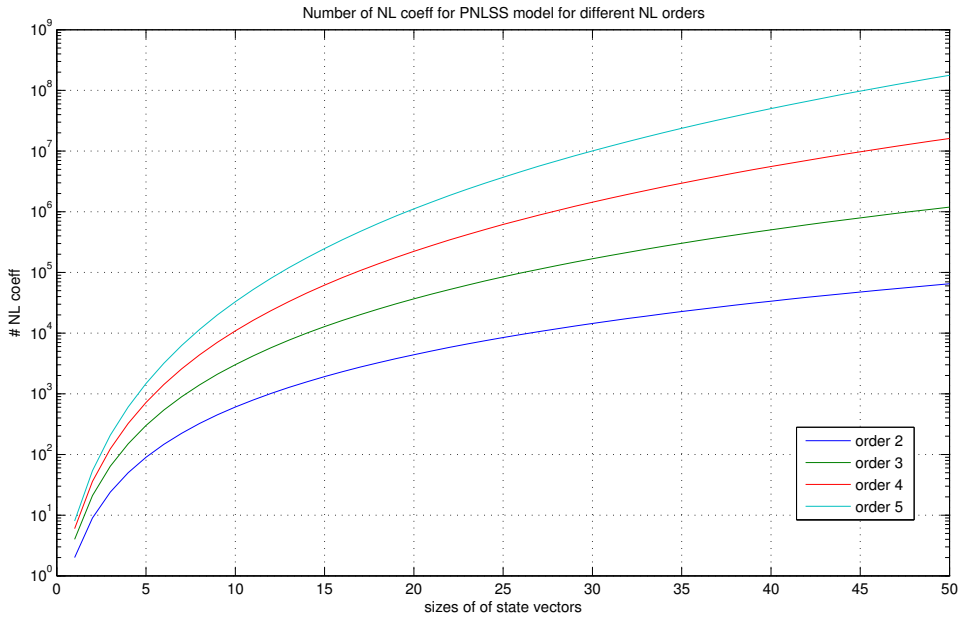

Figure 4.19: Number of NL coeff. vs. model order

for comparison. The Complex LS method [32] is implemented as *invfreqz* in Matlab. Our subspace algorithm behaves clearly better. A minimum RMSE value of -28 dB is obtained for a model order of 76. The magnitude curves and the error curve are shown in fig. 4.20. The result is quite satisfying in term of accuracy. However the model order is too high to pursue the estimation of the nonlinear parameters. The number of nonlinear parameters of the PNLSS model grows considerably with the model order even for low nonlinearity order as can be seen in (4.9). A model order of 76 with a distortion order of three yields 6 083 154 nonlinear coefficients to estimate. The fig. 4.19 shows the evolution of the parameters number in regard to the mode order for a few distortion orders.

To reduce the model complexity, a pole-zero pruning has been applied on the estimated FRF, by:

- deleting poles and zeros far away from the unit circle, which have little influence

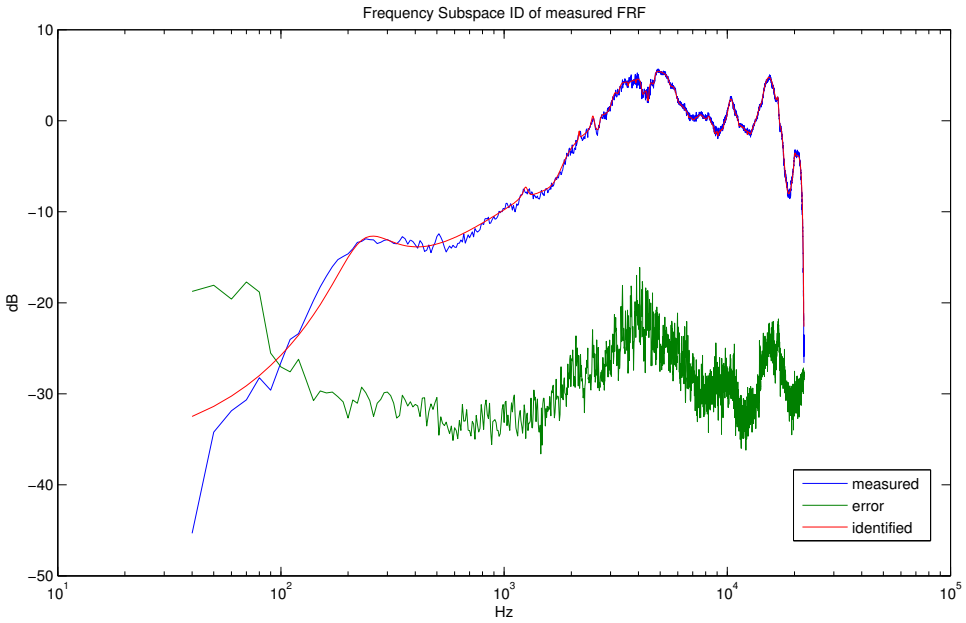

Figure 4.20: FRF Identification (order 76) on measured FRF

on the FRF

- deleting the pairs of pole-zero close to the unit circle, which create small disturbances on the FRF

In addition, to ensure the stability of the estimated model the poles outside of the unit circle are pushed back inside. That has no consequence on the FRF. These methods are detailed in chapter 6 . After model reduction, the FRF model was down from 76 to 22, with an RMSE of -26 dB. The resulting curves are shown in fig. 4.21. We see the error curve is more noisy as a consequence of the FRF smoothing, which is not problematic. However, we see an error appear in the low frequency around 200 Hz where the FRF is oversimplified. We are at the limit of the model reduction procedure and the number of parameters to identify is still very high: 52 371, for a distortion order of only three.

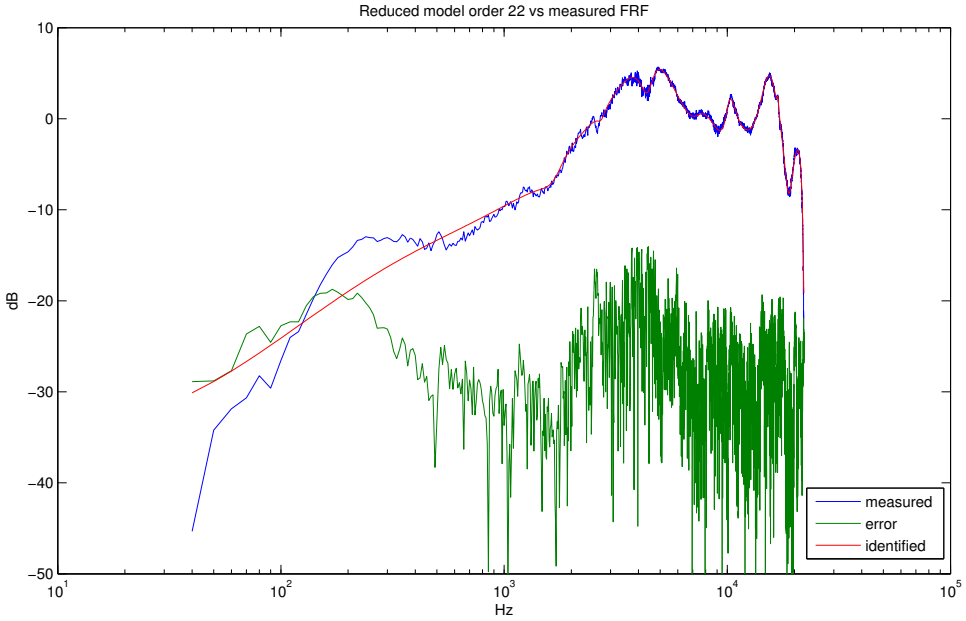

Figure 4.21: Reduced order FRF (order 22) vs. measured FRF

4.5 Discussion

Our experiment brought us to the conclusion that the loudspeaker as a whole needs at least about 20 states variables to model correctly its linear part and, even for low order of nonlinearity, the number of monomials needed for a polynomial state-space model is very high. As an example, a nonlinear order of three and 20 states variables leads to 1750 monomials and 42483 parameters to identify [20]. The NL optimization procedure cannot be applied. The system complexity has to be further reduced. One way to proceed is to consider the loudspeaker as two systems in cascade: the electromechanical part (the 'motor'), that transforms the input voltage (audio signal) into displacement and the mechanico-acoustic part that transforms the displacement into acoustic wave (the 'cone'). The hope is to be able to lower the complexity of the parts to identify and then be able to apply the polynomial state-space system identification on each part separately.

Chapter 5

Fractional Order Model

In this chapter we focus on the motor identification. As we will show in the following, the linear identification of the motor presents by itself specific challenges that requires innovative solutions. The contribution of this chapter is the application of fractional calculus to loudspeaker identification. To the knowledge of the authors, this is a novel approach in that domain [33].

5.1 Fractional Order System

First we introduce the notion of Fractional Order System that will be of use in the following sections.

5.1.1 Fractional Order Derivatives

The fractional order calculus has a long history and mathematicians tried to develop theoretical results for it. Its development for dynamic systems is based on the generalization of the differential operator $D_t = \frac{\mathrm{d}}{\mathrm{d}t}$ to $D_t^\alpha = \frac{\mathrm{d}^\alpha}{\mathrm{d}t^\alpha}$, where α could be a fractional number, and for that matter, any non-zero real number. The advantages of fractional derivatives become apparent in modeling of physical processes. Fractional derivatives

and integrals also appear in the control of dynamic systems, when the system under control or/and the controller is described by a fractional differential equation. The mathematical modeling and simulation of systems described by fractional derivatives leads to differential equations of fractional order and the necessity of solving such equations. For a general exposition of fractional systems one may refer to [34,35] and the references therein.

There are three equivalent definitions most frequently used for fractional derivative of a function: Grunwald-Letnikov, Riemann-Liouville, and Caputo. The Riemann-Liouville approach starts with generalization of Cauchy's formulae for repeated integration to non-integer orders, defining the fractional order integration as:

$$J_t^{\alpha} f(t) = D_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t f(\tau)(t-\tau)^{\alpha-1} \, \mathrm{d}\tau \tag{5.1}$$

where $\Gamma(.)$ is the Gamma function defined by the following expression known as factorial function

$$\Gamma(\alpha) = \int_0^{\infty} e^{-t} t^{\alpha-1} dt \tag{5.2}$$

for which, when α is an integer, it reduces to the conventional factorial i.e. $\Gamma(\alpha+1) = \alpha!$ The definition of fractional derivative can easily be derived by taking an n^{th} order derivative of an α^{th} order integral to obtain an $n - \alpha = q$ order derivative:

$$D_t^q f(t) = D_t^{n-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} D_t^n \int_0^t f(\tau)(t-\tau)^{\alpha-1} \, \mathrm{d}\tau \tag{5.3}$$

which is the Riemann-Liouville expression of fractional derivative. It should be noted that for $q = 1 (n = 2, \alpha = 1)$, the above expression becomes the usual first order derivative. Furthermore, most properties of conventional (integer) derivatives can be extended to the non-integer order case. Since in the analysis of dynamic systems one can take advantage of Laplace operator to represent the system by transfer function, it is possible to write:

$$L\left\{D_t^{\alpha} f(t)\right\} = s^{\alpha} L\left\{f(t)\right\} - \sum_{k=0}^{n-1} s^k D_t^{\alpha-1-k} f(0) \tag{5.4}$$

which becomes very simple if all the derivatives are zero at the time origin i.e.

$$L\left\{D_t^{\alpha} f(t)\right\} = s^{\alpha} L\left\{f(t)\right\} \tag{5.5}$$

Consequently, the Laplace transforms for various functions and their inverses can be derived. It is worth mentioning that the Caputo's definition of fractional derivative given by:

$$D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha-n+1}} \, \mathrm{d}\tau \tag{5.6}$$

for $n-1 < \alpha < n, n \in \mathbb{Z}$, where derivative and integration are reversed, allows to use integer order initial conditions when applying Laplace transform and solving fractional order differential equations.

5.1.2 Fractional Order System

A FO differential equation with commensurate orders can be described by:

$$\sum_{n=0}^{N} a_n D_t^{n\alpha}(y) = \sum_{m=0}^{M} b_m D_t^{m\alpha}(u) \tag{5.7}$$

The corresponding transfer function can be written as:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{m=0}^{M} b_m s^{m\alpha}}{\sum_{n=0}^{N} a_n s^{n\alpha}} \tag{5.8}$$

with its state-space realization given by:

$$\begin{aligned} D_t^{\alpha} x(t) &= Ax(t) + Bu(t),\ 0 < \alpha \leq 1 \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{5.9}$$

where $x(t) \in \mathbb{R}^N, u(t) \in \mathbb{R}, y(t) \in \mathbb{R}$, are respectively the state, input, and output and $A \in \mathbb{R}^{N\times N}, B \in \mathbb{R}^{N\times 1}, C \in \mathbb{R}^{1\times N}, D \in \mathbb{R}$, the state-space parameters matrices.

Definition 1. *The Mittag-Leffler function is defined by:*

$$E_{\alpha\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)} \tag{5.10}$$

where α, β are complex parameters with $Re(\alpha), Re(\beta) > 0$.

The Mittag-Leffler function is a generalization of the exponential function:

$$E_{11}(z) = e^z$$

Theorem 1. *The solution of* (5.9) *is given by:*

$$x(t) = \Phi_0(t)x(0) + \int_0^t \Phi(t-\tau)Bu(\tau)\mathrm{d}\tau, \tag{5.11}$$

where

$$\Phi_0(t) = \sum_{k=0}^{\infty} \frac{A^k t^{k\alpha}}{\Gamma(k\alpha+1)} = E_{\alpha 1}(At^\alpha) \tag{5.12}$$

$$\Phi(t) = \sum_{k=0}^{\infty} \frac{A^k t^{k\alpha+\alpha-1}}{\Gamma(k\alpha+\alpha)} = t^{\alpha-1}E_{\alpha\alpha}(At^\alpha) \tag{5.13}$$

It is interesting to see that, when $\alpha = 1$, the solutions above reduce to the well-known: $\Phi_0(t) = \Phi(t) = E_{11}(At) = e^{At}$. The proof is constructive.

5.1.3 Stability Analysis

The state equation corresponding to (5.9) in Laplace domain can be written as:

$$\begin{aligned} s^{\alpha}X(s) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s) \end{aligned} \tag{5.14}$$

Where: s is the Laplace complex frequency, α the FO ($\alpha \in \mathbb{R}^*$), U, X, Y respectively input, state, output, and A, B, C, D the state-space parameters matrices.
The characteristic polynomial of (5.14) is given by:

$$w(s) = det(s^{\alpha}I - A) = a_n s^{n\alpha} + a_{n-1}s^{(n-1)\alpha} + \cdots + a_0 \tag{5.15}$$

with its associated natural degree polynomial written as:

$$\tilde{w}(\lambda) = a_n\lambda^n + a_{n-1}\lambda^{(n-1)} + \cdots + a_0,\ \lambda = s^{\alpha} \tag{5.16}$$

Theorem 2. *The FO system* (5.9) *is stable iff the fractional degree characteristic polynomial* (5.15) *has no zeros in the closed right half of the Riemann complex surface, i.e.*

$$w(s) = det(s^{\alpha}I - A) \neq 0,\ Re(s) \geq 0 \tag{5.17}$$

or equivalently, the following condition is satisfied

$$|arg\lambda_i(A)| > \alpha\frac{\pi}{2},\ i = 1, 2, \ldots n \tag{5.18}$$

where $\lambda_i(A)$ is the i^{th} eigenvalue of matrix A.

Remark: it can be shown [36] that the fractional system with the characteristic polynomial (5.15) is unstable for all $\alpha > 2$. Therefore, the stability of fractional system in this paper is considered for $\alpha \in [0, 2]$. Furthermore, the stability analysis should be divided in two different interval $0 < \alpha \leq 1$ and $1 \leq \alpha < 2$ as it can be seen in the following analysis.

First, one can immediately specify that the FO system (5.14) is stable if and only if the eigenvalues of the matrix A lie in the specified regions according to the intervals $0 < \alpha \leq 1$ and $1 \leq \alpha < 2$ as shown in fig. 5.1 and fig. 5.2.

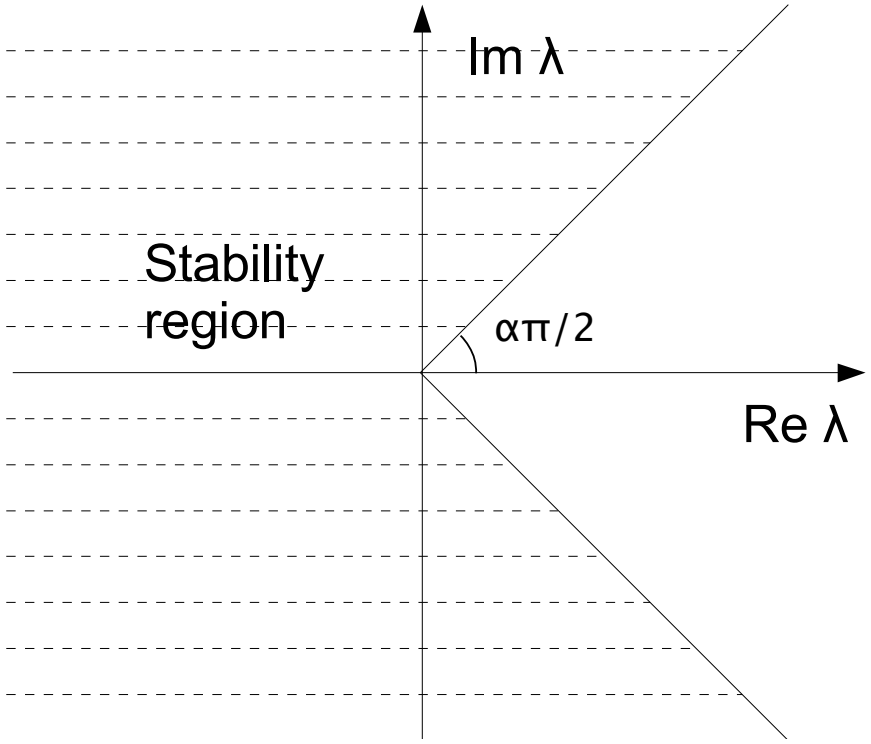

Figure 5.1: **Stability region for** $0 < \alpha \leq 1$

With the aid of [36] and [37], the following theorems can be written with respect to the state-space representation of fractional systems.

Theorem 3. *The fractional system* (5.14) *with* $1 \leq \alpha < 2$ *is stable if and only if one of the following equivalent conditions is satisfied:*

1. *The eigenvalues of the matrix*

$$\bar{A} = \begin{bmatrix} A\,sin(\alpha\,\pi/2) & A\,cos(\alpha\,\pi/2) \\ -A\,cos(\alpha\,\pi/2) & A\,sin(\alpha\,\pi/2) \end{bmatrix}$$

have negative real parts.

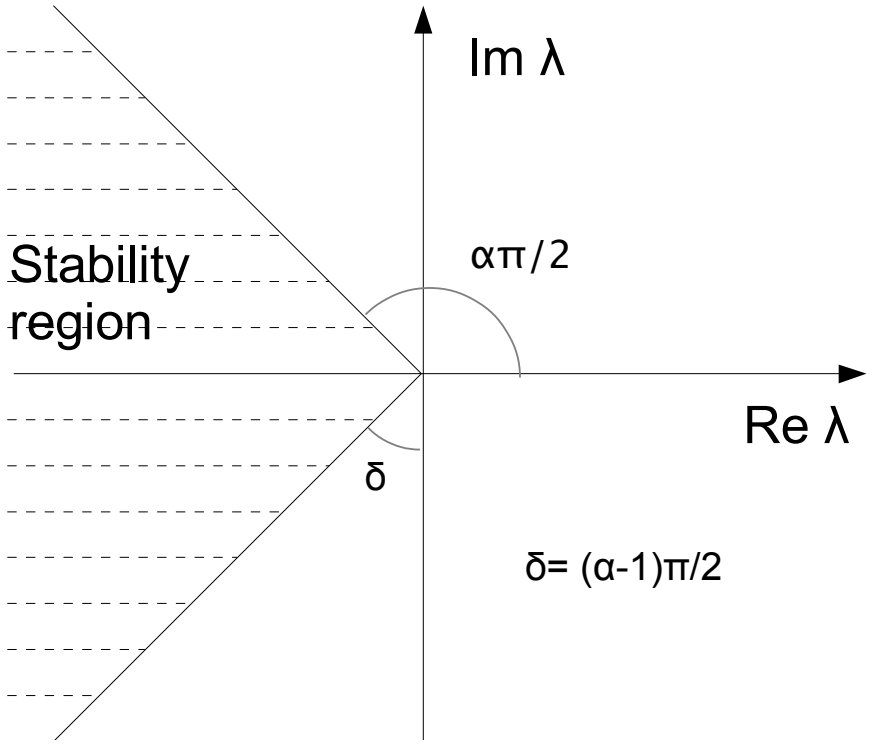

Figure 5.2: **Stability region for** $1 \leq \alpha < 2$

2. *There exist positive definite Hermitian matrices* $P > 0$ *and* $Q > 0$ *such that*

$$zPA + z^*A^TP = -Q$$

where $z = v + jw$

with $\tan(\pi - \alpha\pi/2) = v/w$

3. *There exists a matrix* $P = P^T > 0$ *such that the LMI*

$$\begin{bmatrix} (AP + PA^T)sin(\alpha\pi/2) & (AP - PA^T)cos(\alpha\pi/2) \\ (PA^T - AP)cos(\alpha\pi/2) & (AP + PA^T)sin(\alpha\pi/2) \end{bmatrix} < 0$$

is feasible.

Theorem 4. *The fractional system* (5.14) *with* $0 < \alpha \leq 1$ *is unstable and all eigenvalues of* A *lie in the instability region shown in fig. 5.1 if and only if the eigenvalues of* $\tilde{A}$ *have negative real parts, where*

$$\tilde{A} = \begin{bmatrix} -A\,sin(\alpha\pi/2) & A\,cos(\alpha\pi/2) \\ -A\,cos(\alpha\pi/2) & -A\,sin(\alpha\pi/2) \end{bmatrix}$$

or (5.14) *is stable if and only if there does not exist any non-negative rank one complex matrix* Q *such that*

$$rAQ + QA^T\bar{r} \geq 0$$

where $r = sin(\alpha\pi/2) + jcos(\alpha\pi/2)$ *and* $\bar{r}$ *denotes the complex conjugate of* r.

Corollary 1. *The fractional system* (5.14) *with* $0 < \alpha \leq 1$ *is stable if there exist positive definite matrices* $X_1 = X_1^*$ *and* $X_2 = X_2^*$ *such that*

$$\bar{r}X_1A^T + rAX_1 + rX_2A^T + rAX_2 < 0$$

where $r = exp(j(1 - \alpha)\pi/2)$.

The stability analysis provides us guidelines for the choice of the fractional order of the model to identify. Considering the fact that a loudspeaker is a stable system, the choice of $\alpha = 1/2$ seems reasonable. Another important point is that this analysis provides also guidelines to eventually stabilize the identified model. In fact, due to measurement noise and nonlinear distortion, frequency identification algorithms can deliver a model which is not stable [38]. Different methods exist for stabilization, including stability constraints added to the algorithm [25], added delay applied to the measured data [39], pruning and mirroring of the unstable poles [40], and separation of stable part through decomposition method (see chapter 6). The specific nature of fractional order system has to be taken into account when applying these methods.

Let us assume that the outcome of the subspace identification method leads to the realization $\{A, B, C, D\}$ with unstable and stable eigenvalues associated with the matrix A. As pointed out above, several algorithms exist for extracting stable identified model. A simple method is separation of stable and unstable part of the system by using spectral projection for block diagonalization. Sign function based spectral projection method, for example, is computationally efficient to perform this task. So one can eventually use the orthogonal transformation matrix to transform $\{A, B, C, D\}$ to $\{\bar{A}, \bar{B}, \bar{C}, \bar{D}\}$ such that

$$\bar{A} = UAU^{-1} = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}$$

$$\bar{B} = UB = \begin{pmatrix} B_1 & B_2 \end{pmatrix}^T$$

$$\bar{C} = CU^{-1} = \begin{pmatrix} C_1 & C_2 \end{pmatrix}$$

$$\bar{D} = D$$

which yields the desired additive decomposition

$$G(s) = C(sI - A)^{-1}B + D = \bar{C}(sI - \bar{A})\bar{B} + \bar{D} = G_1(s) + G_2(s)$$

where the stable and unstable part are, respectively,

$$G_1(s) = C_1(sI_k - A_{11})^{-1}B_1 + D$$
$$G_2(s) = C_2(sI_{n-k} - A_{22})^{-1}B_2 + D$$

Consequently, extracting the stable part and ignoring the unstable part of this decomposition leads to stable approximation which is the optimal L_2 norm approximant. A second option is replacing the unstable matrix A_{22} with A_{22}^{-1} in the z-domain. The equivalent in the s-domain is flipping all the unstable poles around the imaginary axis. The position of zeros remain the same.

5.1.4 Illustrative Example

As an illustrative example we will examine hereafter the response of a simple fractional order circuit, as shown in fig. 5.3, where $F = 1/Cs^{\alpha}$ When $\alpha = 1$, F is an

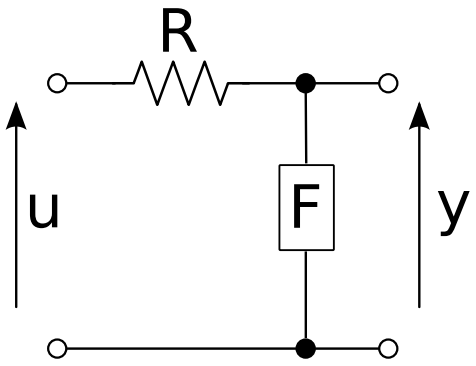

Figure 5.3: **Fractional order circuit**

ordinary capacitance and we have a classical RC circuit. When $\alpha \neq 1$, F is called a

fractance or constant phase device.

Frequency Response

In the Laplace domain, we have straightforwardly:

$$H_\alpha(s) \triangleq \frac{Y(s)}{U(s)} = \frac{1}{1 + RCs^\alpha} \tag{5.19}$$

And then the frequency response:

$$H_\alpha(j\omega) = \frac{1}{1 + RC(j\omega)^\alpha} \tag{5.20}$$

For $\alpha = 1$, we have the familiar first order low-pass filter:

$$H_1(j\omega) = \frac{1}{1 + RCj\omega}$$

with the cut-off frequency $\omega_0 = {}^1/_{RC}$, and an asymptotic attenuation rate of -20 dB/decade for frequencies beyond.

In the general case ($\alpha \neq 1$) the cut-off frequency can be defined similarly as $\omega_0 = (RC)^{-1/\alpha}$ and the asymptotic attenuation rate is -20α dB/decade. E.g. $\alpha = 1/2 \rightarrow \omega_0 = {}^1/\sqrt{RC}$ and the attenuation rate becomes -10 dB/decade. In fig. 5.4 we show the Bode plots of the capacitance and fractance circuits where the capacitance value C and fractance value C' are scaled in such a way that the two cut-off frequencies

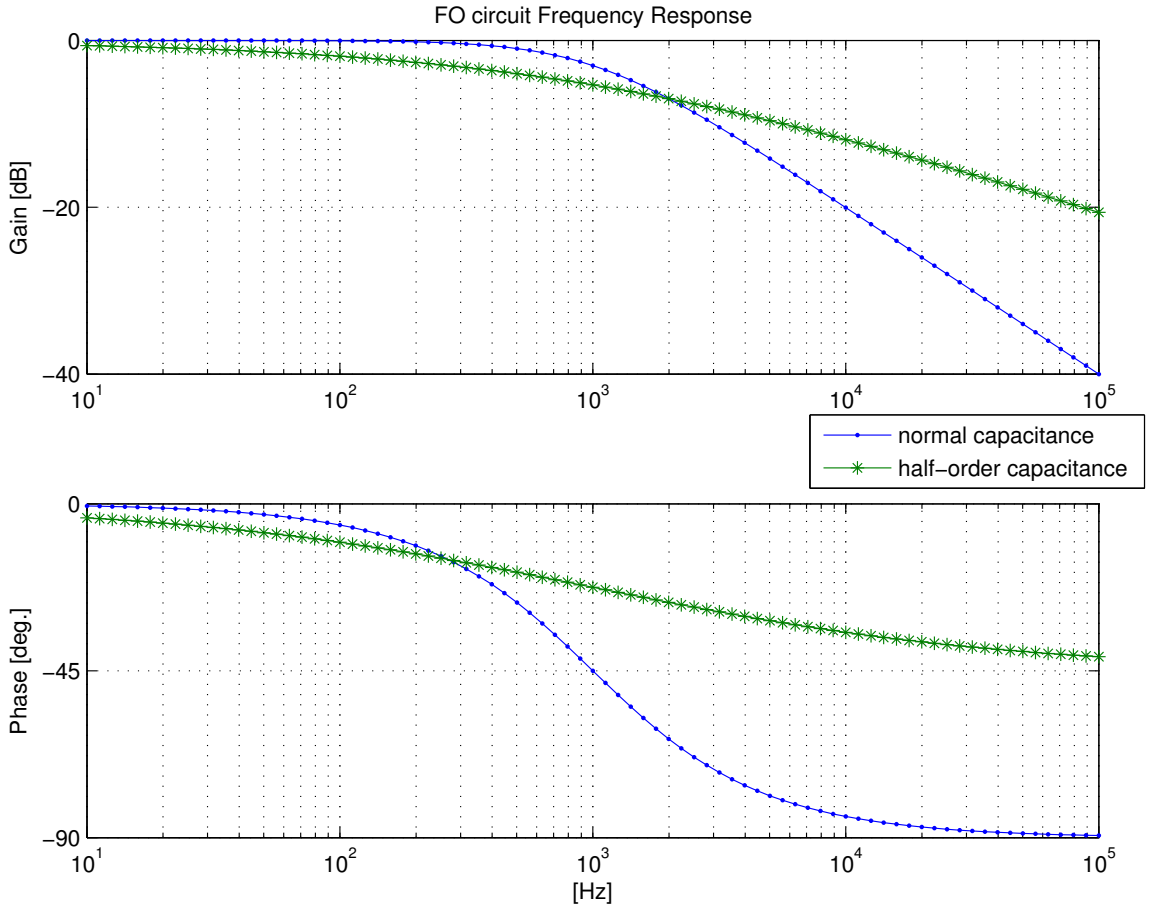

Figure 5.4: **FO circuit Bode plots**

coincide: $RC = \sqrt{RC'} = 1/2000\,\pi$. At the cut-off frequency we have:

$$\begin{aligned} |H(j\omega_0)| &= \frac{1}{\sqrt{2+\cos\alpha\pi}} \\ &= 1/\sqrt{2} \equiv -3 \text{ dB, for } \alpha = 1 \\ &= \frac{1}{\sqrt{2+\sqrt{2}}} \equiv -5.4 \text{ dB, for } \alpha = 1/2 \\ Arg(H(j\omega_0) &= Arg(j^{-\alpha}) = -\alpha\pi/2 \\ &= -\pi/2, \text{ for } \alpha = 1 \\ &= -\pi/4, \text{ for } \alpha = 1/2 \end{aligned}$$

Time Domain Response

The corresponding differential equation for the above circuit is:

$$y(t) + RCD_t^\alpha y(t) = u(t) \tag{5.21}$$

and corresponding state-space representation:

$$D_t^\alpha x = Ax + Bu \tag{5.22}$$

$$y = Cx \tag{5.23}$$

with: $A = -1/RC = -B, C = 1$.

Assuming a relaxed system, $x(0) = 0$ and using Theorem 1, we have:

$$y(t) = \frac{1}{RC}\int_0^t \Phi(t-\tau)u(\tau)\mathrm{d}\tau \tag{5.24}$$

where

$$\Phi(t) = t^{\alpha-1}E_{\alpha\alpha}(-\frac{t^\alpha}{RC}) \tag{5.25}$$

Therefore the impulse response of the circuit is:

$$h_\alpha(t) \triangleq \frac{t^{\alpha-1}}{RC}E_{\alpha\alpha}(-\frac{t^\alpha}{RC}) \tag{5.26}$$

For $\alpha = 1$ (capacitance):

$$h_1(t) = \frac{1}{RC} e^{-t/RC} \tag{5.27}$$

For $\alpha = 1/2$ (fractance):

$$h_{\frac{1}{2}}(t) = \frac{1}{RC\sqrt{t}} E_{\frac{1}{2},\frac{1}{2}}(-\frac{\sqrt{t}}{RC}) \tag{5.28}$$

It is important to see that for $\alpha < 1 : \lim_{0^+} h_\alpha(t) = +\infty$.

That difficulty lets us consider the step response. In the Laplace domain, the step response is:

$$G(s) \triangleq \frac{1}{s} \frac{1}{1 + RCs^\alpha} \tag{5.29}$$

In the litterature [41], it can be found that the step response in the time domain is given by:

$$L^{-1}\{G(s\} = 1 - E_{\alpha,1}(-\frac{t^\alpha}{RC}) \triangleq g_\alpha(t) \tag{5.30}$$

For $\alpha = 1$ (capacitance):

$$g_1(t) = (1 - e^{-t/RC}) \tag{5.31}$$

For $\alpha = 1/2$ (fractance):

$$g_{\frac{1}{2}}(t) = 1 - E_{\frac{1}{2},1}(-\frac{\sqrt{t}}{RC}) \tag{5.32}$$

Using the same relative scaling of capacitance and fractance values as before, we get the following step responses (see fig. 5.5). It is interesting to see that the initial slope is steeper for $\alpha = 1/2$.

5.2 Loudspeaker FO Model

We have now the tools to elaborate a simple fractional order model of loudspeaker. Substituting a fractional inductance into the loudspeaker model differential equations (2.1) and (2.2), we get:

$$u(t) = Ri(t) + L\frac{d^\alpha i}{dt^\alpha} + Bl\frac{dx}{dt} \tag{5.33}$$

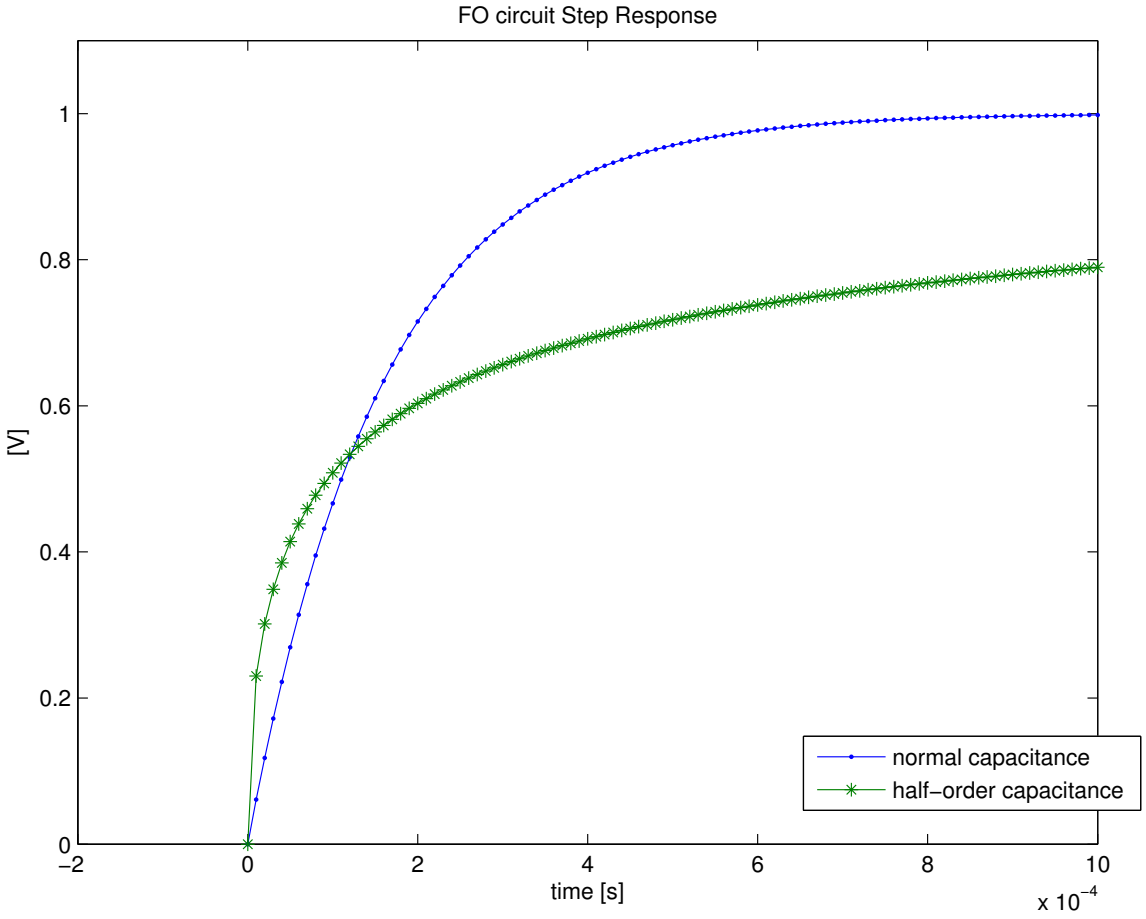

Figure 5.5: **FO circuit step responses**

$$Bli(t) = m\frac{d^2x}{dt^2} + r\frac{dx}{dt} + kx(t) \tag{5.34}$$

In the complex frequency domain:

$$U(s) = RI(s) + Ls^{\alpha}I(s) + BlsX(s) \tag{5.35}$$

$$BlI(s) = ms^2X(s) + rsX(s) + kX(s) \tag{5.36}$$

Algebraic manipulations leads to:

$$Z(s) = \frac{U(s)}{I(s)} = R + Ls^{\alpha} + \frac{Bl^2s}{ms^2 + rs + k} \tag{5.37}$$

$$Y(s) = \frac{I(s)}{U(s)} = \frac{ms^2 + rs + k}{Lms^{2+\alpha} + Rms^2 + Lrs^{1+\alpha} + (Rr + Bl^2)s + Lks^{\alpha} + kR} \tag{5.38}$$

Using the following parameters values (similar to the ones used in chapter 2, we obtain convincing impedance characteristics (fig. 5.6 and fig. 5.7):

- m= 14.35 g
- r= 0.786 kg/s
- k= 1852 N/m
- Bl= 4.95 N/A
- L= 0.1197 μH
- R= 3.3 Ohms
- α= 0.5

To translate the FO differential equations (5.33) and (5.34) to a state-space realization (5.9), we can, in the case of $\alpha = 0.5$ rewrite them as:

$$D_t^2x(t) = -\frac{k}{m}x - \frac{r}{m}D_t^1x(t) + \frac{Bl}{m}i(t)$$
$$D_t^{1/2}i(t) = -\frac{Bl}{L}D_t^1x(t) - \frac{R}{L}i(t) + \frac{1}{L}u$$

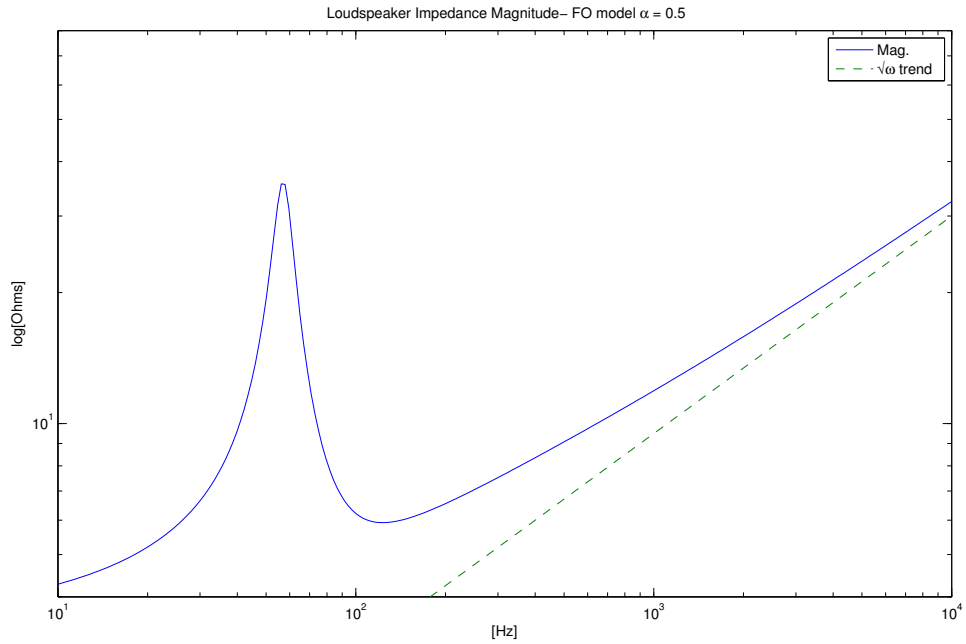

Figure 5.6: Impedance magnitude of FO loudspeaker model (α=0.5)

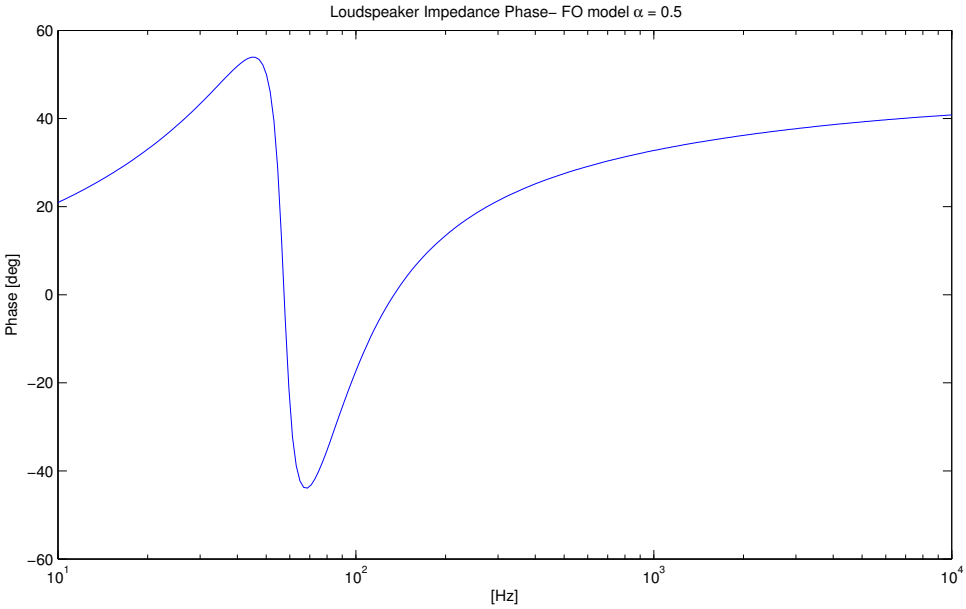

Figure 5.7: Impedance phase of FO loudspeaker model (α=0.5)

Then using the property $D_t^{\alpha+\beta} = D_t^{\alpha} D_t^{\beta}$ [34, 35], and defining the state vector as:

$$[\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5]^T = [x, D_t^{1/2}x, D_t^1 x, D_t^{3/2}x, i]^T$$

we obtain:

$$\begin{aligned}
D_t^{1/2}\zeta_1 &= \zeta_2 \\
D_t^{1/2}\zeta_2 &= \zeta_3 \\
D_t^{1/2}\zeta_3 &= \zeta_4 \\
D_t^{1/2}\zeta_4 &= -\frac{k}{m}\zeta_1 - \frac{r}{m}\zeta_2 + \frac{Bl}{m}\zeta_5 \\
D_t^{1/2}\zeta_5 &= -\frac{Bl}{L}\zeta_3 - \frac{R}{L}\zeta_5 + \frac{1}{L}u
\end{aligned}$$

which can be written in state-space notation as follows:

$$D_t^{1/2}\zeta = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{k}{m} & 0 & -\frac{r}{m} & 0 & \frac{Bl}{m} \\ 0 & 0 & -\frac{Bl}{L} & 0 & -\frac{R}{L} \end{pmatrix} \zeta + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{pmatrix} u \triangleq A\zeta + Bu \tag{5.39}$$

Another realization is possible using:

$$\zeta = [\zeta_1, \zeta_2, \zeta_3]^T \triangleq [x, D_t^1 x, i]^T$$

and:

$$\begin{aligned}
D_t^1\zeta_1 &= \zeta_2 \\
D_t^1\zeta_2 &= -\frac{k}{m}\zeta_1 - \frac{r}{m}\zeta_2 + \frac{Bl}{m}\zeta_3 \\
D_t^{\alpha}\zeta_3 &= -\frac{Bl}{L}\zeta_2 - \frac{R}{L}\zeta_3 + \frac{1}{L}u
\end{aligned}$$

which can be written in state-space notation as follows:

$$D_t^\gamma \zeta = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{k}{m} & -\frac{r}{m} & \frac{Bl}{m} \\ 0 & -\frac{Bl}{L} & -\frac{R}{L} \end{pmatrix} \zeta + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{L} \end{pmatrix} u \triangleq A\zeta + Bu \tag{5.40}$$

with the following definition: $D_t^\gamma \triangleq diag\{D_t^1, D_t^1, D_t^\alpha\}$.

The system:

$$D_t^\gamma \zeta(t) = A\zeta(t) + Bu(t)$$

$$y(t) = C\zeta(t)$$

has the following transfer function:

$$G(s) = C(diag\{S, S, S^\alpha\} - A)^{-1}B$$

This state-space realization is much simpler: the state-vector dimension is three instead of five. It is not using commensurate differentiation orders and therefore allows more freedom in the choice of α. Beyond this simple case, it is worth mentioning that obtaining minimum realization of FO system is the subject of on-going research [42].

5.2.1 Loudspeaker Impedance

As we have seen the loudspeaker can be studied in two parts: the electromechanical part (motor) and the acoustical part (diaphragm). The motor is responsible for the low frequencies behavior and most of the nonlinearities while the diaphragm produces high frequencies irregularities due to modal vibrations. Equations shows that the motor dynamics is fully observable through its electrical response i.e. its impedance curve. In this chapter we will therefore focus on the modeling and identification of the impedance (see fig. 5.8).

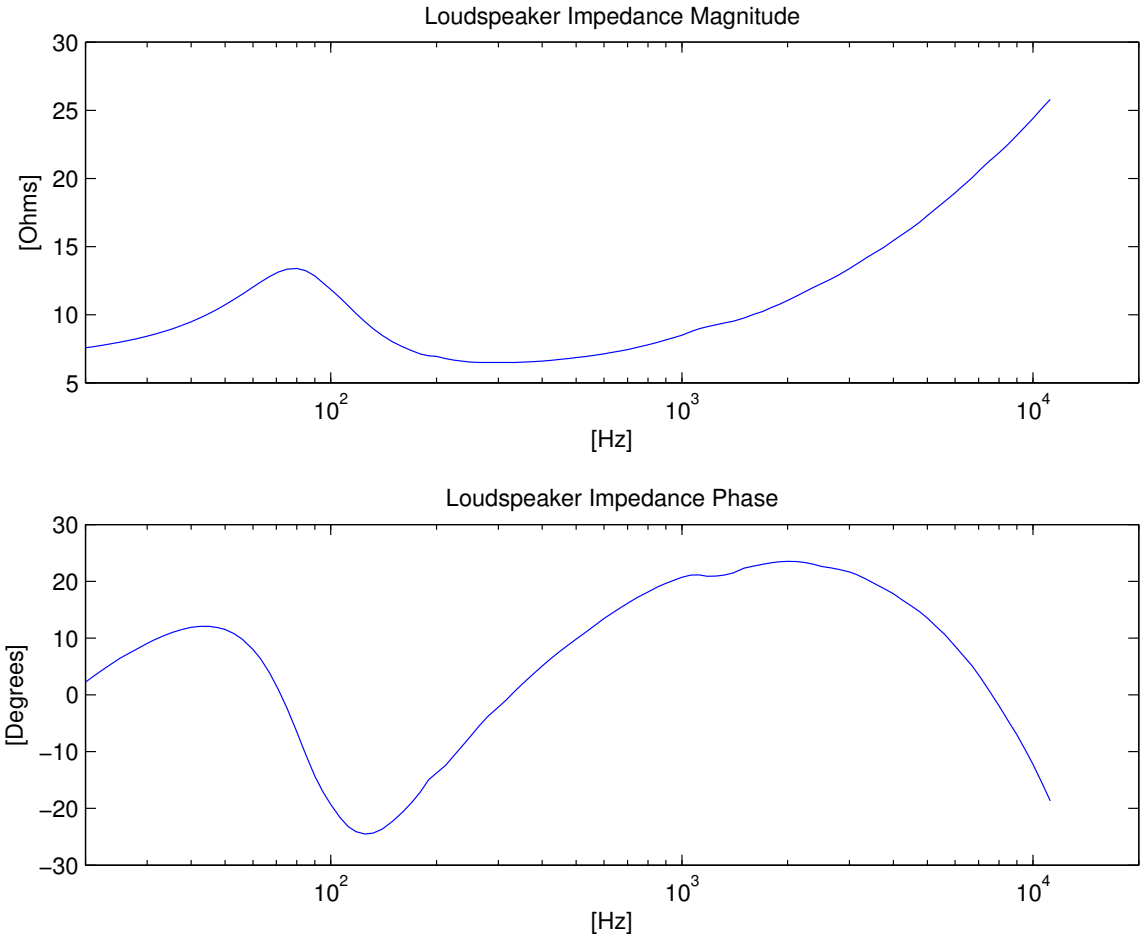

Figure 5.8: **Example of loudspeaker measured impedance.**

5.2.2 Empirical Evidence of Fractional Order

From to (2.1) and (2.2), the electrical impedance can be expressed as:

$$Z(s) = \frac{mLs^3 + (rL + Rm)s^2 + (rR + Bl^2 + kL)s + kR}{ms^2 + rs + k} \tag{5.41}$$

That simplified model is used widely, however many studies have shown its insufficiencies. In particular it doesn't capture the influence of eddy currents in the iron pole structure [43] or the visco-elastic behavior of the suspension ('creep') [44] nor the thermal dependance [45]. Various improved models with added complexity have been proposed [46–48] but there is no general agreement about any of them.

The impedance curves shows three important features, consistent with (5.41):

- The impedance tends toward a non-zero value at low frequencies, i.e. the voice-coil has a DC resistance.
- A strong mode stands out. This is due to the mechanical resonance of the moving part attached to the voice-coil combined with the spider and surround springs.
- At high frequencies there is a continuous rise of the impedance magnitude, which is consistent with an inductive behavior of the voice-coil.

Looking more closely, one can notice something unusual: the phase at high frequency doesn't tend towards 90 degrees as one would expect from an inductance. A linear regression on the log-magnitude of the impedance versus log-frequency, in the high frequencies range shows that the magnitude rises with the square-root of the frequency (see fig. 5.9).

This fact is well-known in the electroacoustic community and has been described in the aforementioned papers [43, 46–48]. The semi-inductive behavior is attributed

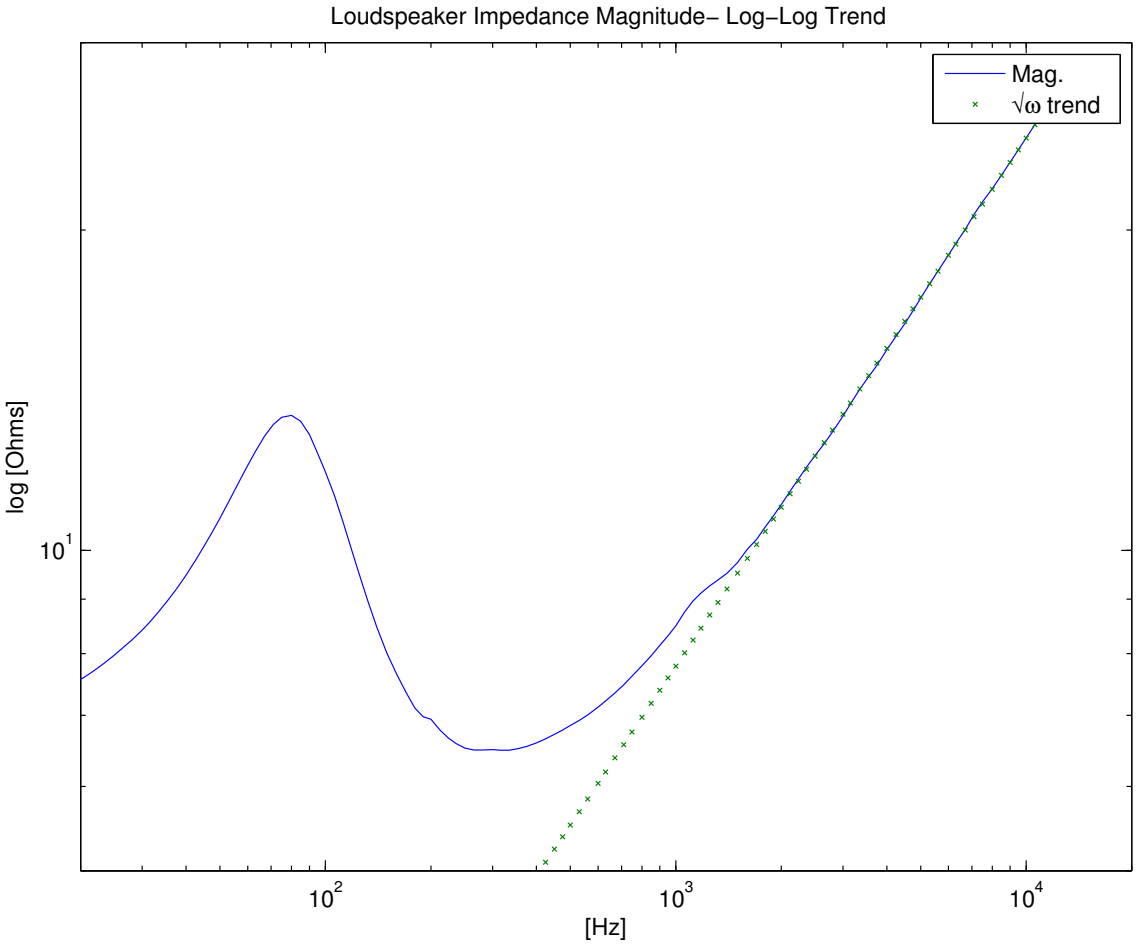

Figure 5.9: **Loudspeaker semi-inductive behaviour (log-log scale).**

to eddy currents in the iron pole structure of the motor and lossy coils have been successfully modeled using fractional derivative [49]. This fact made us believe that the loudspeaker motor is indeed a fractional order system and should be identified as such. Thus, we believe that the application of FO System Identification on loudspeaker is one of the most comprehensive method to overcome the difficulties of previous approaches.

5.3 Identification Method

Frequency domain subspace Identification has proven to be an efficient and flexible method for linear system identification (see section 4.2.3). In this section that method is extended with minor modifications to the parametric estimation of FO system. The modifications are:

- the frequencies used have now fractional exponent: $z_k = (j\omega_k)^\alpha$
- a frequency normalization is applied at the beginning and undone at the end
- the matrix whitening prior SVD (step 3c of the algorithm) is no more used

Frequency Normalization The frequency powers that appear in the calculation can yield large numbers and ill-conditioned matrices. Frequency normalization mitigates that issue and improves numerical stability [27]. Frequency scaling is therefore applied at the beginning of the algorithm and undone at the end.

5.4 Experimental Results

The identification algorithm has been applied on the measured impedance curve (fig. 5.8) with respectively: $\alpha = 1, z_k = j\omega_k$ and $\alpha = 1/2, z_k = \sqrt{j\omega_k}$ and for a range of system orders $n_a = 1, 2 \ldots, 30$. The resulting relative rms error $\epsilon[dB] =$

$20log(||G - \tilde{G}||/||G||)$, has been recorded for each and the results are displayed in fig. 5.10. We see that the half differential order yields lower error than the integer order and that the minimum is reached for a lower system order: 8 instead of 30. We observe also that the range of errors is greater for half differentiation order. The modeled impedance curves are displayed for the best case of integer differentiation order and half order along with the the measured impedance in fig. 5.11 and fig. 5.12 respectively. The curves are graphed in magnitude along with the magnitude of the complex error curve $|G - \tilde{G}|$. The overall error obtained in case of integer order is -17dB and -24dB for half-order. It is interesting to see in the case of integer frequency order that the error is mainly concentrated in the low frequencies and that the high frequencies show signs of over-fitting. In he case of half-order the error curve is much flatter and the low frequencies show a better fit. There is little sign of over-fitting, the system order seems appropriate. A fine tuning with a nonlinear optimization procedure like Levenberg-Marquardt would likely reduce the error further.

5.5 Discussion

The subspace algorithm presented could be improved. Output noise reduction could be put in place as described in [25, 27]. The observed output noise covariance matrix is used to reduce the noise influence prior SVD in the step 3c of the algorithm (see [27] for details). That method yields good results for integer differential order but is problematic for fractional order. The noise reduction method needs to adapted in that case.

Fractional differential order opens new degrees of freedom with the choice of differen-

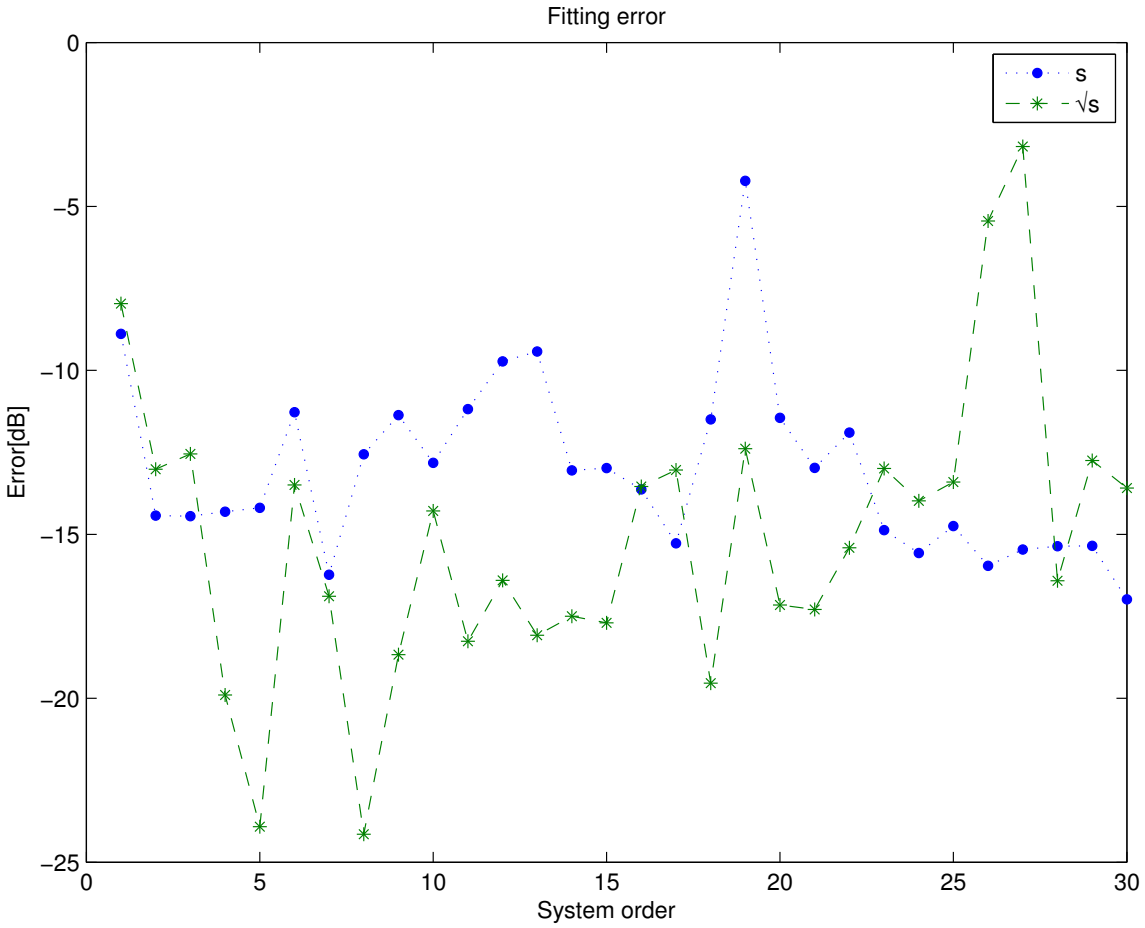

Figure 5.10: **Identification Error vs System Order.**

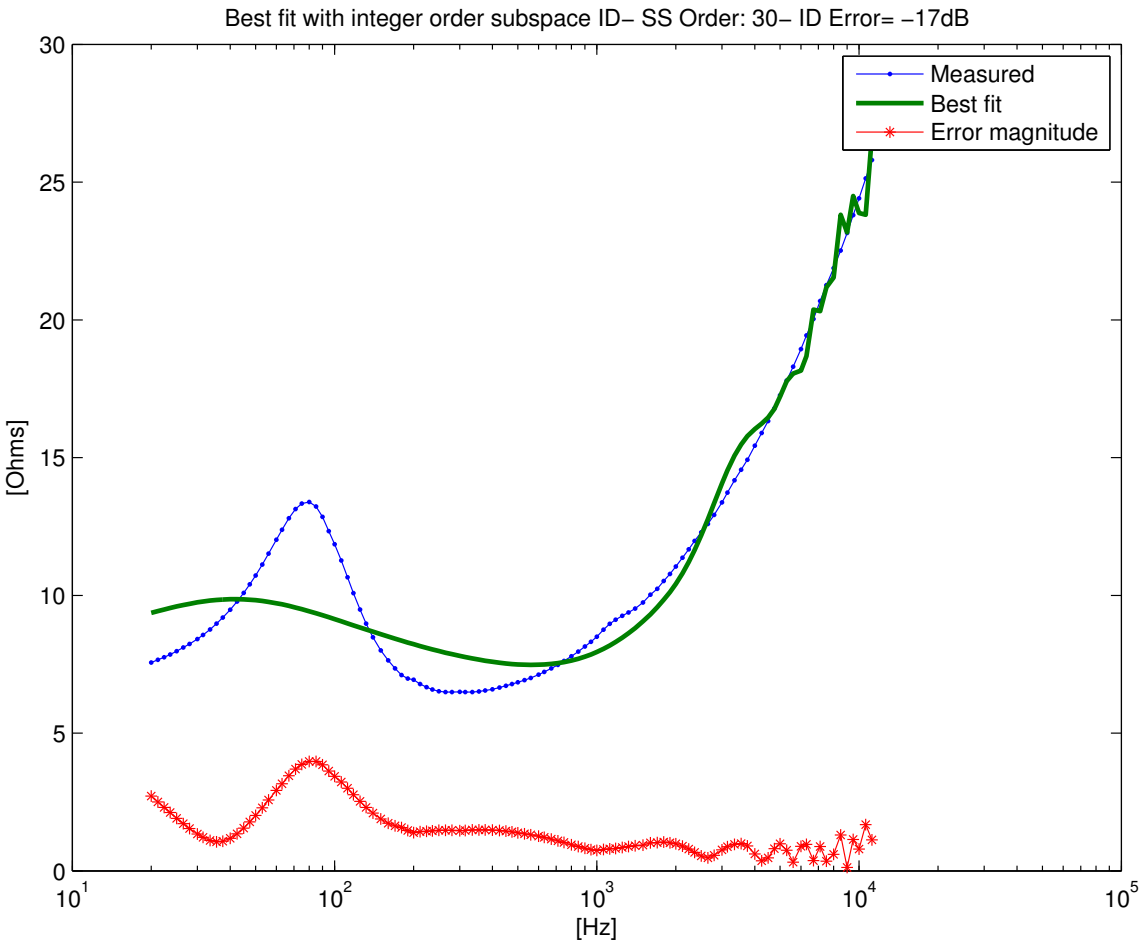

Figure 5.11: **Estimation based on integer order dynamics.**

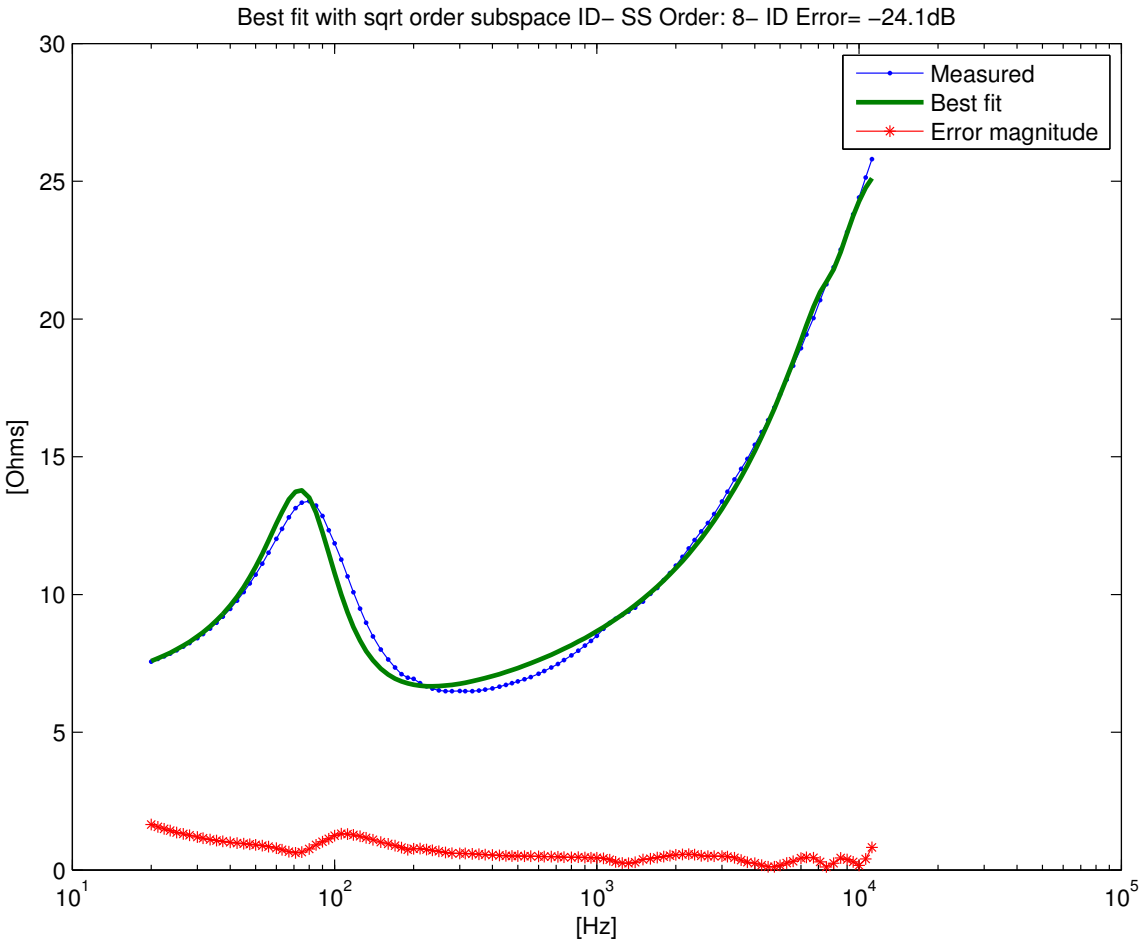

Figure 5.12: **Estimation based on half order ($\sqrt{j\omega}$) dynamics.**

tial order step, commensurate vs. non commensurate orders, continuous vs. discrete order distribution [50]. Recent study [51] shows that different orders and parameters combinations can result in very close transfer functions and the ambiguity grows rapidly with decreasing order values. For example, use of $\alpha = 0.1$ and a system order of 6 result in -30 dB error on the impedance fit. That ambiguity makes the black model approach impossible. On the other hand, the white model approach necessitate a complete understanding of the physical phenomenon and is unrealistic as we have seen. Therefore a grey model approach is the only practical one. The choice of grey model should be guided by the question: what are the useful parameters to identify? In our present case, the identification of the mode (resonance frequency, damping) and the DC resistance have physical pertinence and design consequences. On the other end, fractional order is necessary to model the behavior at frequencies above resonance. That suggest the use of a mixed model combining integer and fractional orders. Finally, the nonlinearities inherent to the operation of loudspeaker are not covered by a fractional order system which is inherently linear. The polynomial approach described in [20] could be used similarly with state-variables of fractional order.

5.6 Conclusion

In this chapter we have presented a novel fractional order approach to loudspeaker identification. First we have shown that the electrical impedance of a loudspeaker exhibits FO behavior. Then a summary of the theory of fractional calculus has been exposed. An illustrative example of a simple FO low-pass filter has been analyzed. A FO model for the loudspeaker has been naturally derived from the classical model. A frequency domain subspace identification method adapted to FO state-space has been described. Experimental results of real data have been presented and have

demonstrated that the FO order approach results in a better modeling of the data with a lower fitting error and a smaller model order than the traditional integer order approach. In a brief discussion we have indicated several directions for further research and improvements. Altogether we consider that the FO approach gives convincing results when applied to loudspeaker modeling and identification.

Chapter 6

Model Reduction and Stabilization

6.1 Model Reduction

6.1.1 Introduction

Noise and nonlinearities introduces superfluous and/or unstable pole/zeros (see section 4.4.2). There are two separate issues: poles/zeros that contribute only into minute details for the FRF, what we can consider being in excess, and unstable poles (and non-minimum phase zeros). Both issues are addressed by editing the pole-zero distribution of the model. In this section we focus on pruning. The mirroring will be described in section 6.2.

6.1.2 Distant Poles/Zeros

As shown below, pole-zero pairs contribute only to small changes in the FRF gain when they are far from the frequency locus in the complex plane ($j\omega$ axis or $e^{j\omega}$ circle). The same can be said of poles and zeros which are close to each other. Removing the excess of poles/zeros results in a model order reduction which is desirable for NL identification (as seen in section 4.4.2). In the following we show how each poles and

zeros affects the FRF. The discussion is done for the discrete case but it can be easily extended to the continuous case. Let's consider a rational transfer function $G(z)$ with K zeros and L poles and express its gain in dB:

$$\begin{aligned} G(z) &= G_0 \frac{\prod_{k=1}^{K}(z - z_k)}{\prod_{l=1}^{L}(z - p_l)} \\ 20\log|G(z)| &= 20\log|G_0| + 20\log\prod_{k=1}^{K}|z - z_k| - 20\log\prod_{l=1}^{L}|z - p_l| \\ &= 20\log|G_0| + 20\sum_{k=1}^{K}\log|z - z_k| - 20\sum_{l=1}^{L}\log|z - p_l| \end{aligned} \tag{6.1}$$

The gain at frequency ω is $20\log|G(e^{j\omega})|$. Each zero z_k brings an maximum attenuation for $\omega = \angle z_k$ equal to $20\log|e^{j\angle z_k} - z_k| = 20\log|1 - |z_k||$ and a minimum attenuation for $\omega = \angle z_k + \pi$ equal to $20\log|-e^{j\angle z_k} - z_k| = 20\log|1 + |z_k||$. For the poles the situation is reversed: each pole p_l brings an maximum amplification for $\omega = \angle p_l$ equal to $-20\log|1 - |p_l||$ and a minimum amplification for $\omega = \angle p_l + \pi$ equal to $-20\log|1 + |p_l||$. The situation is illustrated in the following figure (fig. 6.1).

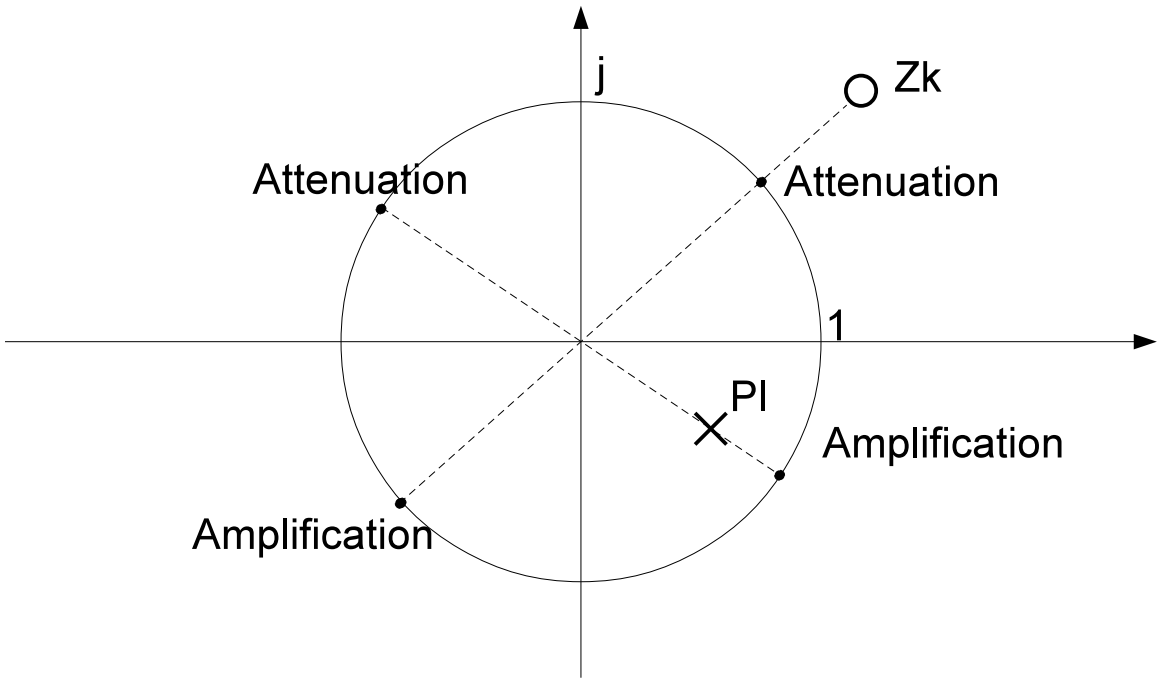

Figure 6.1: **Influence of pole and zero positions on FRF gain.**

The total peak-to-peak ripple effect on the FRF magnitude is then given by:

$$\begin{aligned}\text{Ripple [dB]} &= 20\log\frac{\text{Max. gain}}{\text{Min. gain}} \\ &= 20\log\frac{|1+|z_k||}{|1-|z_k||}, \text{ for a zero} \\ &= 20\log\frac{|1+|p_l||}{|1-|p_l||}, \text{ for a pole}\end{aligned} \tag{6.2}$$

We see in fig. 6.2 that poles and zeros far away from the unit circle have little influence on the FRF magnitude. E.g. a pole (zero) of magnitude $\frac{1}{3}$ or magnitude 3 induces a ripple of 6 dB.

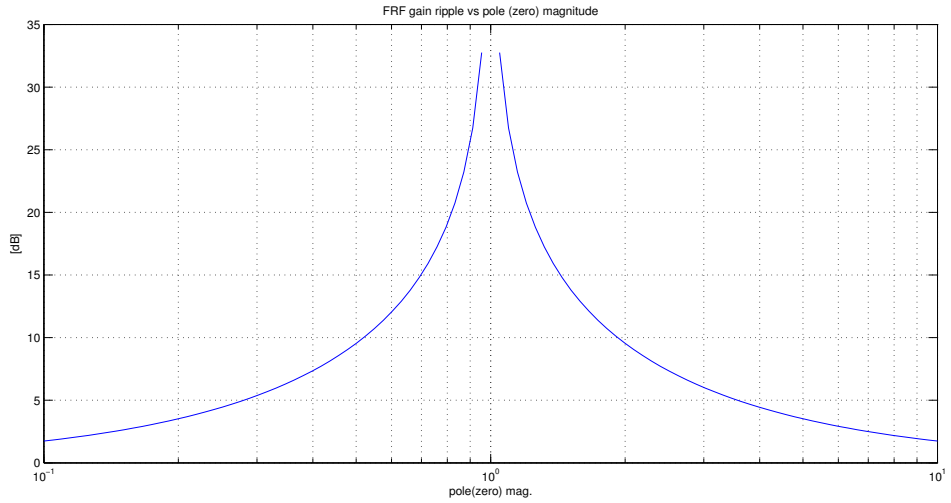

Figure 6.2: **Total ripple effect of a pole (zero) on FRF gain as a function of its distance to the unit circle.**

In the phase domain:

$$\angle G(z) = \angle G_0 + \sum_{k=1}^{K}\angle(z-z_k) - \sum_{l=1}^{L}\angle(z-p_l) \tag{6.3}$$

Each zero adds a contribution to the total phase response and the range of this contribution is relatively invariant with the zero location (see fig. 6.3). Each pole subtracts a contribution in the same way. Suppressing a pole or zero will have about

the same effect on the FRF phase regardless of its distance to the unit circle. If the phase is of importance, caution must be exerted and the overall response must be verified case by case.

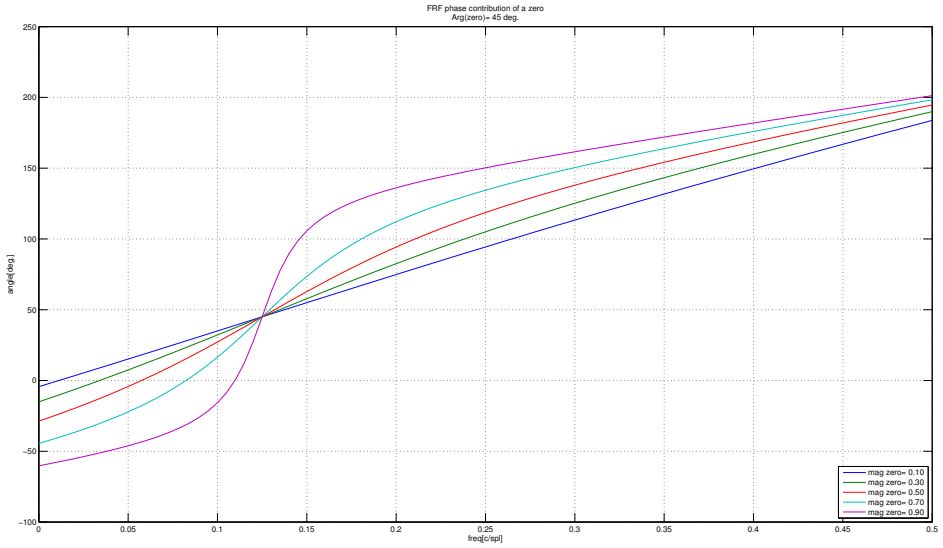

Figure 6.3: **Contribution of a zero of given argument on FRF phase as a function of its distance to the unit circle.**

6.1.3 Poles-Zeros Pairs

The reasoning can be extended to a pair of pole-zero close to each other. From (6.1) we see that the gain contribution of that pair is:

$$\frac{e^{j\omega} - z_k}{e^{j\omega} - p_l} = 1 + \frac{p_l - z_k}{e^{j\omega} - p_l} = 1 + \epsilon(\omega) \tag{6.4}$$

with:

$$\epsilon(\omega) \triangleq \frac{p_l - z_k}{e^{j\omega} - p_l} \tag{6.5}$$

We have

$$|e^{j\omega} - p_l| \geq |1 - |p_l|| \tag{6.6}$$

Using this last equation, we obtain an upper bound for $\epsilon(\omega)$

$$|\epsilon(\omega)| = \left|\frac{p_l - z_k}{e^{j\omega} - p_l}\right| \leq \left|\frac{p_l - z_k}{1 - |p_l|}\right| \triangleq \epsilon_{sup} \tag{6.7}$$

Assuming that $\epsilon_{sup} < 1$ we have

$$1 - \epsilon_{sup} \leq |1 - |\epsilon(\omega)|| \leq |1 + \epsilon(\omega)| \leq |1 + |\epsilon(\omega)|| \leq 1 + \epsilon_{sup} \tag{6.8}$$

That gives us an upper bound of the gain ripple:

$$\text{Ripple [dB]} = 20 \log \frac{1 + \epsilon_{sup}}{1 - \epsilon_{sup}} \tag{6.9}$$

E.g. for a pair of pole-zero with $p_l = 0.8, z_k = 0.8 - 0.01$, we have $\epsilon_{sup} = \frac{0.01}{1-0.8} = 0.05$ and a maximum ripple of about 0.87 dB. This situation is illustrated by fig. 6.4.

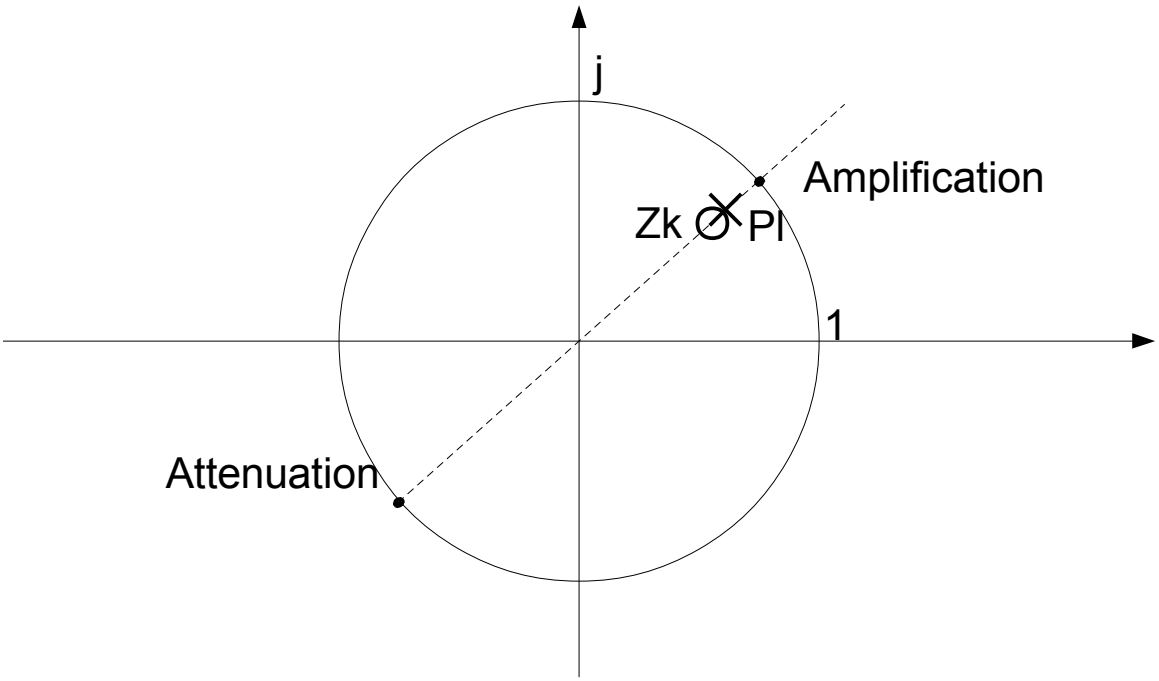

Figure 6.4: **Effect of a pole-zero pair on FRF gain.**

The influence of the phase can be be estimated too. A quick geometric reasoning let us to:

$$\angle(1 + \epsilon(\omega)) \leq \tan^{-1}(\epsilon_{sup}) \approx \epsilon_{sup} \text{ for } |\epsilon_{sup}| \ll 1 \tag{6.10}$$

Thus we conclude that we can use the criteria $\left|\frac{p_l - z_k}{1-|p_l|}\right| \ll 1$ to eliminate some pole-zero pairs.

6.1.4 Balanced Model Reduction

An alternative and simple model reduction method base on balanced realization will be introduced in the following paragraphs.

A. Balanced Realization

Suppose that a minimal and stable state-space representation of a discrete time system is given by $\{A, B, C, D\}$. Let the controllability and observability Gramians associated with this system be written as

$$W_c = \sum_{k=0}^{\infty} A^k B B^T (A^T)^k \tag{6.11}$$

$$W_o = \sum_{k=0}^{\infty} (A^T)^k C^T C A^k \tag{6.12}$$

It is not difficult to show that W_c and W_o satisfy the discrete Lyapunov equations

$$A W_c A^T - W_c = -B B^T \tag{6.13}$$

$$A^T W_o A - W_o = -C^T C \tag{6.14}$$

A minimal state-space representation is called balanced if the controllability and observability Gramians are equal and diagonal i.e. if

$$W_c = W_o = diag\{\sigma_1, \sigma_2, \ldots \sigma_n\}, \tag{6.15}$$

with $\sigma_i > \sigma_{i+1}, i \in [1, n-1[$. Let T be a transformation matrix (yet to be determined) such that $\{\hat{A} = T^{-1}AT, \hat{B} = T^{-1}B, \hat{C} = CT, \hat{D} = D\}$ is internally balanced. The following algorithm adopted from [52] is a possible way to obtain the required transformation.

Algorithm

1. Compute the Cholesky factors of the Gramians. Let L_c and L_o denote the lower triangular Cholesky factors of W_c and W_o that is,

$$W_c = L_c L_c^T$$
$$W_o = \Sigma L_o L_o^T \tag{6.16}$$

2. Compute the singular value decomposition of the product of the Cholesky factors; that is,

$$L_o^T L_c = V \Sigma U^T \tag{6.17}$$

3. Form the balancing transformation

$$T = L_c U \Sigma^{-1/2} \tag{6.18}$$

It is noted that

$$T^{-1} = \Sigma^{-1/2} V^T L_o^T \tag{6.19}$$

4. Form the balanced state-space matrices

$$\hat{A} = T^{-1} A T = \Sigma^{-1/2} V^T L_o^T A L_c U \Sigma^{-1/2}$$
$$\hat{B} = T^{-1} B = \Sigma^{-1/2} V^T L_o^T B \tag{6.20}$$
$$\hat{C} = C T = C L_c U \Sigma^{-1/2} \tag{6.21}$$

B. Model Reduction using Balanced Realization

There are three methods for model reduction based on balanced realization applicable for continuous an discrete systems.

Method 1: (Direct Truncation)

Consider the internally balanced realization of a system partitioned based on large and small singular values

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u$$
$$y(k) = \begin{pmatrix} C_1 & C_2 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} \tag{6.22}$$

Remove from the system matrices the blocks corresponding to the smaller singular values and specify the reduced order model by the triple $\{A_{11}, B_1, C_1\}$. Note that, if direct term from a proper realization exists, then it remains in the reduced-order model.

Method 2: (Singular Perturbation Balanced Truncation)

Partition the system as in Method 1 and use the following reduced-order model

$$\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}u$$
$$y(k) = \bar{C}\bar{x}(k) \tag{6.23}$$

where $\bar{A} = A_{11} - A_{12}A_{22}^{-1}A_{21}$, $\bar{B} = B_1 - A_{12}A_{22}^{-1}A_{21}$ and $\bar{C} = C_1 - C_2A_{22}^{-1}A_{21}$. The reduced-order system $\{\bar{A}, \bar{B}, \bar{C}\}$ is also internally balanced.

6.2 Model Stabilization

6.2.1 Introduction

Frequency domain identification algorithms do not guarantee a stable model. Even if the measured system is stable, which is the case with loudspeakers, output noise and nonlinearities *can lead the identification process to output an unstable model.* The issue is well known [38]. Model stability is desirable in general and is necessary in our case because our final nonlinear identification is done *by minimization of the output error in the time domain* .

6.2.2 Short Review of Existing Approaches

To stabilize the model obtained with subspace identification they are different approaches:

- impose stability constraints during the identification process. In [25], the unstable eigenvalues of the discrete model are projected inside the unit circle. Other methods involves a iterative constrained optimization [53].
- add a delay to the measured FRF [39].
- prune the unstable poles or mirror them [40].

6.2.3 Mirroring

Poles pruning was described in section 6.1. This section describes a classical technique used to fix unstable poles in a transfer function. The discussion is done for the discrete case but it can be easily extended to the continuous case. The principle is to replace each outstanding pole by its inverse conjugate which has an inverse magnitude but the same angle:

$$p_0 = |p_0|e^{j\omega_0} \Leftrightarrow \frac{1}{p_0^*} = \frac{1}{|p_0|}e^{j\omega_0} \tag{6.24}$$

The unstable pole is mirrored with respect to the unit circle (see fig. 6.5).

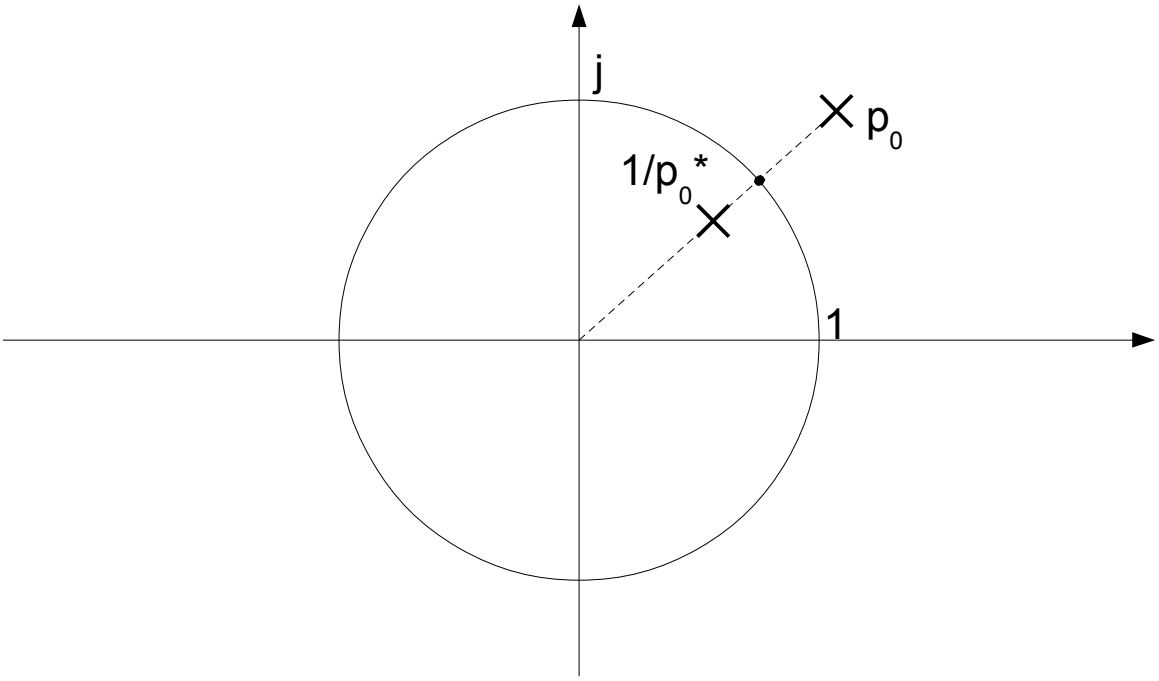

Figure 6.5: **Pole mirroring.**

That transformation des not change the magnitude of the FRF (up to a constant gain). Lets consider for example, the unstable filter:

$$H(z) = \frac{1}{1 - p_0 z^{-1}}, \text{ with } |p_0| > 1 \tag{6.25}$$

After mirroring the unstable pole, we obtain a stable FRF:

$$\tilde{H}(z) = \frac{1}{1 - p_0^{*-1} z^{-1}} \tag{6.26}$$

On the unit circle, the gain of the stabilized FRF is proportional to the gain of the original one:

$$|\tilde{H}(e^{j\omega})| = \left|\frac{1}{1 - p_0^{*-1}e^{-j\omega}}\right| = \left|\frac{p_0^* e^{j\omega}}{p_0^* e^{j\omega} - 1}\right| = \left|\frac{p_0}{p_0 e^{-j\omega} - 1}\right| = |p_0||H(e^{j\omega})| \tag{6.27}$$

It is worth noting that pole mirroring affects the phase response. The relationship between original and stabilized FRF is:

$$\tilde{H}(e^{j\omega}) = H(e^{j\omega})\frac{1 - p_0 e^{-j\omega}}{1 - p_0^{*-1} e^{-j\omega}} \tag{6.28}$$

and the second term is an all-pass filter that introduces a phase distortion.

6.2.4 Delay

In this section we will show how adding a delay to the target FRF can be used to stabilize the identified model. First it is worth mentioning that, due to instrumentation and measurement limitations, the delay between the voltage and current measured may be off by a few samples. That introduces a slope error on the phase. In high frequencies the impedance should behave as an inductance with constant phase. The measured impedance shows a linear phase due to a added delay (see fig. 6.6). A least-square estimation of the phase slope gives a delay error of about 0.6427 sample.

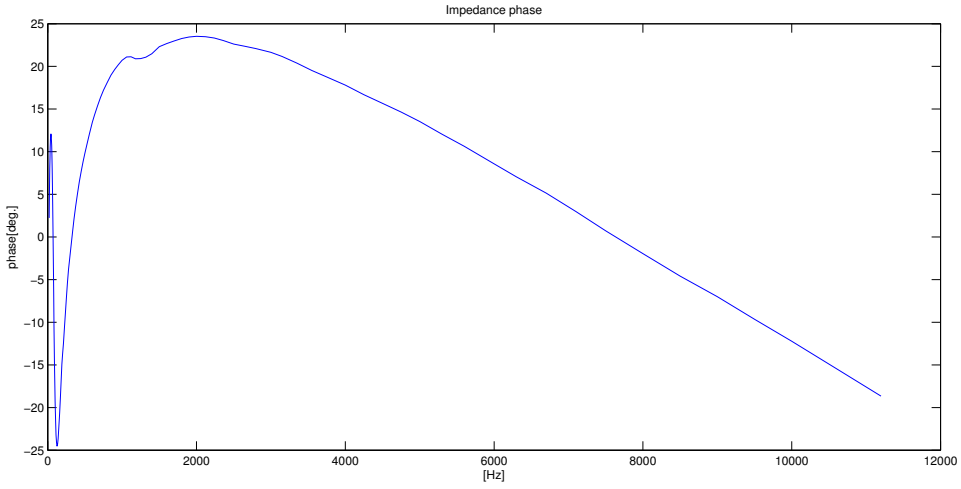

Figure 6.6: **Phase of measured impedance.**

The subspace identification performed on this data results in a unstable model as shown by the pole-zero map (see fig. 6.7).

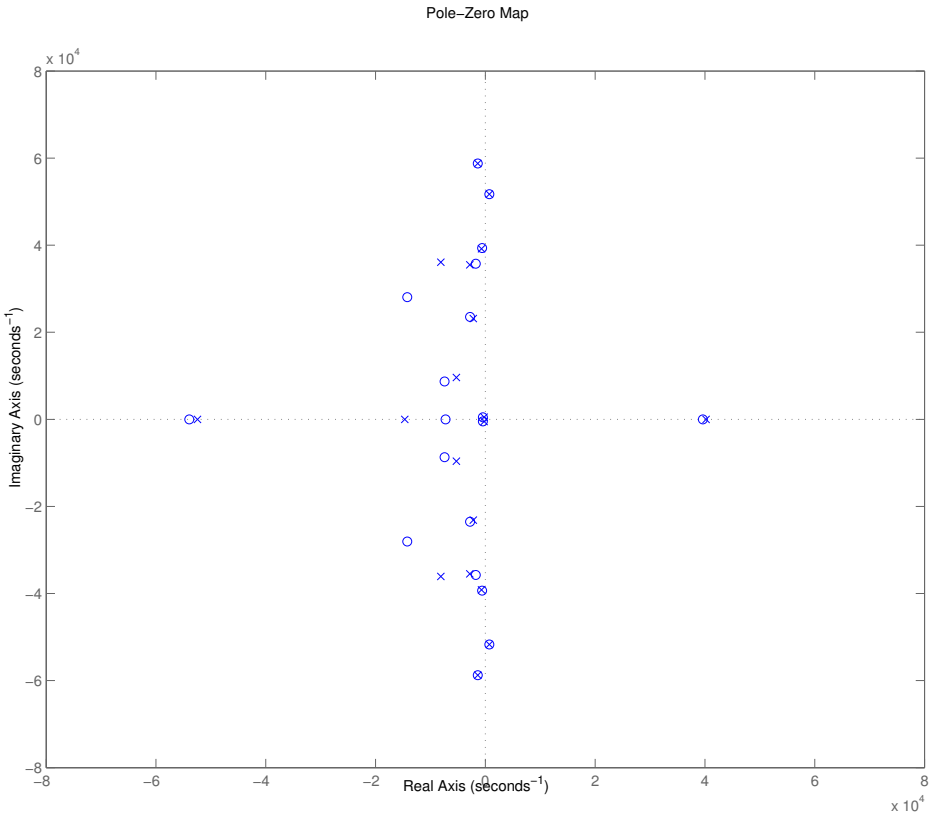

Figure 6.7: **Pole-Zero map of identified model for measured impedance with uncompensated delay. ×: poles, ○: zeros. Integer order dynamics. Best fit.**

When the delay is compensated by $G_d(f) = G(f)\exp(j2\pi f\tau)$, where G_d, G, f, τ are respectively the delayed FRF, the measured FRF, the frequency values, the correction delay, and the identification is performed on the delayed FRF, the identified model is then stable as shown by the pole-zero map (see fig. 6.8).

This demonstrates the important impact that the delay has on the stability of the

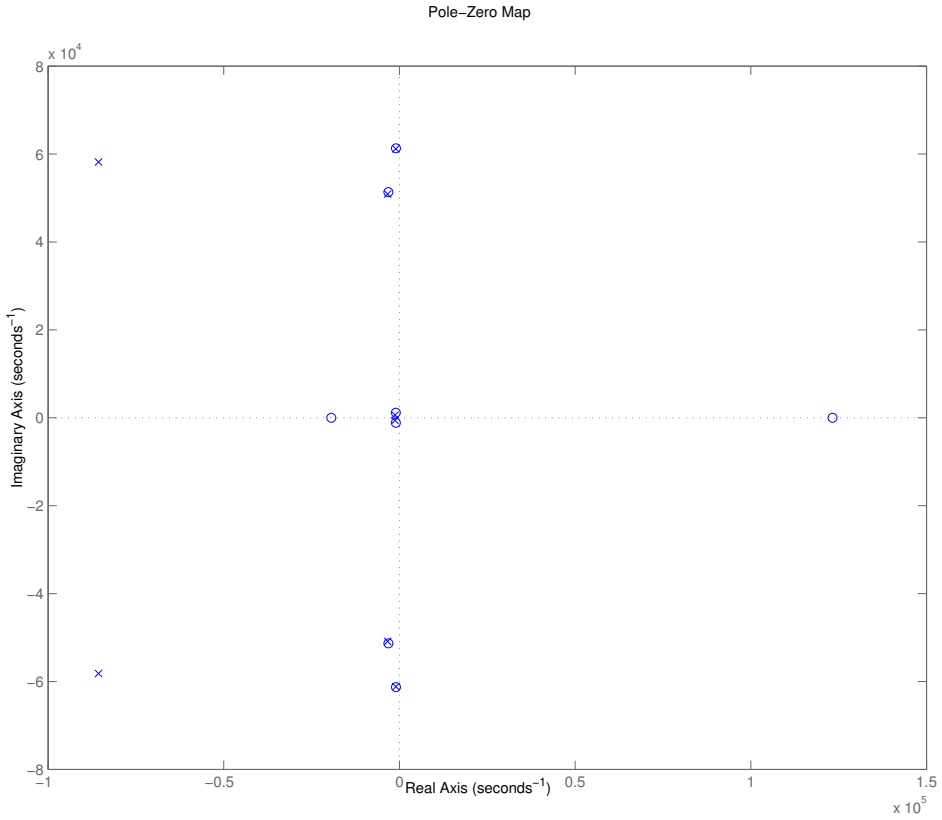

Figure 6.8: **Pole-Zero map of identified model for measured impedance with compensated delay. ×: poles, ○: zeros. Integer order dynamics. Best fit.**

identified model. In fact, varying incrementally the correction delay results in vastly different results in terms of stability, model order and goodness of fit. This have been observed in [39]. In that paper a two tier optimization method is proposed. First, for a given model, the algorithm finds the delay value that achieves the best fitting error while preserving stability. Second, for a given delay value, the parameters of the model

are optimized to minimize the fitting error. That approach provides an automatic delay selection, but involves solving a set of ODE for the parameters optimization and a gradient search for the delay. That rather involved method has not been developed in this study.

6.2.5 Additive Decomposition

As pointed out before, the resulting subspace identification method leads to an unstable system in spite of the fact that the original system is stable. Model reduction for unstable systems can be performed in several ways as are described in the above subsections. In general the idea is based on the fact that unstable poles are usually important for the dynamics of the system, hence they should be preserved. This can be achieved via an *additive decomposition* of the transfer function as

$$G(s) = G_-(s) + G_+(s) \tag{6.29}$$

with $G_-(s)$ stable, $G_+(s)$ unstable, applying balanced truncation to G_- to obtain $\hat{G}_-$ and setting

$$\hat{G}(s) = \hat{G}_-(s) + G_+(s) \tag{6.30}$$

thereby preserving the unstable part of the system. Such a procedure can be implemented using the spectral projection methods for block-diagonalization which leads to

$$\begin{aligned} \tilde{A} &\triangleq U^{-1}AU = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix} \\ \tilde{B} &\triangleq U^{-1}B \triangleq \begin{pmatrix} B_1 \\ B_2 \end{pmatrix},\ \tilde{C} \triangleq CU \triangleq \begin{pmatrix} C_1 & C_2 \end{pmatrix},\ \tilde{D} \triangleq D \end{aligned} \tag{6.31}$$

This yields the desired additive decomposition as follows:

$$\begin{aligned} G(s) &= C(sI-A)^{-1}B + D = \tilde{C}(sI-\tilde{A})^{-1}\tilde{B} + \tilde{D} \\ &= \begin{pmatrix} C_1 & C_2 \end{pmatrix} \begin{pmatrix} (sI_k - A_{11})^{-1} & \\ & (sI_{n-k} - A_{22})^{-1} \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} + D \\ &= \{C_1(sI_k - A_{11})^{-1}B_1 + D\} + \{C_2(sI_{n-k} - A_{22})^{-1}B_2\} \\ &\triangleq G_-(s) + G_+(s) \end{aligned} \tag{6.32}$$

Then apply the procedure of Section 6.1.4 to G_- and obtain the reduced order model by adding the transfer functions of the stable reduced and the unstable unreduced parts as summarized above. However, in our case, it is the opposite; namely, we discard the unstable part of the resulting decomposition $G_+(s)$ and apply balanced model reduction on $G(s)$ if needed.

6.2.6 Specific Issues of Fractional Systems

The transfer function is influenced by the distance of poles and zeros to the complex frequencies axis. With continuous fractional systems, the complex locus of frequencies lies on $(j\omega)^\alpha$ instead of the usual $j\omega$ (see fig. 6.9). That has to be taken in account in pruning and mirroring actions.

$$G(s) = \frac{\prod_{k=1}^{K} s^\alpha - z_k}{\prod_{l=1}^{L} s^\alpha - p_l} \tag{6.33}$$

Solutions of $s^\alpha - c = 0$.

Let's have:

$$c \triangleq re^{j\theta}, \text{with: } r > 0, -\pi < \theta \leq \pi \tag{6.34}$$

Solutions are the set:

$$s = \{r^{1/\alpha} e^{j\frac{\theta+2k\pi}{\alpha}} : \text{k integer}\} \tag{6.35}$$

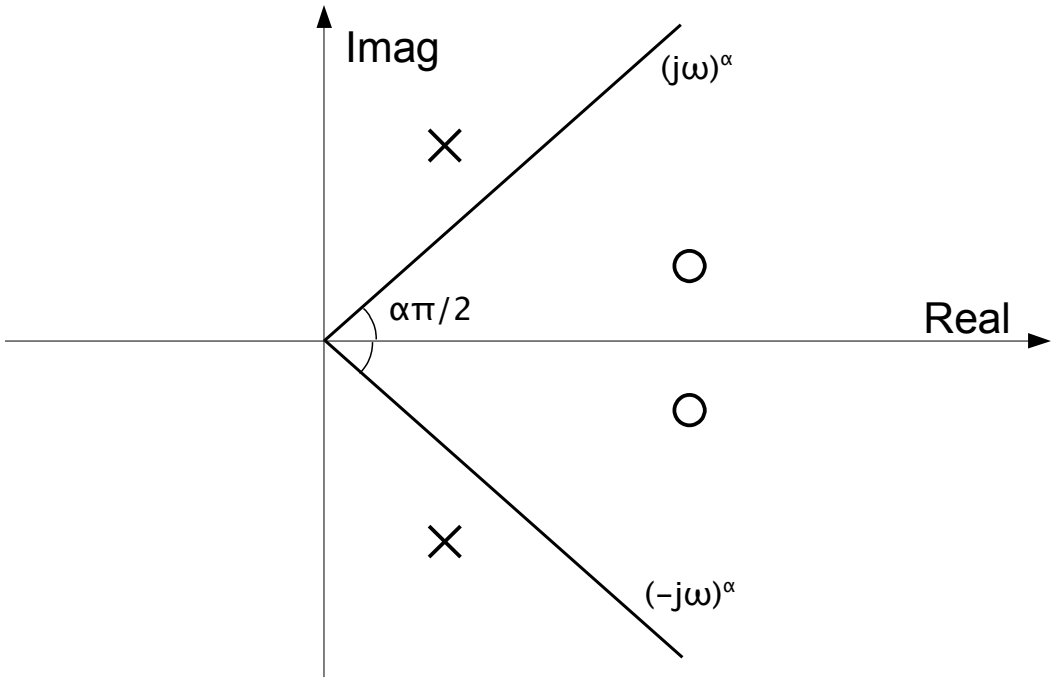

Figure 6.9: **Frequency axis for a fractional order system of degree** α

1. $\alpha = p$ integer. The set of solutions contains p distinct values evenly distributed on circle of radius $r^{1/p}$ (which is a well-known result):

$$s = \{r^{1/p} e^{j\frac{\theta+2k\pi}{p}} : \text{k= 0,...p-1}\} \tag{6.36}$$

2. $\alpha = 1/p$, p integer. The solution is unique:

$$s = \{r^p e^{j(\theta+2k\pi)p} : \text{k integer}\} = r^p e^{j\theta p} = c^p \tag{6.37}$$

3. $\alpha = p/q$, p,q non commensurate integers. The set of solutions contains p distinct values evenly distributed on circle of radius $r^{p/q}$:

$$s = \{r^{q/p} e^{j(\theta+2k\pi)q/p} : \text{k= 0,...p-1}\} \tag{6.38}$$

4. α irrational. The set of solutions contains an infinite (countable) number of values evenly distributed on circle of radius $r^{1/\alpha}$.

As an illustrative example, the set of solutions of $s^{5/3} - 2 = 0$ is:

$$s = \{2^{3/5} e^{j2k\pi\, 3/5} : \text{k= } 0,\ldots 4\}$$

$$= \{1.5157, -1.2262 - 0.8909i, 0.4684 + 1.4415i, 0.4684 - 1.4415i, -1.2262 + 0.8909i\}$$

Chapter 7

Conclusions

The nonlinearities inherent to the operation of loudspeaker are not covered by a fractional order system which is inherently linear. For that purpose, the polynomial approach can be extended with state-variables of fractional order. In the following section, we combine the advantages of both approaches, namely polynomial state space and fractional order, to obtain a comprehensive nonlinear modeling of the loudspeaker.

7.1 Combination of Polynomial and FO Approaches

By injecting FO derivatives of state-variables into the PNLSS model we obtain the general FO nonlinear model

$$D_t^\alpha \zeta = A\zeta + Bu + Ep(\zeta^T, u) \tag{7.1}$$

$$y(t) = C\zeta + Du + Fq(\zeta^T, u) \tag{7.2}$$

where

- ζ is the state-vector
- p and q are polynomial vectors of combinations of state and input variables

- $D_t^{\alpha}\zeta = (D_t^{\alpha_1}\zeta_1, \ldots, D_t^{\alpha_k}\zeta_k)^T$, vector of fractional derivatives

For FO derivative, we can use for example the Caputo's definition:

$$D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha-n+1}} \, \mathrm{d}\tau \tag{7.3}$$

with $n-1 < \alpha \leq n, n \in \mathbb{Z}$

This general model can readily be applied to loudspeaker by using fractional inductance model (see ch. 5)

$$\begin{aligned} D_t^2 x(t) &= -\frac{k}{m}x - \frac{r}{m}D_t^1 x(t) + \frac{Bl}{m}i(t) \\ D_t^{\alpha} i(t) &= -\frac{Bl}{L}D_t^1 x(t) - \frac{R}{L}i(t) + \frac{1}{L}u \end{aligned}$$

combined with the PNLSS loudspeaker model described in ch. 4, as to obtain

$$(D_t^1\zeta_1, D_t^1\zeta_2, D_t^{\alpha}\zeta_3)^T = A\zeta + Bu + Ep(\zeta^T, u)$$

$$y(t) = C\zeta$$

with

$$\begin{aligned} \zeta &= (x, D_t^1 x, i)^T, \text{ state vector} \\ A &= \begin{pmatrix} 0 & 1 & 0 \\ -\frac{k}{m} & -\frac{r}{m} & \frac{Bl}{m} \\ 0 & -\frac{Bl}{L} & -\frac{R}{L} \end{pmatrix} \\ B &= \begin{pmatrix} 0 & 0 & \frac{1}{L} \end{pmatrix}^T \\ C &= \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}, \text{ for current output} \end{aligned}$$

and where E coefficients are obtained by Taylor expansion of the electromechanical parameters in regard to the cone excursion x and polynomial vector $p(\zeta^T, u)$ contains monomials in ζ and u.

7.2 Full Identification Procedure

We have seen in chapter 4 that, because of the additive nature of nonlinearities in the polynomial state-space model, the overall identification procedure can proceed in two successive parts: first the linear part then the nonlinear part. Identification procedure summary of system described by :

1. measurement of the FRF which gives the BLA of the system (7.1):
2. subspace identification in the frequency domain that yields estimates of linear parameters A, B, C, D
3. least-square identification in the time domain to obtain estimates of nonlinear parameters E, F

Frequency domain ID for FO systems is described in chapter 5. LS identification in the time domain (as described in ch. 4) necessitates a discrete time implementation of the FO nonlinear state-space system. Unfortunately it is not a simple task as explained in [35, 54] and we didn't explore that avenue in this study.

7.3 Discussion and Conclusion

The goal of this thesis was to find a comprehensive nonlinear modeling of loudspeakers, from voltage input to acoustic output.

In chapter 4, we have presented a technique based on polynomial state-space for nonlinear modeling of loudspeaker which can effectively be used in identification process. It is an overall approach that encompasses the whole system in one model which explains both linear and linear characteristics. The linear part is described by a classical state-space system and the non-linear part is described by polynomial combination of state and input variables. The nonlinear part is additive to the linear part which

is convenient for identification and the precision of the model is scalable by the mean of the polynomial order. PNLSS is a powerful and general tool to model NL systems, including systems with nonlinear feedback and is well adapted to loudspeaker description.

In chapter 5 we have found that fractional differential equations greatly improves the linear modeling of the loudspeaker. This new approach results in a compact and efficient fractional state-space model that fits experimental data with better accuracy and lower system order than traditional approach. The same linear identification algorithm used in chapter 4 can be re-used with little modification. FO approach is novel in the domain of loudspeaker modeling. It provides a clear and powerful mathematical theory that explains experimental data.

Finally we have shown that the two approaches can be easily combined into a model that encompasses the nonlinearities and the FO derivatives into a fractional order polynomial state-space system that potentially better describe the behavior of loudspeaker. Complete identification needs time domain modeling of nonlinear FO SS system. That should be the subject of future study.

However, we have seen that polynomial nonlinear state-space require large number of parameters, and FO differential equations are difficult to realize in the time domain. Research is ongoing to develop more parsimonious representation of nonlinear systems like NL block models [55], port-hamiltonian systems [56], and to pursue time-domain realizations of FO systems [42].

Altogether, the results obtained with this study were promising and show that our approach is worth of further investigation. Overall, the contributions of this thesis have been a generic and comprehensive nonlinear model of loudspeaker, the application of FO calculus to better explain its linear characteristics and the elaboration of associated identification methods. Finally, ecause of its polynomial nature, the pro-

posed model opens the possibility of inversion and therefore feedforward linearization ('anti-distortion') of great interest for the audio industry.

Appendix A

Codes for Best Linear Approximation

These codes and the ones given in the subsequent appendices were written in Matlab R2012a.

```
function [Gbla, VarN, Varbla]= BLA(u, y, M, P, N, w)
% Estimate the BLA of the FRF by averaging on M realizations
% of P+1 periods of random multisine u and response y
% {in}
        % u= input signal (sequencial random multisine) of length N(P+1)
   M
        % y= DUT response
        % M= # of random stimulus realizations
        % P= # of periods -1
        % N= period length;
        % w= multisine frequencies in rad/spl in ]0, pi]
% {out}
        % Gbla= complex FRF one-sided, estimated at freq w
        % VarN= Variance due to noise
```

```
        % Varbla= Variance of Gbla (noise + stochastic nonlinearities)

k= round(w/2/pi*N)+1; % frequency indices
K= length(k);
Grz= zeros(K,M); % M estimates of G
Var_nG= zeros(K,M);
realiz_u= reshape(u, (P+1)*N,M); % M realizations of P+1 periods
realiz_y= reshape(y, (P+1)*N,M); % M realizations of P+1 periods

for m= 1:M
        frames_u= reshape(realiz_u(N+1:end,m),N,P); % pick realiz m and
    throw away first period
        U= fft(frames_u);
        avgU= mean(U,2);
        % varU= var(U,0,2); % Variance is zero
        frames_y= reshape(realiz_y(N+1:end,m),N,P); % pick realiz m and
    throw away first period
        Y= fft(frames_y);
        avgY= mean(Y,2);
        varY= var(Y,0,2);
        Grz(:,m)= avgY(k)./avgU(k);
        Var_nG(:,m)= 1/P*varY(k)./(abs(avgU(k)).^2); %noise variance on
    G
end;
Gbla= mean(Grz,2);
VarN= 1/M*mean(Var_nG,2); % Variance due to noise
Varbla= 1/M*var(Grz,0,2); % overall variance of Gbla
% VarNL= abs(Varbla- VarN); % Variance due to stochastic nonlinearities-
     can be locally negative!
```

```
function [u, w]= Excitation_signal(M,P,N,Level,option,shape)
% Creates a train of M realizations of P+1 periods of random multisine
% {in}
        % M= # of random realizations
        % P= # of periods -1
        % N= period length;
        % Level= rms level [lin]
        % option= harmonic grid (see multisine)
        % shape= spectrum shape (see multisine)
% {out}
        % u= excitation signal of length N*(P+1)*M
        % w= multisine frequencies in rad/spl in ]0, pi[
u= zeros(N*(P+1)*M,1);

if (nargin<6)
        shape= 'white';
end;

for m=1:M
    [v, w] =  multisine(N, option,shape);
    v= v*Level;
    for p= 0:P
        n=((m-1)*(P+1)+p)*N;
        u(n+1:n+N)= v;
    end;
end;
```

```
function [u, w] =  multisine(T, flavor, shape)
% Calculates multisine of length T with different flavors and
% spectrum shape
% E(u)= 0; E(u^2)= 1;
% {input}
% T= multisine length in samples;
% flavor     = 'full' for full multisine with all freq from 2pi/T to pi
%              = 'odd' for odd harmonics of 2pi/T up to pi
%              = 'oddodd' for harmonics of rank 4k+1
%              = 'R10', 'R20', 'R40', 'R80' for 10, ...80 harmonics per
    dec.
%              rounded to the nearest line
% shape    = 'white' for flat spectrum (default)
%              = 'pink' for pink spectrum (-10dB/dec)
%              = 'red' for red spectrum (-20dB/dec)
% {output}
% u= multisine signal
% w= multisine freq in rad/spl in ]0..pi]


flavor= lower(flavor);
if flavor(1)=='r' % log scale
        switch flavor
                case 'r10'
                        M= 10;
                case 'r20'
                        M= 20;
                case 'r40'
                        M= 40;
                case 'r80'
                        M= 80;
        end;
```

```
        f= loglines(floor(T/2), M);
else
        switch flavor
                case 'full',
                        r=1;
                case 'odd'
                        r=2;
                case 'oddodd'
                        r=4;
                otherwise
                        r=1;
        end;
        f= (1: r: fix(T/2))';
end;

F= length(f);

if (nargin==3)
        switch lower(shape)
                case 'white'
                        A= ones(F,1);
                case 'pink'
                        A= 1./sqrt(f);
                case 'red'
                        A= 1./f;
                otherwise
                        A= ones(F,1);
        end;
else % white (default)
        A= ones(F,1);
end;
```

```
U= zeros(T,1);
ph= 2*pi*rand(F,1);
for k= 1:F
        U(f(k)+1)= A(k)*exp(1j*ph(k));
end;


u= real(ifft(U));
u= u/std(u); % normalization
w= f*2*pi/T;
```

```
function delay = IOdelay(u, y, M)
% Estimate the delay between input signal u and output signal y
%{in}
%       M= size of FFT used to estimate the Impulse response from x to y
% {out}
%        delay= delay from u to y in samples
%
L= length(u);
window= ones(M,1);% rectangular window
overlap= 0;
sizefft=M;
fs= 1;
Syu= cpsd(y(1:L),u,window,overlap,sizefft,fs,'twosided');
Suu= pwelch(u,window,overlap,sizefft,fs,'twosided');
e= max(Suu)/1E3; % As big as possible
H= Syu./(Suu+e);
H(fix(M/2)+2:end)= 0; % eliminate negative freq to get analytic IR
h= ifft(H); % Impulse Response
% h= ifft(H./(abs(H)+eps)); % phat transform. Alternative.
[peak, delay]= max(abs(h)); % delay estimation
delay= delay-1;
```

Appendix B

Codes for Frequency Domain Subspace Identification

```
function [A_, B_, C_, D_]= mckelvey_algo1(G, n)
% Estimate linear SS using mckelvey_algo1
% G= FRF full circle (column)
% n= expected system order
% A_, B_, C_, D_= estimated state-space matrices
h= real(ifft(G)); %impulse response
r= 2*n; q= 2*n;
H= hankel(h(2:q+1), h(q+1:q+r));% Ignore h(0)
[U,S,V]= svd(H);
Us= U(:,1:n); %n first left singular vectors
% A_, C_ estimation
J1= [eye(q-1), zeros(q-1,1)];
J2= [zeros(q-1,1), eye(q-1)];
J3= [1, zeros(1, q-1)];
%A_= pinv(J1*Us)*(J2*Us);
A_= (J1*Us)\(J2*Us);
C_= J3*Us;
% B_, D_ estimation
```

```
M= length(G)/2;
I= eye(n); X= zeros(M+1,n+1);
for k= 0:M
    X(k+1,:)= [C_/(exp(1j*pi*k/M)*I-A_), 1];
end;
% Xre= [real(X); imag(X)];
% Gre= [real(G(1:M+1)); imag(G(1:M+1))];
% bd= Xre\Gre;
bd= real(X'*X)\real(X'*G(1:M+1));% cf Pintelon 2002 Appendix B.3
B_= bd(1:n); D_= bd(n+1);
end
```

```
function [SG, Cost]= mckelvey_algo2(G, sigmaG, f, na, r, fs, option)
% Estimate linear SS using mckelvey_algo2
%
%{in}
        % G: FRF
        % sigmaG: std-dev on G
        % f: G freq values in [Hz]
        % na: model order
        % r: estimation dimension > 1.5 na
        % fs: sampling rate [Hz]
        % option:  'w'(default) for weighted estimation,
        %              'unw' for unweighted
        %
%
% {out}
        % SG: estimated SS system [A,B,C,D]
        % Cost: weighted estimation cost
%
%% a) Matrices construction
f= f/fs; % convert to [cycle/Spl]
F= length(f);
Wtilde=zeros(r,F);
Gtilde= zeros(r,F);
Cs= zeros(r,F);
for l= 1:r
    for k=1:F
        Wtilde(l,k)= exp(2j*pi*f(k)*(l-1));
        Gtilde(l,k)= Wtilde(l,k)*G(k);
        Cs(l,k)= Wtilde(l,k)*sigmaG(k);
    end;
end;
```

```
Wre= [real(Wtilde), imag(Wtilde)];

Gre= [real(Gtilde), imag(Gtilde)];
Z= [Wre; Gre];
CG= real(Cs*Cs');

%% b) Elimination of input term by QR factorization
[Q,R]= qr(Z',0);
RT= R';
R22T= RT(r+1:end,r+1:end);

%% c) Noise reduction and SVD
C2= sqrtm(CG);
% C2= eye(r);
[L,S,V]= svd(C2\R22T);
Or= C2*L(:,1:na);% Estimation of extended observability matrix

%% d) Estimation of A and C
A= Or(1:r-1,:)\Or(2:r,:);
C= Or(1,:);

%% e) Estimation of B and D

% Frequency variables
z= exp(2j*pi*f);
H= zeros(F,na+1);
Ina= eye(na);
for k=1:F
    H(k,:)= [C/(z(k)*Ina - A),1];
end;

% Weight matrix
```

```
if nargin== 7 && strcmpi(option, 'unw')
        W= eye(F);
else
        W= diag(sigmaG.^(-2));
end;

% Input matrices estimation
BD= pinv(real(H'*W*H))*real(H'*W*G); % cf Pintelon 2002 Appendix B.3
% pinv to avoid conditionning issue
B= BD(1:na);
D= BD(na+1);

%% Output
SG= ss(A,B,C,D,1/fs); % discrete model with sampling rate fs
Cost=norm(W*G-W*H*BD)/sqrt(length(G)); % weighted rms error
```

```
function [SG, Cost, G_]= SubSpace_CT_simple(G, stdG, xi, na)
% Estimate linear SS using Subspace algo for continuous time system
% without data orthogonalization nor noise reduction.
% Pintelon & Schoukens- System ID freq domain- Wiley 2012
%
% {in}
%         G: FRF (vector)
%         stdG: std-dev on G (vector)
%         xi: complex freq values in [rad/s]
%         na: model order
% {out}
%         SG: estimated SS system [A,B,C,D]
%         Cost: weighted estimation cost [dB]
%         G_: estimated output spectrum
%

r= 2*na; % estimation dimension > 1.5 na

%% Frequency scaling
F= length(xi);
wscale= max(abs(xi))/2;
xi= xi/wscale; % Frequency scaling for numerical stability
%% a) Matrices construction
Wtilde=zeros(r,F);
Gtilde= zeros(r,F);
Cs= zeros(r,F);
for l= 1:r
    for k=1:F
        Wtilde(l,k)= xi(k)^(l-1);
        Gtilde(l,k)= Wtilde(l,k)*G(k);
        Cs(l,k)= Wtilde(l,k)*stdG(k);
```

```
    end;
end;
Wre= [real(Wtilde), imag(Wtilde)];
Gre= [real(Gtilde), imag(Gtilde)];
Z= [Wre; Gre];
% CG= real(Cs*Cs');

%% b) Elimination of input term by QR factorization
[~,R]= qr(Z',0);
RT= R';
R22T= RT(r+1:end,r+1:end);

%% c) Noise reduction and SVD
% C2= sqrtm(CG);
% C2= eye(r);
% [Lsv,Sv,Rsv]= svd(pinv(C2)*R22T);
% Or= C2*Lsv(:,1:na);% Estimation of extended observability matrix
[Lsv,~,~]= svd(R22T);
Or= Lsv(:,1:na);% Estimation of extended observability matrix

%% d) Estimation of A and C
A= Or(1:r-1,:)\Or(2:r,:);
C= Or(1,:);

%% e) Estimation of B and D

% Frequency variables
H= zeros(F,na+1);
Ina= eye(na);
for k=1:F
    H(k,:)= [C/(xi(k)*Ina - A),1];
end;
```

```
% Weight matrix
W= diag(1./stdG);

% Input matrices estimation
BD= pinv(real(H'*W*H))*real(H'*W*G); % cf Pintelon 2002 Appendix B.3
%% pinv to avoid conditionning issue
B= BD(1:na);
D= BD(na+1);
% B= zeros(a,1); D=0;

%% Output
A= A*wscale; C= C*wscale; % Frequency denormalization
SG= ss(A,B,C,D); % continuous model
G_= H*BD; % Estimated FRF
Cost=db(norm(W*G-W*G_)/norm(W*G)); % weighted relative error in dB
```

Appendix C

Codes for Nonlinear Optimization

```
function [E,F,r,loops]= LM_algo(lspkd,u,y,order, Np, option)

% Levenberg-Marquardt Algorithm on PNLSS model:
% nonlinear least-square optimization of polynomial coeff (PNLSS model)
%  that explains best the given input-output pair u,y.
%
% {input}
%       lspkd: digital time ss linear model {A, B, C, D}
%       u: input signal vector
%     y: measured output signal of unknown PLNSS model. Target to
   approach.
%       order: polynomial order
%       Np: size of monomial vector
%       option: to include or not input u in monomials (see PNLSS)
% {output}
%       E: polynomial coeff for PNLSS state equation
%       F: polynomial coeff for PNLSS output equation
%       r:  residual
%       loops: #loops run
```

```
%


%% initial parameters
Na= length(lspkd.a);
theta= zeros((Na+1)*Np,1);
%{
 theta contains all columns of E and F stacked on top of each other-
ZERO to start.
%}
lambda= 10; % Initial damping- high value favors steepest descent
alpha= 10; % Growth factor
epsilon= 1e-3*norm(y);% Final error threshold- TBD
imax= 10; %TBD
i=1; % run index



%% calculate initial residual r(theta), cost f= ||r||
[f,r]= cost(y, lspkd, theta,Na,Np,u,order,option);

%% calculate Jacobian J, and first candidate set of parameters theta_
J= jacobianPNLSSvsEF(lspkd,theta,order,u, Np,option);
G= J'*r; % Gradient
H= J'*J ;% Hessian
Hld= H+lambda*diag(diag(H));
theta_= theta - pinv(Hld)*G; % parameters update;

%% calculate new residual r_, new cost function f_;
[f_,r_]= cost(y, lspkd, theta_,Na,Np,u,order,option);
fprintf('run %3.0f; cost %3.2f dB\n',i, db(f_/norm(y)));

%% iterative search of best parameters vector theta
%
```

```
while ( isnan(f_)||(f_ > epsilon)) && (i < imax)% && (max(G)>epsilon_2)
        if (isnan(f_)||(f_ >= f))
                lambda = lambda*alpha;% increase (lambda);
                % do nothing; new parameters are rejected;
        else
                lambda = lambda/alpha; % decrease (lambda);
                theta= theta_; % accept new parameters;
                % calculate residual r(theta), cost function f
                [f, r]= cost(y, lspkd, theta, Na, Np, u, order, option);
                % calculate Jacobian J, gradient G,  Hessian H;
                J= jacobianPNLSSvsEF(lspkd,theta,order,u,Np,option);
                G= J'*r;% Gradient
                H= J'*J ;% Hessian
        end;
        Hld= H+lambda*diag(diag(H));
        theta_= theta - pinv(Hld)*G; % parameters update;
        % calculate new residual r_, new cost function f_;
        [f_,r_]= cost(y, lspkd, theta_, Na, Np, u, order, option);
        fprintf('run %3.0f; cost %3.2f dB\n',i+1, db(f_/norm(y)));
        i= i+1;
end;
theta= theta_; %final parameters estimate;
[E, F]= unflatten_theta(theta,Na,Np);
r= r_;
loops= i;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sub-Function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [f, r]= cost(y, lspkd, theta, Na, Np, u, order, option)
[E, F]= unflatten_theta(theta,Na,Np);
```

```
r= PNLSSfilter(lspkd,E,F,u,order,option)-y;
f= norm(r);
end


function J= jacobianPNLSSvsEF(sys,th,d,u,Np,option)
% Calculates the jacobian of sys output subject to u
% in regard to th= vector of nonlinear parameters.
% {input}
%   sys: linear part of PNLSS model
%   th: flattened NL polynomial matrices E, F of PLNSS model
%   d: polynomial degree
%   u: input model signal.
%   Np: size of monomial vector
%   option: option to include u in monomials
% {output}
%   J: jacobian matrix of sys output.
%   Each row is the gradient of the PNLSS model output sample vs.
%   the NL coeff vector th
%   J is T*N with T= length(u) and N= length(th)
%
%% Init Matrices
[A,B,C,D]= ssdata(sys);
Na= length(A);
[E,F]=  unflatten_theta(th,Na,Np);
x= zeros(Na,1);% state vector
dx= zeros(Na, Np, Na);% each dx(i,j,:)= dx/dEij is a vector of length Na
T= length(u);
J= zeros(T,length(th));
%% init monomials pwr's
switch option
        case 'x'        % polynomial in x only
                a= getmonomialpwrs(Na,d);
```

```
        case 'xu'       % polynomial in x and u
                a= getmonomialpwrs(Na+1,d);
        case 'lin'      % 'lin': linear option- no polynomial
                a= ones(length(F),1);
end;
%% Recursive Calculus of Jacobian
for t= 1:T
        switch option
                case 'x'        % polynomial in x only
                        p= monomials(x,a);
                case 'xu'       % polynomial in x and u
                        p= monomials([x;u(t)],a);
                case 'lin'      % 'lin': linear option- no polynomial
                        p= zeros(Np,1);
        end;
        dp= diffmonomials(p,a,x);  % jacobian dp/dx, matrix Np x Na
        for j= 1:Np
                for i= 1:Na
                        v= squeeze(dx(i,j,:)); % v= dx/dEij
                        J(t,i+(j-1)*(Na+1))= (C+F*dp)*v; %dy/dEij
                        v= (A+ E*dp)*v;
                        v(i)= v(i)+ p(j);
                        dx(i,j,:)= v; % update dx/dEij
                end; % i= Na
                J(t,j*(Na+1))= p(j);% dy/dFj
        end; % j= Np
        x= A*x+B*u(t)+E*p; % update state
end; % t= T
end
```

```
function [E, F]= unflatten_theta(q,na,np)
% Unflatten vector of polynomial coeff
% {input}
%       q: flat vector of polynomial coeff
%                 q is ((na+1)*np,1)
%       na first dim
%       np second dim
% {output}
%       E is (na,np)
%       F is (1,np)
%
Q= reshape(q,na+1,np);
E= Q(1:na,:);
F= Q(na+1,:);
```

```
function y= PNLSSfilter(linss ,E,F,u,d, option)
% Apply PNLSS system on the input vector u with nonlinearity of order d
% {in}
%         linss= linear ss system
%         E, F= monomial coeff of order d
%         u= input column vector
%         d= maximum polynomial order (>= 2)
%         option
%                  'lin ': linear option- no polynomial added to the SS
%                  equations
%                  'x ': polynomial in x only
%                  'xu ': polynomial in x and u (default)
% {out}
%         y= output column vector
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initializations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[A,B,C,D]= ssdata(linss);
y=0*u;
Na= length(A);
x= zeros(Na,1);% initial zero state
%%
if nargin== 5
         option= 'xu';% default option
end;
switch option % init monomials pwr's
         case 'x'          % polynomial in x only
                  a= getmonomialpwrs(Na,d);
         case 'xu'         % polynomial in x and u
                  a= getmonomialpwrs(Na+1,d);
         case 'lin'        % 'lin ': linear option- no polynomial
```

```
                a= ones(length(F),1);
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Apply PNLSS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for n=1:length(u)
        p= CalculateP(x,u(n),a,option);
        y(n)= C*x+D*u(n)+F*p;
        x= A*x+B*u(n)+E*p; % Calculates x(n+1)
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Subfunction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function q= CalculateP(x,un,pwr,option)
% Calculate monomial vector applying multi-index powers on elements of x
    and un
% according to option
% {in}
%       x, u: column vectors
%       pwr: monomials powers. pwr(i,j)= exponent of z(j) in monomial(i)
%       option
%               'lin': linear option- no polynomial added to the SS
%               equations
%               'x': polynomial in x only
%               'xu': polynomial in x and u (default)
% {out}
%       p: column vector of monomials of x
%
switch option % init monomials pwr's
        case 'x'
                q= monomials(x,pwr);
        case 'xu'
```

```
                q= monomials([x;un],pwr);
        case 'lin'
                q= zeros(size(pwr,1),1);
end;
```

```
function dp= diffmonomials(p,a,x)
% yields all monomials derivative of degree 2 to d from x1,...xn
% {in} p: column vector of monomials
%        a: matrix of monomials powers
%        x: column vector of monomials entries
% {out} dp: jacobian dp/dx, matrix Np x Nx
%
Np= length(p);
Nx= length(x);
dp= zeros(Np,Nx);
for n= 1:Np
        for m= 1:Nx
                dp(n,m)= a(n,m)*p(n)/(x(m)+eps);
        end;
end;
```

```
function a= getmonomialpwrs(Nx,d)
% yields all monomial multi-index powers of degree 2 to d for vector of
    size Nx
% {in}
%       Nx: vector size
%       d: degree >= 2
% {out}
%       a: matrix of monomials powers.
%           a(i,j)= exponent of x(j) in monomial(i)
%
Np= nchoosek(Nx+d,d)- 1- Nx;
a= zeros(Np,Nx);
k=1;
for r= 2:d
        [tmp, ar]= multinom(Nx,r); % ar: matrix of monomials powers of
    degree r
        L= length(ar);
        a(k:k+L-1,:)= flipud(ar);
        k= k+L;
end;
```

```
function p= monomials(x,a)
% Get vector p of monomial values by applying multi-index powers a on x
% {in}
%       x: column vector
%       a: monomials powers. a(i,j)= exponent of x(j) in monomial(i)
% {out}
%       p: column vector of monomials of x
%
Nx= length(x);
Np= size(a,1);
p= ones(Np,1);
for np= 1:Np
        for nx= 1:Nx
                p(np)= p(np)*x(nx)^a(np,nx); % apply powers on x values
        end;
end;
```

Appendix D

Codes for System Reduction and Stabilization

```
function [p_,z_]= prunepz(sys)
% prunes poles, zeros of a sys
%{in}
%       sys= system
%{out}
%       p_= reduced list of poles
%       z_= reduced list of zeros

[p_,z_]= pzmap(sys);
[p_]= removepz(p_,3); % Trim remote poles first
[z_]= removepz(z_,3); % Trim remote zeros

% disp('zeros added at DC');
% z_= [z_; 1];
% shownewsys(p_, z_, sys); % plot new frf along with old
% waitforbuttonpress;
% disp('zeros added at Nyquist');
% z_= [z_; -1];
```

```
% shownewsys(p_, z_, sys); % plot new frf along with old
% waitforbuttonpress;
stblsys= MinPhaseSS(zpk(z_,p_,1,1));
[p_,z_]= pzmap(stblsys);
%
[p_,z_]= prunePZpairs(p_,z_,0.3);
shownewsys(p_, z_, sys); % plot new frf along with old

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [z_]= removepz(z, thr)
% remove values in z, with mag bigger than thr or
% smaller than 1/thr
% {in}
%        z= vector of complex values
%        thr= threshold on mag
% {out}
%        z_= trimmed vector
n=1;
while (n <= length(z))
        logdist= abs(db(z(n))); % log distance from unit circle
        if logdist > db(thr)
                z(n,:)=[]; % remove value
        else
                n= n+1; % move to next
        end;
end;
z_= z;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function shownewsys(p, z, sys0)
% plot new frf and pz map along with old
% p, z= poles, zeros list
% sys0= old sys for ref
%
dt= sys0.Ts;
sys= zpk(z,p,1,dt);
figure(1);
pzplot(sys0,'g',sys,'r');
legend('new map', 'original');
%
figure(2);
f= logspace(log10(20),log10(20e3),301);
G0= squeeze(freqresp(sys0,2*pi*f));
G_= squeeze(freqresp(sys, 2*pi*f));
semilogx(f, db([G_, G0]));
legend('new frf', 'original');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [p_,z_]= prunePZpairs(p,z,thr)
% remove p-z pairs that are closer than some threshold
% {in}
%         p, z: poles, zeros list
%         thr: threshold parameter
% {out}
%         p_, z_: trimmed poles, zeros list
k= 1;
while k <= length(p)
        nmin=1;
        dist= abs(p(k)- z(1));
        for n=2:length(z) % search closest zero to p(k)
                newdist= abs(p(k)- z(n));
```

```
                if newdist < dist
                        nmin= n;
                        dist= newdist;
                end;
        end;
        if dist/abs(exp(1j*angle(p(k)))- p(k)) < thr
                p(k,:)= [];             %remove value
                z(nmin,:)=[];           %remove value
        else
                k= k+1; % move to next
        end;
end;
p_= p; z_= z;
end
```

```
function stblsys= StabilizeSS(badsys)
% Stabilize a SS system by fixing the unstable poles.
% The outstanding poles are mirrored back inside the unit circle.
% The fixed SS has the same FRF magnitude as the bad one.
% ({in} badsys= unstable SS
% {out} stblsys= stable SS
%
[z,p,g]= zpkdata(badsys); % yield cells
p= p{1}; %extract data inside cell
z= z{1}; %extract data inside cell
for k= 1:length(p)
        if abs(p(k)) >1
                p(k)= 1/conj(p(k));
                g= g*abs(p(k)); % correct overall gain
        end;
end;
Ts= badsys.Ts; % Sampling period
stblsys= zpk(z,p,g,Ts);
stblsys= ss(stblsys); % translation to SS
```

```
function [G, z]= zpplot3D(zeros,poles,range)
% plot the rational complex function defined by zeros & poles in 3D
% {in}
%       zeros= list of zeros
%       poles= list of poles
%       range= [xmin, xmax, ymin, ymax zmin zmax] for plot
% {out}
%       z= matrix of complex numbers used a plotting domain
%       G= function plotted
%
x=linspace(range(1),range(2),101);
y=linspace(range(3),range(4),101);
[x,y]=meshgrid(x,y);
z= x+1i*y;
G= z*0;
[N,M]= size(G);
% %
for n= 1:N
        for m= 1:M
                num= prod(z(n,m)- zeros);
                den= prod(z(n,m)- poles);
                G(n,m)=db(num/den);
        end;
end;
G= max(G,range(5));
G= min(G,range(6));
%%
figure
surfl(x,y,G)
shading interp
colormap(copper)
```

```
grid on
box on
% axis equal
xlabel('real')
ylabel('imag')
zlabel('dB')
title(['Zeros: ', num2str(zeros'), '; Poles: ',num2str(poles')]);
```

```
function [p,z]= fo_pzmap(fosys,info,f)
%FO_PZMAP display the (poles-zeros)^alpha map of a fractional order
    system along with the stability limits.
%
% SYNOPSIS: [p,z]= fo_pzmap(fosys,info,f)
%
% INPUT fosys: fractional order system
%    info: [diff order, ss order, total error]
%
% OUTPUT p: list of poles^alpha
%       z: list of zeros^alpha
%              f: frequency range of stability limits
% REMARKS
%
% created with MATLAB ver.: 7.14.0.739 (R2012a) on Mac OS X  Version:
    10.8.4 Build: 12E55
%
% created by: Pascal Brunet
% DATE: 29-Jun-2013
%
%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% plot pzmap
alpha= info(1);
bestn= info(2);
besterr= info(3);

[p,z]= pzmap(fosys);
phi= alpha*pi/2;
if nargin==3
```

```
    b= (2*pi*f).^alpha.*exp(1j*phi);
else
    rmax= ceil(max([real([p;z])/cos(phi);imag([p;z])/sin(phi)]));
    b= (0:1:rmax)*exp(1j*phi);% stability borderline
end;
plot(real(p),imag(p),'xb',real(z),imag(z),'ob',...
    real(b),imag(b),'r',real(b),-imag(b),'r');
grid;

xlabel('real'), ylabel('imag');
info= ['Best fit','- Diff Order: ', num2str(alpha,2),...
    '- SS Order: ',num2str(bestn)...
    '- ID Error= ',num2str(besterr,3),'dB'];
title(info);
```

Bibliography

[1] P. Brunet and S. Temme, "A new method for measuring distortion using a multitone stimulus and non-coherence," in *Audio Engineering Society Convention 121*, October 2006.

[2] A. J. M. Kaizer, "Modeling of the nonlinear response of an electrodynamic loudspeaker by a volterra series expansion," *J. Audio Engineering Society*, vol. 35, no. 6, pp. 421–433, 1987.

[3] W. Klippel, "The mirror filter- a new basis for reducing nonlinear distortion and equalizing response in woofer systems," *J. Audio Engineering Society*, vol. 40, no. 9, pp. 675–691, 1992.

[4] H.-K. Jang and K.-J. Kim, "Identification of loudspeaker nonlinearities using the narmax modeling technique," *J. Audio Engineering Society*, vol. 42, no. 1/2, pp. 50–59, 1994.

[5] J. Suykens, J. Vandewalle, and J. van Ginderdeuren, "Feedback linearization of nonlinear distortion in electrodynamic loudspeakers," *J. Audio Engineering Society*, vol. 43, no. 9, pp. 690–694, 1995.

[6] J. Backman, "A non-linear model of a small transducer," in *Microphones and Loudspeakers, Audio Engineering Society 13th Conference on*, March 1998.

[7] M. J. Reed and M. J. Hawksford, "Practical modeling of nonlinear audio systems using the volterra series," in *Audio Engineering Society Convention 100*, May 1996.

[8] A. Bright, "Active control of loudspeakers: an investigation of practical applications," Ph.D. dissertation, Orsted DTU, Technical University of Denmark, Lyngby, Denmark, 2002.

[9] K. Lashkari, "A novel volterra-wiener model for equalization of loudspeaker distortions," in *Acoustics, Speech and Signal Processing, IEEE International Conference on*, vol. 5, May 2006.

[10] F. Agerkvist, "Modelling loudspeaker non-linearities," in *Audio Engineering Society, 32nd Int. Conference on DSP For Loudspeakers*, September 2007.

[11] B. Pedersen, "Error correction of loudspeakers," Ph.D. dissertation, Aalborg Universitet, 2008.

[12] M. Rebillat, R. Hennequin, E. Corteel, and B. F. G. Katz, "Prediction of harmonic distortion generated by electro-dynamic loudspeakers using cascade of hammerstein models," in *Audio Engineering Society Convention 128*, May 2010.

[13] W. Klippel, "Tutorial: Loudspeaker nonlinearities—causes, parameters, symptoms," *J. Audio Engineering Society*, vol. 54, no. 10, pp. 907–939, 2006.

[14] N. Thiele, "Loudspeakers in vented boxes: Part 1," *J. Audio Engineering Society*, vol. 19, no. 5, pp. 382–392, 1971.

[15] ——, "Loudspeakers in vented boxes: Part 2," *J. Audio Engineering Society*, vol. 19, no. 6, pp. 471–483, 1971. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=2163

[16] R. H. Small, "Closed-box loudspeaker systems-part 1: Analysis," *J. Audio Engineering Society*, vol. 20, no. 10, pp. 798–808, 1972. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=2022

[17] ——, "Closed-box loudspeaker systems-part 2: Synthesis," *J. Audio Engineering Society*, vol. 21, no. 1, pp. 11–18, 1973. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=2011

[18] M. Schetzen, *The Volterra and Wiener theories of nonlinear systems.* Krieger Pub., 2006.

[19] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Audio Engineering Society Convention 108*, February 2000.

[20] P. Brunet and B. Shafai, "State-space modeling and identification of loudspeaker with nonlinear distortion," in *Modelling, Identification, and Simulation, IASTED International Conference on*, vol. 755, November 2011.

[21] ——, "New trends in modeling and identification of loudspeaker with nonlinear distortion," in *Proceedings 2011 Intl Conf. on Modeling, Simulation and Visualization Methods*, ser. WorldComp'11, H. R. Arabnia and L. Deligiannidis, Eds., vol. MSV, World Academy of Science. CSREA Press, 2011, pp. 140–145.

[22] J. Schoukens, J. G. Nemeth, P. Crama, Y. Rolain, and R. Pintelon, "Fast approximate identification of nonlinear systems," *Automatica*, vol. 39, no. 7, pp. 1267 – 1274, 2003.

[23] J. Schoukens, R. Pintelon, T. Dobrowiecki, and Y. Rolain, "Identification of linear systems with nonlinear distortion," *Automatica*, vol. 41, no. 3, pp. 491 – 504, 2005.

[24] J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens, and R. Pintelon, "Identification of nonlinear systems using polynomial nonlinear state space models," *Automatica*, vol. 46, no. 4, pp. 647 – 656, 2010.

[25] T. McKelvey, H. Akcay, and L. Ljung, "Subspace-based multivariable system identification from frequency response data," *Automatic Control, IEEE Transactions on*, vol. 41, no. 7, pp. 960 –979, July 1996.

[26] R. Pintelon and J. Schoukens, *System identification: a frequency domain approach*, 2nd ed., ser. Control Process and Measurements. Wiley-IEEE Press, April 2012.

[27] R. Pintelon, "Frequency-domain subspace system identification using non-parametric noise models," *Automatica*, no. 38, pp. 1295–1311, 2002.

[28] J. Paduart, "Identification of nonlinear systems using polynomial nonlinear state space models," Ph.D. dissertation, Vrije University Brussels, B-1000 Brussels, 2007.

[29] A. Ranganathan, "The levenberg-marquardt algorithm," Georgia Institute of Technology, Tech. Rep., June 2004.

[30] O. T. K. Madsen, H.B. Nielsen, "Methods for non-linear least squares problems," ebook, April 2004.

[31] U. M. Al-Saggaf and G. F. Franklin, "Model reduction via balanced realizations: An extension and frequency weighting techniques," *IEEE Transactions On Automatic Control*, vol. 33, no. 7, pp. 687–692, Jul 1988.

[32] E.C.Levy, "Complex curve fitting," *IRE Transactions on automatic control*, vol. AC-4, no. 5, pp. 37–43, May 1959.

[33] P. Brunet and B. Shafai, "A fractional order approach to loudspeaker identification," 2013, journal paper unpublished.

[34] A. Loverro, "Fractional calculus: History, definitions and applications for the engineer," Department of Aerospace and Mechanical Engineering- University of Notre Dame, Tech. Rep., July 2007.

[35] R. Magin, M. D. Ortigueira, I. Podlubny, and J. Trujillo, "On the fractional signals and systems," *Signal Processing*, vol. 91, no. 3, pp. 350 – 371, 2011.

[36] I. Petrás, "Stability of fractional-order systems with rational orders: a survey," *Fractional Calculus & Applied Analysis*, vol. 12, no. 3, pp. 269–298, 2009.

[37] M. S. Tavazoei and M. Haeri, "A note on the stability of fractional order systems," *Mathematics and Computers in Simulation*, vol. 79, no. 5, pp. 1566–1576, 2009.

[38] S. Ahmadizadeh, A. Montazeri, and J. Poshtan, "Identification of a stable model for a flexible plate using frequency domain subspace methods," in *World Congress*, vol. 18, no. 1, 2011, pp. 6517–6522.

[39] L. Balogh and R. Pintelon, "Stable approximation of unstable transfer function models," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 12, pp. 2720–2726, 2008.

[40] T. D'haene, R. Pintelon, and G. Vandersteen, "An iterative method to stabilize a transfer function in the s- and z-domains," *Instrumentation and Measurement, IEEE Transactions on*, vol. 55, no. 4, pp. 1192–1196, 2006.

[41] R. Caponetto, *Fractional order systems: modeling and control applications*. World Scientific, 2010, vol. 72.

[42] M. Tavakoli-Kakhki, M. Haeri, and M. S. Tavazoei, "Notes on the state space realizations of rational order transfer functions," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 58, no. 5, pp. 1099–1108, 2011.

[43] J. Vanderkooy, "A model of loudspeaker impedance incorporating eddy currents in the pole structure," in *Audio Engineering Society Convention 84*, March 1988.

[44] J. G. Knudsen, Morten H.; Jensen, "Low-frequency loudspeaker models that include suspension creep," *J. Audio Engineering Society*, vol. 41, no. 1/2, pp. 3–18, 1993.

[45] F. Blasizzo, "A new thermal model for loudspeakers," *J. Audio Engineering Society*, vol. 52, no. 1/2, pp. 43–56, 2004.

[46] J.R.Wright, "An empirical model for loudspeaker motor impedance," *J. Audio Engineering Society*, vol. 38, no. 10, pp. 749–754, October 1990.

[47] J. W. Marshall Leach, "Loudspeaker voice-coil inductance losses: circuit models, parameter estimation, and effect on frequency response," *J. Audio Engineering Society*, vol. 50, no. 6, pp. 442–450, June 2002.

[48] K. Thorborg, A. D. Unruh, and C. J. Struck, "An improved electrical equivalent circuit model for dynamic moving coil transducers," in *Audio Engineering Society Convention 122*, May 2007.

[49] I. Schäfer and K. Krüger, "Modelling of lossy coils using fractional derivatives," *Journal of Physics D: Applied Physics*, vol. 41, no. 4, p. 045001, 2008.

[50] T. T. Hartley and C. F. Lorenzo, "Fractional-order system identification based on continuous order-distributions," *Signal Processing*, vol. 83, no. 11, pp. 2287 – 2300, 2003.

[51] P. Nazarian, M. Haeri, and M. S. Tavazoei, "Identifiability of fractional order systems using input output frequency contents," *ISA Transactions*, vol. 49, no. 2, pp. 207–214, 2010.

[52] K. Zhou and J. C. Doyle, *Essentials of robust control*. Prentice hall Upper Saddle River, NJ, 1998, vol. 104.

[53] S. Lacy and D. Bernstein, "Subspace identification with guaranteed stability using constrained optimization," *Automatic Control, IEEE Transactions on*, vol. 48, no. 7, pp. 1259–1263, 2003.

[54] L. Dorcak, I. Petras, I. Kostial, and J. Terpak, "Fractional-order state space models," *arXiv preprint math/0204193*, 2002.

[55] L. Vanbeylen, "Nonlinear lfr block-oriented model: Potential benefits and improved, user-friendly identification method," *Instrumentation and Measurement, IEEE Transactions on*, vol. 62, no. 12, pp. 3374–3383, 2013.

[56] A. Falaize-Skrzek and T. Hélie, "Simulation of an analog circuit of a wah pedal: A port-hamiltonian approach," in *Audio Engineering Society Convention 135*, Oct 2013.