

Motional Feedback Control for Bass Loudspeakers

Subgroup Theory

by

Aart-Peter Schipper & Alexandros
Skourtis-Cabrera

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Tuesday July 3, 2018 at 13:30.

Student number: 4469143, 4392639
Project duration: April 23, 2018 – July 6, 2018
Thesis committee: Prof. Said Hamdioui TU Delft, chairman
Dr. ir. G. J. M. Janssen, TU Delft, supervisor
Dr. S. Izadkhast, TU Delft, jury member

Preface

This thesis was meant to extensively cover the nonlinear control theory of motion feedback loudspeaker systems. We realized how useful it would be to have a nonlinear model of the loudspeaker in question, to demonstrate the effect of any controller. Creating such a model, however, took so much time that analyzing nonlinear controllers was never endeavoured. Nevertheless, the measurement of a nonlinear system turned out to be quite interesting in itself.

We would like to thank our supervisor, Dr. Janssen, for his interest and support throughout the course of the project. We would also like to thank the Tellegen Hall personnel for their help and for providing most of the measuring equipment. Finally, we want to thank the jury members for evaluating our results.

"I have found ten thousand ways that won't work." - Thomas Edison

*Aart-Peter Schipper & Alexandros Skourtis-Cabrera
Delft, June 2018*

Abstract

The aim of this research is to create a nonlinear model of a loudspeaker to analyze the open-loop distortion as well as the closed loop performance with linear and nonlinear controllers. A method is proposed for measuring the dominant nonlinear parameters of a loudspeakers. Furthermore, the loudspeaker distortion is both measured experimentally and simulated using the nonlinear model. The method for nonlinear system identification suffers from poor accuracy and takes into account neither the Eddy current losses, frequency dependent compliance and damping nor visco-elastic effects of the loudspeaker surround material. The simulations and measurements of distortion are not in agreement.

Contents

1	Introduction	1
1.1	Programme of Requirements	2
1.1.1	Requirement formulation	2
1.1.2	Study-case	3
2	Theory	5
2.1	Ideal Loudspeaker	5
2.1.1	Extension to Three Dimensions.	7
2.1.2	Acoustic Intensity	9
2.2	Linear Loudspeaker	9
2.2.1	Mass-Spring-Damper Subsystem	10
2.3	Non-linear Loudspeaker	10
2.3.1	Causes of Non-linearities in Loudspeakers	11
2.3.2	Effects of Non-linearities in Loudspeakers	13
3	Non-linear Modelling of the Loudspeaker	17
3.1	Free-Body Analysis	17
3.2	State Space Model	18
3.3	Possibilities in non-linear control	19
4	Impedance Measurement	21
4.1	Method for Impedance Measurement	21
4.1.1	Mass Measurement.	22
4.1.2	Measurements with Offset	22
4.2	Results	24
4.2.1	Mass Measurement Results	25
4.2.2	Results of Impedance Measurements with Offsets.	28
4.3	Discussion of Impedance Measurements	31
5	Model Verification	33
5.1	Solving for $K_m(x)$ and $BI(x)$	33
5.2	Simulating Large Signal Behaviour of the Loudspeaker	34
5.3	Discussion of Simulation Results	34
6	Conclusion	37
6.1	Recommendations	37
A	MATLAB code	39
A.1	Model Validation	39
A.1.1	plotTHD.m	39
A.1.2	simTHD.m	41
A.2	Impedance Measurement	45
A.2.1	nl_imp_meas.m	45
A.2.2	nl_imp_extract.m.	47
A.2.3	nl_imp_analysis.m	48
A.2.4	imp_datatoimp.m.	50
A.3	Parameter Estimation.	53
A.3.1	pars_nl.m	53
A.3.2	pars_nl_old.m	56

A.4 Other.	59
A.4.1 fourier.m	59
A.4.2 playRec.m	59
A.4.3 gen_freq.m	63
A.4.4 record_freq.m	63
A.4.5 sim_freq.m	64
B Measurement Equipment	67
Bibliography	69



Introduction

A loudspeaker, like any real electro-mechanical transducer, is a non-ideal device with physical properties and limitations. At low signal amplitudes, where its behaviour can be approximated as linear, the speaker manifests distortion of the input signal in the form of non-flat transfer. At high excursions of the cone and especially when reproducing lower frequencies where high amplitudes are needed to generate the same audio power, non-linearities in the electrical and mechanical properties of the speaker cause additional deformation of the sound in the form of audible harmonic and intermodulation distortion.

One way to reduce the detrimental effects of both the linear and non-linear behaviour of the speaker is by using feedback to correct for this distortion. Some sources of feedback signals that have been used are the back EMF of the speaker voice coil [20] and that of a secondary voice coil mounted on the diaphragm [7], but these methods are not sufficient to produce the best possible results.

The concept of Motional Feedback will be studied and applied as our Bachelor's Graduation Project. The team working on this project consists of three sub-groups each working on a different implementation of this concept, namely: the analogue, digital and theory group. It is meant that the digital group will try to come up with a digital implementation of this motional feedback controller. The analogue group is expected to design an analogue implementation of this system. The theory group will parameterize the speaker in order to create a model of the non-linear loudspeaker and work towards designing an optimal controller. This model as well as the measurement setup and code used to measure the performance of the speaker will be used to validate the controller designed by the other two subgroups. This thesis will cover the work done by the theory sub-group.

In 1968, a motional feedback system was proposed by Philips [13] to suppress linear and non-linear distortion in bass speakers. The system used a piezoelectric accelerometer mounted on the speaker cone to measure its acceleration. The recorded signal was fed back to a control system which compensated for the distortion and improved the performance of the speaker.

Advances in digital technology and methods in modeling and design have allowed for the design of more advanced controllers, and a few implementations of motional feedback systems have been attempted [1] [27]. As part of the Bachelor graduation project of Electrical engineering at TU-Delft, this thesis presents the parameterisation and modelling of a black box non-linear loudspeaker for the purpose of designing an optimal motional feedback system. A linear model can be used to design the feedback controller, but a non-linear model of the system is needed to design an optimal non-linear system, as well as to investigate the effects of a linear controller on a non-linear speaker.

Several successful attempts have been made in the parameterisation and modelling of a non-linear loudspeaker [8] [14]. However, these methods rely on the precise knowledge of the excursion of the loud-

speaker at any time during measurement, which requires the use of specialized equipment such as a laser interferometer. In the absence of such specialized equipment or the necessary budget therefor, a different methodology is required for the parameterization of a non-specified, black box loudspeaker. To this end and to facilitate the design and testing of a motional feedback system for bass loudspeakers by the two other subgroups, a different methodology is proposed for the non-linear analysis of a loudspeaker. This methodology relies on impedance measurements of the speaker using simple circuitry, a USB soundcard and computational methods in MATLAB and Simulink.

First, the programme of requirements for the Motional Feedback System is listed. Then, the underlying theory of the function of the ideal, linear and non-linear speakers is explained in chapter 2. The derivation of the state space model and equivalent electrical circuit of the system is presented in chapter 3, followed by a detailed description of the method used to parameterize the speaker in chapter 4. The results of the impedance measurements are then listed and discussed and the model is validated.

The deliverables of the theory sub-group are (1) MATLAB code and a measurement setup to be used for the evaluation of a loudspeaker system's performance in terms of linear and non-linear distortion, (2) a methodology and measurement setup that allows for the non-linear parameterization of a loudspeaker and (3) a theoretical non-linear model of the loudspeaker implemented using MATLAB and Simulink. By the end of the project, a theoretical controller implemented in MATLAB and Simulink may also be delivered.

1.1. Programme of Requirements

The products to be developed are an analogue and a digital implementation of a motional feedback system for a bass loudspeaker using the feedback signal of a piezo-electric accelerometer mounted on the speaker cone, as well a theoretical model of a loudspeaker and motional feedback controller. The system is a low-cost, small format implementation which can easily be adjusted to be used for different speakers with different characteristics. The system is aimed towards commercial loudspeaker manufacturers to be included in active loudspeaker systems. The consumer good must meet or improve on the specifications listed in section 1.1.1 when using motional feedback. It also has to be available for a lower price than other motional feedback loudspeaker systems with similar specifications available on the market.

1.1.1. Requirement formulation

1. MR: mandatory requirements

- A woofer loudspeaker diaphragm is equipped with a piezo-electric accelerometer. The signal thereof is to be included in a negative feedback loop; this principle is known as Motional Feedback (MFB);
- The system should operate in a bandwidth from 10 – 300 Hz, however, a 1 kHz bandwidth is highly desirable. The highest attainable bandwidth is 2 kHz due to sensor limitations;
- The cost of the system should be no more than € 100 .
- The volume of the controller should be 0.5 L maximum;
- The Total Harmonic Distortion (THD) should be reduced to 0.1%;
- The largest acceptable delay that is introduced as a result of the controller is 120 ms. This is the delay that the user may experience when playing sound through the system;
- The power consumption of the controller should be 100 mW.
- The theoretical model of the loudspeaker must be accurate enough that the relative error in the simulated and measured Total HD is not larger than 1% in the bandwidth stated above.

2. ToRs: Trade-off requirements

- The desired Signal to Noise Ratio (SNR) is at least 100 dB. Nevertheless, a 16 bit digital system may offer some advantages due to faster communication possibilities and lower cost. The SNR of a 16 bit system is at most 80 dB, but this acceptable also;

- The system is optimised for the specific loudspeaker and amplifier that have been made available for this project. The system should ideally be also applicable to other configurations, considering the typical amplifier gain is 20 – 30 dB.
- The system must be stable, which implies that both the gain and phase margins must be reasonable. Precise minima were not given, but a phase margin of 45 degrees was proposed, alongside a gain margin of 3 dB.

1.1.2. Study-case

1. Functional Requirements

- (a) The MFB system must operate whenever the loudspeaker system is turned on without requiring additional steps from the user.
- (b) The loudspeaker system's user interface may contain a switch to turn motional feedback on and off.

2. System Requirements

(a) Utilisation features

- i. The lifespan of the feedback controller and accelerometer must be at least as long as the lifespan of the loudspeakers in which it is included.
- ii. If support and/or maintenance is provided for the loudspeaker system, this must include support for the MFB system.

(b) Production and putting into use features

- i. Inclusion of the MFB system must take place during the development of the loudspeaker system in cooperation between the loudspeaker manufacturer and the company implementing motional feedback.
- ii. The loudspeaker must undergo testing by the company before and after the inclusion of the MFB system to ensure MFB meets performance specifications.
- iii. The company implementing motional feedback will provide the piezo-electric accelerometer and controller to the loudspeaker manufacturer. The manufacturer must install the MFB hardware into the consumer product during assembly. Placement of the controller inside the loudspeaker will be discussed with the manufacturer on a case by case basis.

(c) Discarding features

- i. If the hardware of the MFB system is enclosed in a casing, the casing must be made from recyclable materials.
- ii. In case the MFB system's lifespan exceeds that of the speaker itself, the manufacturer must provide to the consumer the option of returning the MFB hardware for use in a refurbished product

3. Development of manufacturing methodologies

- (a) The digital version of the MFB controller will be implemented as an ASIC.
- (b) The ASIC must be adjustable after manufacturing to meet specifications in any loudspeakers in which it is included; Only one version of the ASIC will be developed and manufactured.
- (c) The theoretical model of the loudspeaker and controller will be implemented in MATLAB and Simulink.
- (d) A protocol and measurement setup will be developed for quick testing and validation of the loudspeaker system before and after the inclusion of MFB. Testing on a loudspeaker must not take longer than 20 minutes.

4. Liquidation/recycling methodologies

- (a) At the end of the product's lifespan, the discarding thereof must comply to the norms referring to processing of small chemical waste.

5. Business strategies, marketing an sales opportunities

- (a) The manufacturer of the loudspeaker must explicitly state the inclusion of the MFB feature on the packaging and documentation of the final product
- (b) the logo of the company implementing motional feedback must be included on the packaging and casing of the final product by the manufacturer.

2

Theory

An extremely simplified model of a woofer loudspeaker is given in figure 2.1. The system is considered mass-less, with the coil having infinite conductance. The coil is enveloped by a homogeneous, inward pointing magnetic flux density \mathbf{B} , that is orthogonal to the coil windings. Such a magnetic field cannot be generated in practice, since $\nabla \cdot \mathbf{B}$ must vanish. Nevertheless, it is possible to create a field that has properties similar to that of the ideal distribution, as will be discussed in section 2.2. The half spaces that are separated by the piston extend to infinity and are filled with air with equilibrium density ρ_0 and equilibrium (hydrostatic) pressure P_0 . The present analysis is limited to one dimension, but following the same principles, the equations for three dimensions can be derived as is done in section 2.1.1¹. Thus, the velocity \mathbf{u} and acoustic pressure p are assumed to be a function of x , the only spatial dimension, and time t . The derivations in this chapter are based on [19] and [3].

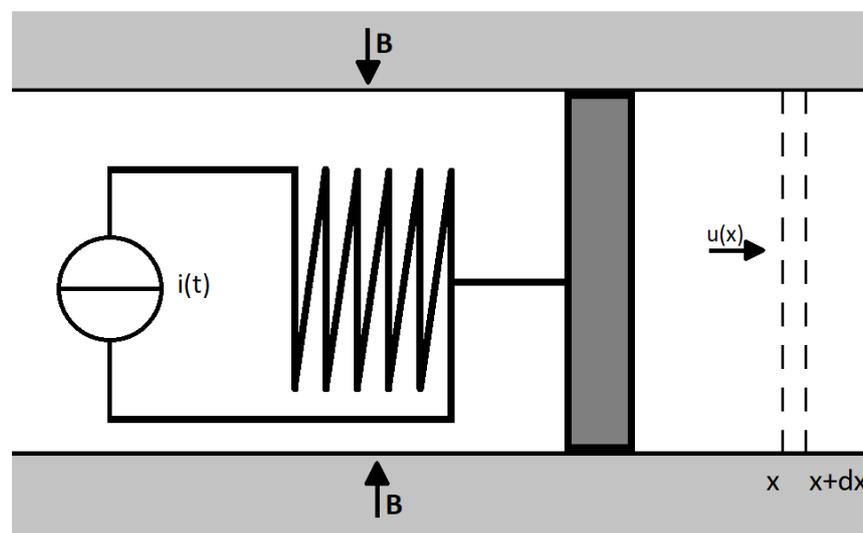


Figure 2.1: Simplified model of a loudspeaker with one dimensional acoustic environment

2.1. Ideal Loudspeaker

Referring to figure 2.1, we will consider the infinitesimal static volume $dV = Adx$, where A is the surface of the piston. If there is a flow of air molecules in the positive x direction, the mass balance of particles in

¹A thorough derivation of the linearized wave equation, listing all assumptions can be found in [19].

the volume can be evaluated; the result is given in equation 2.1. The mass increase is denoted by dm and W is the volume displacement.

$$\frac{\partial}{\partial t} m = \rho(x) \frac{\partial}{\partial t} W(x) - \rho(x+dx) \frac{\partial}{\partial t} W(x+dx) \quad (2.1)$$

The mass increase can be related to the change in density: $dm = A\rho dx$. Furthermore, the rate of change of volume displacement can be related to the particle velocity \mathbf{u} : $\frac{\partial}{\partial t} W = A\mathbf{u}$. Henceforth, the local rate of change of density can be computed using equation 2.2.

$$\frac{\partial}{\partial t} \rho = - \frac{\rho(x+dx)\mathbf{u}(x+dx) - \rho(x)\mathbf{u}(x)}{dx} = \frac{\partial}{\partial x}(\rho\mathbf{u}) \quad (2.2)$$

The expression of equation 2.2 can be simplified by assuming that the pressure can be written as $\rho = \rho_0(1+s)$. The variable s is the condensation, which is assumed to be very small, i.e. $s \ll 1$. Furthermore, ρ_0 is assumed to be approximately constant in space and time. The resulting equation is given in 2.3 and is known as the linear continuity equation (in one dimension).

$$\frac{\partial}{\partial t} s + \frac{\partial}{\partial x}(\rho_0\mathbf{u}) = 0 \quad (2.3)$$

Referring back to figure 2.1, we will now consider the volume dV to move along with the airflow. Equation 2.4 relates the force that is exerted on the volume to the pressure difference across the differential volume. p is the acoustic pressure, which is related to the instantaneous pressure P by $p = P - P_0$. The effects of gravity are ignored.

$$d\mathbf{F} = p(x)A - p(x+dx)A = - \frac{\partial}{\partial x} p dV \quad (2.4)$$

The force $d\mathbf{F}$ of equation 2.4 can be related to the acceleration of the particles via Newton's second law: $d\mathbf{F} = \mathbf{a}dm$. Using $dm \approx \rho_0 dV$ and $\mathbf{a} \approx \frac{\partial}{\partial t} \mathbf{u}$, we arrive at the linearized Euler's equation, that is given in 2.5. This equation is valid if spatial variations of \mathbf{u} are much smaller than temporal variations of \mathbf{u} and if the condensation is small. These approximations are valid for acoustic excitations of low amplitude.

$$\rho_0 \frac{\partial}{\partial t} \mathbf{u} = - \frac{\partial}{\partial x} p \quad (2.5)$$

Acoustic processes are approximately adiabatic [19] [3], i.e. there is no heat transfer or work done on the system. This assumption is valid if the temperature gradients that arise due to acoustic excitation are small. Furthermore, the assumption implies that the dissipation of acoustic energy is small. For an adiabatic process, the following relation holds, where Γ is a constant:

$$PV^\Gamma = \Gamma \quad (2.6)$$

The exponent γ in equation 2.6 is the ratio of heat capacities: $\gamma = \frac{c_p}{c_v}$. A fundamental derivation of these thermodynamic properties can be found in e.g. [24]. For an ideal, diatomic gas, $\gamma = \frac{7}{5}$. Considering a system of fixed mass m , the pressure and density can be related to the equilibrium pressure and density: $P = P_0 \left(\frac{\rho}{\rho_0}\right)^\gamma$. Linearizing this equation and substituting $p = P - P_0$ yields the expression of 2.7.

$$p = \frac{\partial p}{\partial \rho}(\rho_0)(\rho - \rho_0) = \rho_0 c_0^2 s \quad (2.7)$$

Now we have related the pressure to the condensation s using the constants ρ_0 and c_0 , which is the speed of sound. The above equation can be combined with equations 2.3 and 2.5. Firstly, temporal differentiation has to be applied to equation 2.3, and spatial differentiation to equation 2.5. The continuity equation becomes: $\frac{\partial^2}{\partial t^2} s + \frac{\partial^2}{\partial x \partial t} \mathbf{u} = 0$ and the Euler equation becomes: $\rho_0 \frac{\partial^2}{\partial x \partial t} \mathbf{u} = - \frac{\partial^2}{\partial x^2} p$. Now finally, the result of equation 2.7 can be utilized to formulate the linearized wave equation:

$$c_0^2 \frac{\partial^2}{\partial x^2} p = \frac{\partial^2}{\partial t^2} p \quad (2.8)$$

The solution to the wave equation can readily be found by means of separation of variables: $p = X(x)T(t)$. Using $-\omega^2$ as integration constant², and defining $k = \frac{\omega}{c_0}$, the full solution is given by the expression in equation 2.9³.

$$p(x, t) = \int_{-\infty}^{\infty} \Pi(\omega, k) e^{j(\omega t - kx)} d\omega \quad (2.9)$$

The factor $\Pi(\omega)$ contains the spectral information of the pressure signal. The exact solution depends on the boundary conditions that are imposed on the system. The boundary conditions of the system are created by the simplified loudspeaker of figure 2.1. The current that flows through the voice coil will generate a force that acts on the air via the piston. This force is known as the Lorentz force:

$$\mathbf{F}_L = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (2.10)$$

The electric field $\mathbf{E} = \mathbf{0}$, for the simplified case of figure 2.1. The total force that acts upon the piston can be found by applying 2.10 to charges inside the coil. We will assume that there is a homogeneous charge distribution σ_l along the coil. Furthermore, as indicated at the beginning of this chapter, the particle velocity \mathbf{v} is orthogonal to the magnetic flux density \mathbf{B} . The total can now be found by integration along the wire:

$$\mathbf{F}_L = \int_l \sigma_l \mathbf{v} \times \mathbf{B} dl = Bl i \quad (2.11)$$

Equation 2.11 indicates that the current through the voice coil is proportional to the force that is exerted on the piston, with Bl being the factor of proportionality. The pressure at the piston surface is: $p = \frac{\mathbf{F}_L}{A} = Bl i$. If it is assumed that the piston excursion is small compared to the wavelength of the acoustic signal⁴, the piston may be assumed to be stationary in $x = 0$. The boundary condition may thus be expressed as: $p(0, t) = \frac{Bl}{A} i(t)$. Now if the current is a sinusoidal signal $i = I e^{j\omega_0 t}$, the solution to the boundary condition problem can be found. Such a current signal is hardly arbitrary, since any signal can be decomposed into a infinite sequence of complex exponentials via the Fourier transform.

$$\Pi(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} p(0, t) e^{-j(\omega t)} dt = \frac{Bl I}{A 2\pi} \int_{-\infty}^{\infty} e^{j(\omega_0 - \omega)t} dt = \frac{Bl I}{A} \delta(\omega_0 - \omega) \quad (2.12)$$

The result of equation 2.12 can be used to compute the pressure as a function of time by inserting the expression for $\Pi(\omega)$ in equation 2.9. The resulting pressure is $p(x, t) = \frac{Bl I}{A} e^{j(\omega_0 t - k_0 x)}$, where k_0 is defined as $k_0 = \frac{\omega_0}{c_0}$. The fact that this pressure wave is a complex valued expression is not surprising, since the current was also complex valued. Obviously, it is not possible to create a complex valued current in practice, but this assumption simplifies the mathematics. Taking the real part of the current: $\text{Re}\{i(t)\} = I \cos(\omega_0 t)$ and the pressure: $\text{Re}\{p(x, t)\} = \frac{Bl I}{A} \cos(\omega_0 t + \phi(x))$, where $\phi(x) = k_0 x$. It is clear now that the current and pressure are proportional, except for a position dependent phase shift.

2.1.1. Extension to Three Dimensions

The wave equation in three dimensions is slightly more complicated than the one dimensional equation of 2.8. Using the Laplace operator ∇^2 , which can be expressed in Cartesian coordinates as: $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$, the linear 3D wave equation becomes:

$$c_0^2 \nabla^2 p = \frac{\partial^2 p}{\partial t^2} \quad (2.13)$$

The separation of variables technique can be applied once again by writing $p(\mathbf{r}, t) = R(\mathbf{r})T(t)$, where $\mathbf{r} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + z\hat{\mathbf{z}}$ is the position vector and $x\hat{\mathbf{x}}, y\hat{\mathbf{y}}, z\hat{\mathbf{z}}$ are unit vectors in the x, y and z direction respectively.

²A negative integration constant is chosen in the present investigation, but this is not necessary. The integration constant can be any complex number; a merit that can be used to describe lossy wave propagation.

³This solution comprises a wave travelling in the positive x direction. There exists a solution with a wave travelling in the opposite direction also.

⁴This assumption may seem completely arbitrary, since Newton's third law implies that the massless piston and coil move with the same speed as the acoustic wave. A practical system, however, has a much larger mechanical impedance than the air, making the approximation valid.

The equations that remain are ordinary differential equations, with either spatial or temporal dependence. Using the same integration constant as employed previously, the equation for $T(t)$ can be solved easily: $T(t) = e^{j\omega t}$, where the constant has been absorbed into the other equation. The second equation is the Helmholtz equation:

$$\nabla^2 R + k^2 R = 0 \quad (2.14)$$

It is possible to solve the Helmholtz equation using separation of variables using separation of variables, but it makes sense to assume spherical symmetry. The Laplace operator in spherical coordinates is: $\nabla^2 = \frac{1}{r^2} \frac{\partial}{\partial r} \left(\frac{1}{r^2} \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin \phi} \frac{\partial}{\partial \phi} \left(\sin \phi \frac{\partial}{\partial \phi} \right) + \frac{1}{r^2 \sin^2 \phi} \frac{\partial^2}{\partial \theta^2}$, with r the distance to the origin, θ the polar angle and ϕ the azimuthal angle. Fortunately, since we have assumed that the wave is spherically symmetrical, the partial derivatives with respect to ϕ and θ vanish. The differential equation that is left only depends on r :

$$\frac{\partial^2}{\partial r^2} R + \frac{2}{r} \frac{\partial}{\partial r} R + k^2 R \quad (2.15)$$

The above equation may seem complicated to solve, but the solution is actually rather easy to guess. From practical experience it is known that sound waves are attenuated over distance. Specifically, the sound pressure level decreases with $\frac{1}{r}$. A solution of the form $R(r) = \frac{1}{r} e^{-jk r}$ is therefore proposed. The first and second order derivatives of this function are: $\frac{\partial}{\partial r} R = -jkR - \frac{1}{r} R$ and $\frac{\partial^2}{\partial r^2} R = -k^2 R + j\frac{2k}{r} R + \frac{2}{r^2} R$. Clearly, this is a solution to the Helmholtz equation, provided that the wave is spherically symmetrical. The complete solution to the equation involves the superposition of all spectral components as indicated by equation 2.16.

$$p(r, t) = \int_{-\infty}^{\infty} \Pi(\omega) \frac{1}{r} e^{j(\omega t - kr)} d\omega \quad (2.16)$$

Note that this expression for p satisfies the wave equation in all points except in the origin, i.e. $r = 0$. This is not quite so surprising, as it is expected that a pressure source is located in the origin. The expression does, however, satisfy the inhomogeneous wave equation⁵ that is given in equation 2.17.

$$c_0^2 \nabla^2 p = \frac{\partial}{\partial t} p - \delta(\mathbf{r}) S(\omega t) \quad (2.17)$$

The Dirac delta function in equation 2.17 has been extended to three dimensions, such that $\iiint_V \delta(\mathbf{r}) dV = 1$ provided that the point $\mathbf{r} = \mathbf{0}$ is part of V ; the integral is zero otherwise. $S(\omega t)$ is an arbitrary input signal. To demonstrate that the expression previously found for p is indeed consistent with the inhomogeneous wave equation, both sides of the expression will be integrated over a volume that includes the origin:

$$c_0^2 \iiint_V \nabla^2 p dV = \iiint_V \frac{\partial^2}{\partial t^2} p dV - s(t) \quad (2.18)$$

The expression on the left can be simplified considerably by application of the divergence theorem and by assuming that the volume is a sphere with radius R_0 : $\iiint_V \nabla^2 p dV = \oint_S \nabla p \cdot d\hat{\mathbf{r}} =$, where $\hat{\mathbf{r}}$ is a unit vector pointing in the positive r direction. The gradient can thus be reduced to: $\nabla p = \frac{\partial}{\partial r} p \hat{\mathbf{r}}$, since the angular components are irrelevant for the chosen control volume. The resulting expression is given in equation 2.19.

$$c_0^2 \iiint_V \nabla^2 p dV = -4\pi c_0^2 \int_{-\infty}^{\infty} \Pi(\omega) (1 + jkR_0) e^{j(\omega t - kR_0)} d\omega \quad (2.19)$$

The second integral is slightly more complicated. However, by bringing the differential operator $\frac{\partial^2}{\partial t^2}$ inside all the integrals, it becomes a multiplication with a factor $-\omega^2$. Furthermore the order of the integrals can be rearranged to give the expression of 2.20.

$$\iiint_V \frac{\partial^2}{\partial t^2} p dV = \int_{-\infty}^{\infty} -2\pi\omega^2 \Pi(\omega) \iiint_V \frac{1}{r} e^{j(\omega t - kr)} dV d\omega \quad (2.20)$$

⁵The inhomogeneity of the wave equation could imply that there is a source term in the linearized continuity equation or the linearized Euler equation. The effects of gravity, viscosity and convection also result in source terms. A complete derivation was given by Lighthill [21].

The volumetric integral on the right hand side of equation 2.20 can readily be computed, which yields the expression: $\frac{2}{k^2} e^{j\omega t} (1 - jkR_0) e^{jkR_0} - 1$. With the integrals gone, except for the integral over $d\omega$, the input signal can be computed. The resulting equation for $S(t)$ is given below:

$$s(t) = 4\pi c_0^2 \int_{-\infty}^{\infty} \Pi(\omega) e^{j\omega t} d\omega \quad (2.21)$$

Applying the Fourier theorem once more, the spectral components can be found, so that the pressure can be calculated. Now let us assume that the input signal is given as $s(t) = S e^{j\omega_0 t}$:

$$\Pi(\omega) = \frac{S}{4\pi c_0^2} \int_{-\infty}^{\infty} e^{j(\omega_0 - \omega)t} dt = \frac{S}{2c_0^2} \delta(\omega_0 - \omega) \quad (2.22)$$

Using the result of equation 2.22 and equation 2.16, the pressure can be found. The result is: $p(r, t) = \frac{S}{4\pi r c_0^2} e^{j(\omega_0 t - k_0 r)}$. This is not entirely dissimilar to the result of the one dimensional case, except for the $\frac{1}{r}$ factor that causes attenuation. It also must be noted that acoustic waves are not so different from electromagnetic waves from a mathematical point of view, as will become clear in the next section.

2.1.2. Acoustic Intensity

The time averaged sound intensity I , measured in $[W/m^2]$ is given in equation 2.23. The intensity is defined here for a pure sinusoidal signal with period $T = \frac{2\pi}{\omega_0}$. For the velocity \mathbf{u} we write $\mathbf{u} = u\hat{\mathbf{u}}$, where $\hat{\mathbf{u}}$ is a unit vector in the direction of propagation.

$$I = \frac{1}{T_0} \int_0^{T_0} p u dt \quad (2.23)$$

The velocity \mathbf{u} can be found by solving the linear Euler equation in three dimensions. For a plane wave, however, the relation between pressure and velocity is a fixed quantity: $p = \rho_0 c_0 u$. The factor of proportionality is known as the acoustic impedance z . This is analogous for the electrical impedance, which is a proportionality factor in the voltage-current relation. It turns out that both in case of a spherical or a plane harmonic wave, the intensity can be written as:

$$I = \frac{\Pi^2}{2\rho_0 c_0} \quad (2.24)$$

Π is the amplitude of the pressure signal. If the pressure is a function of frequency, we can use Parseval's identity to compute the intensity. $\Pi(\omega)$ is the Fourier transform of $p(x_0, t)$, where x_0 is the position where the intensity is measured.

$$I = \frac{1}{2\rho_0 c_0} \int_{-\infty}^{\infty} |\Pi(\omega)|^2 d\omega \quad (2.25)$$

It is commonplace to express the intensity on the decibel scale, as expressed in equation 2.26. p_f and I_f denote reference values. p is the RMS (Root Mean Square) pressure. The convention is to use $p_f = 20 \cdot 10^{-6}$ [Pa] as reference level. The corresponding reference intensity is $I_f = 10^{-12}$ $[W/m^2]$. If the aforementioned conventions are used, the intensity is referred to as the Sound Pressure Level (SPL).

$$SPL = 10 \log \left(\frac{I}{I_f} \right) = 20 \log \left(\frac{p}{p_f} \right) \quad (2.26)$$

2.2. Linear Loudspeaker

In this section, a practical implementation of a moving coil loudspeaker will be discussed. A schematic depiction of a loudspeaker diaphragm and voice coil motor is given in figure 2.2. The permanent magnets, combined with the top plate and the pole piece, generate a nearly uniform magnetic field that is orthogonal to the windings of the voice coil. The voice coil inside the magnetic field operates as a linear motor. This is similar to the ideal loudspeaker that was described in section 2.1.

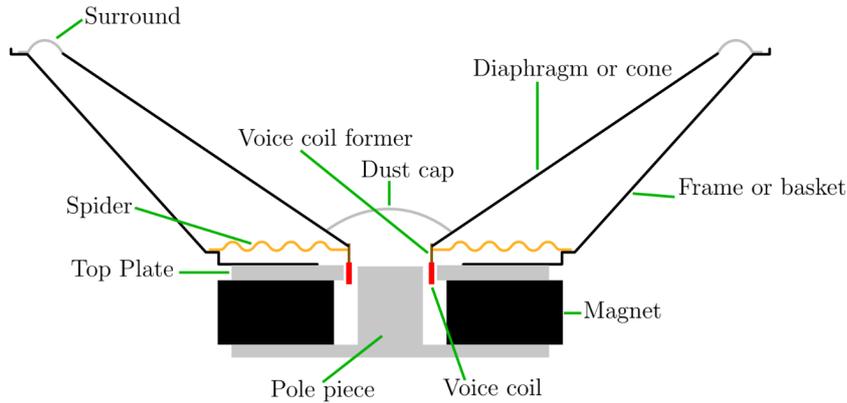


Figure 2.2: Schematic representation of a loudspeaker diaphragm.[10]

2.2.1. Mass-Spring-Damper Subsystem

The means of transforming electric power into sound is provided by the mechanical system, consisting of a spider, cone and surround. The system can be modelled as a combination of a mass, spring and damper. The mass is the mass of the cone, voice coil and half the mass of the spider and the surround, as the latter two are only partially in motion. The spring force is generated by the spider and the surround. The damping is a result of mechanical friction, the acoustic load and electrical damping. In practice only a small amount of the input power is converted into actual sound, which implies that the acoustic load is very small. The reactive load that is imposed on the system by acoustics is also small, as the mass of the air that is affected by the loudspeaker is small compared to the mass of the system. The effects of compression of air may be noticeable if the loudspeaker is mounted inside a closed cabinet. However, if the cabinet is sufficiently large and if the cone excursion is small, this parameter can be neglected.

The Thiele-Small Model, shown in figure 2.3, is often employed to represent the linear speaker by an equivalent electrical circuit. The components R_E and L_E represent the resistance and inductance of the voice coil, while R_M , C_M and L_M represent the mechanical mass-spring damper system of the speaker cone. The circuit depicted in figure 2.3 holds for a closed-box loudspeaker and it also includes the acoustic load, represented by R_A and C_A

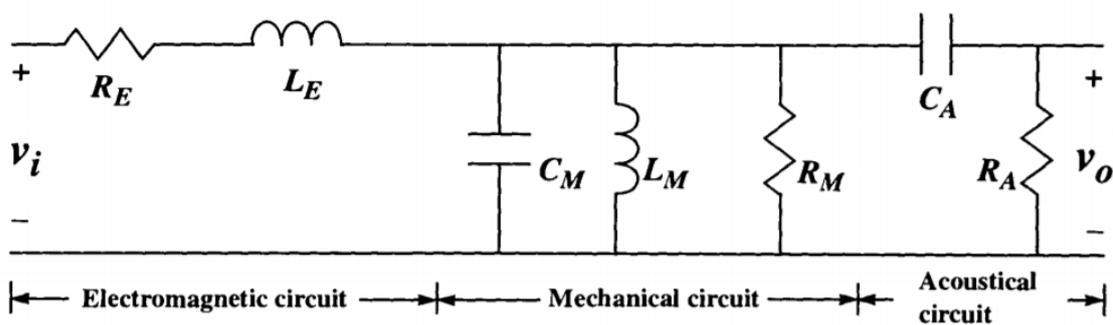


Figure 2.3: A simplified equivalent circuit of the loudspeaker, with acoustic load.[25]

2.3. Non-linear Loudspeaker

In section 2.2 the linear loudspeaker is discussed. This consideration is sufficient for modeling the behaviour of the loudspeaker when reproducing sound at low amplitudes. However, at higher amplitudes the electro-mechanical properties of the speaker driver as well as the physical limitations of the suspension and enclosure introduce non-linearities in the behaviour of the speaker, causing audible distortion of the reproduced sound. This non-linearities have a more prominent effect at low frequencies where higher am-

plitudes are required to produce sound.

2.3.1. Causes of Non-linearities in Loudspeakers

Klippel [18], Bai and Huang [22] give an overview of the main causes of non-linearities in loudspeaker systems. Below, a number of these phenomena are discussed, concentrating mainly on those which have a detrimental effect on the performance of low frequency actuators such as woofers.

Stiffness of Suspension K_m

The stiffness of the suspension of a loudspeaker K_m is related to the mechanical properties of the two suspension components of the speaker cone: the spider and surround (figure 2.1). For small displacement K_m is constant and the suspension of the cone can be modelled as an ideal spring. The restoration force F_k acting on the spring as a function of the displacement x of the cone from its equilibrium position is given by:

$$F_k = -K_m(x)x \quad (2.27)$$

As x becomes large K_m increases as a function of x . This introduces an additional x dependency in equation 2.27 which is multiplied with x making the F_k non-linear. The frequency dependency of the stiffness is linear. A related parameter to K_m is the compliance C_m , which is the inverse of the stiffness.

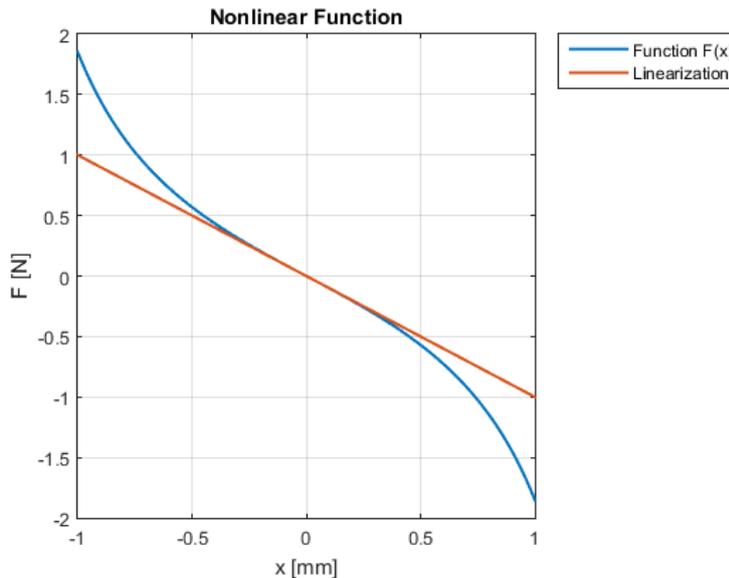


Figure 2.4: Nonlinear function with linearized graph through the origin. The nonlinear function is a primitive model for the restoration force, which is related to the suspension stiffness $K_m(x)$.

Force Factor Bl

Figure 2.5 shows the cross-section of a loudspeaker driver. A magnetic flux density B is present in the pole gap. According to the Lorentz force law, a current carrying wire will experience a force in the presence of a magnetic field which is given by

$$\mathbf{F}_L = i\mathbf{L} \times \mathbf{B} \quad (2.28)$$

where i is the current in the coil, B the magnetic field density and l the length of wire which lies inside the magnetic field. In the case of a loudspeaker, current flow is perpendicular to the direction of the magnetic

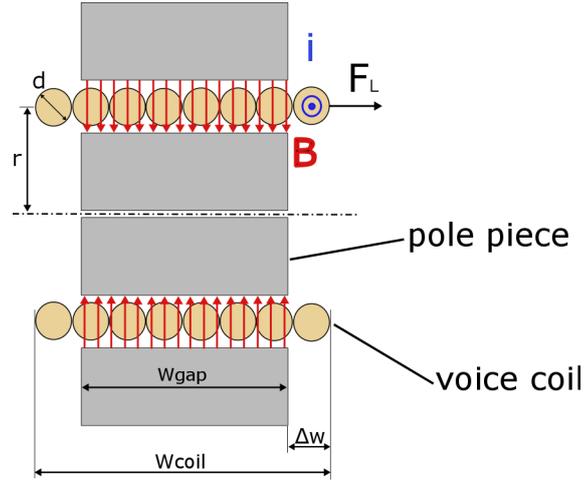


Figure 2.5: cross-section of a typical configuration of a loudspeaker driver's pole piece and voice coil

field. Furthermore, the effective wire length of the voice can be expressed as:

$$l_{eff} = 2\pi r n = 2\pi r \frac{w_{eff}}{d_w} \quad (2.29)$$

where r is the radius of the voice coil and n the number of coil windings inside the magnetic field. n is equal to the effective width of the voice coil w_{eff} (the width of the part of the coil which lies within the magnetic field of the air gap) divided by d_w , the thickness of the coil wire. The width of the voice coil is typically equal to or larger than than the width of the air gap. As long as the displacement of the coil is smaller than the difference between the widths $\Delta w = w_{coil} - w_{gap}$, F_L is a linear function of i . When the displacement is larger than Δw , the voice coil partially leaves the gap and w_{eff} (and consequently l_{eff} and F_L) decreases. The change in l_{eff} for displacements larger than $x > \Delta w$ is

$$dl_{eff} = \frac{2\pi r}{d_w} dx \quad (2.30)$$

Equation 2.30 holds if the magnetic field is homogeneous within the pole gap and vanishes outside it. In the more general and realistic case where this assumption does not hold, the effect of the voice coil leaving the pole gap is better described by the force factor $Bl(x)$, which is given by

$$Bl(x) = \int B(x) dl \quad (2.31)$$

The Lorentz force can also be expressed as

$$F_L = Bl(x)i \quad (2.32)$$

Because the force factor is a non-linear function of x , F_L is also a non-linear function of x and i .

Another effect of the displacement dependency of Bl is the a non-linear back EMF generated by the movement of the coil:

$$u_{emf} = Bl(x)v \quad (2.33)$$

where v is the velocity dx/dt of the coil and speaker cone.

Voice Coil Inductance L_e

The magnetic flux generated by current-carrying coil is given by the Maxwell-Faraday equation (2.34) as the surface integral of the magnetic induction.

$$\Phi = \int_{\Sigma} \mathbf{B}(\mathbf{r}, t) \cdot d\mathbf{A} \quad (2.34)$$

The magnetic induction itself depends on the permeability of the material which the magnetic field penetrates:

$$B = \mu H \quad (2.35)$$

where H is the magnetic field strength. the permeability of the air surrounding the magnetic core is the free space permeability μ_0 while the permeability of the metal core is $\mu_c > \mu_0$. The inductance of the coil is a function of Φ :

$$L_e = \frac{N}{I} \Phi \quad (2.36)$$

For positive displacement the coil moves away from the magnetic core and the magnetic field penetrates mainly the surrounding air, decreasing the flux and therefore also decreasing L_e . For negative displacement the magnetic field penetrates the steel surrounding the magnet (as well as the magnet) which has much higher permeability. This causes the Φ and L_e to increase. This makes L_e a non-linear function of the displacement.

The inductance also depends on the voice coil current. The total magnetic field strength H_{tot} consists of the magnetic field produced by the permanent magnet H_m and the field produced by the current in the coil H_c . A high positive current increases H_{tot} beyond the saturation point of the core and the permeability μ is decreased. A negative current decreases H_{tot} and μ is increased. This phenomenon introduces a non-linearity in the magnetic flux density:

$$B = \mu(i, x) H_{tot} \quad (2.37)$$

Other Causes of Non-Linearities

Other sources of non-linearities in loudspeakers include the Doppler effect, which is caused by the vibration of the loudspeaker box driven by low frequency speakers. The sound source has a constantly changing position with respect to the listener as result of this vibration, and for higher frequencies with a short wavelength, this can cause significant modulation in the perceived frequency. Additional nonlinear distortion can be introduced if the geometry of the speaker enclosure varies too much with respect to its dimensions as a result of varying air pressure within the enclosure. Defects in a speaker system such as loose components can also introduce parasitic, nonlinear oscillations.

2.3.2. Effects of Non-linearities in Loudspeakers

As detailed in the previous section, there are numerous contributing factors to the non-linear behaviour of a loudspeaker. According to [12], the most dominant nonlinear parameters of the lumped-parameter model presented in Section 2.3 are the compliance C_m , related to the nonlinear stiffness of the suspension K_m , the force factor Bl and the voice coil inductance L_e . All of the aforementioned nonlinear parameters depend on the cone excursion x while L_e additionally depends on the loudspeaker current i . The non-linearity of the parameters affects the loudspeaker's behaviour at different frequencies. It is implied in [12] that C_m is the dominant non-linear parameter at $f < f_s$, where f_s is the resonance frequency. Furthermore, at $f = f_s$, the force factor Bl is the dominant non-linearity, which is translated in strongly nonlinear R_M in Figure 2.3. Well above the resonance frequency, $f \gg f_s$, the dominant nonlinear parameter becomes L_e . In [18], a method is described that aims to quantify the sources of the nonlinear behaviour via a series of measurements. However, the (audible) result is similar in the above cases of distortion, as will be detailed in the subsequent sections.

Harmonic and Interharmonic Distortion

In order to illustrate the effect of non-linearities, the non-linear function of Figure 2.4 can be expanded using Taylor expansion. The resulting expression is given in Equation 2.38. The number of coefficients is in practice limited, because of the computational complexity. A third order polynomial is used by [1] for the suspension stiffness. It is stated by [14] that a Gaussian sum may be the preferred choice over polynomial expansion, because it is more accurate outside the initial range.

$$F(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 \dots \quad (2.38)$$

The offset term a_0 in the above equation is not of major concern, since it does not produce an audible frequency. Nevertheless, all terms except the a_1 term contribute to distortion. In order to understand the consequence of the higher order term, it is assumed that the nonlinear stiffness is an important contributor to distortion. As mentioned previously, this is the case below the resonance frequency. Thus, the output y of the system may be written as in Equation 2.39 as a function of the input current i .

$$y = y_0 + \alpha_1 i + \alpha_2 i^2 + \alpha_3 i^3 + \alpha_4 i^4 + \alpha_5 i^5 + \dots \quad (2.39)$$

It may be assumed that the input is a sinusoidal signal, e.g. $i = \cos \omega_0 t$. This assumption is very reasonable, since the input signal can be decomposed into an infinite set of sinusoids by the Fourier transform. The higher order terms α_n with $n \geq 2$ in equation 2.39 will now generate harmonic distortion (HD). This can be understood by the notion that $i^2 = \cos^2(\omega_0 t) = \frac{1}{2} + \frac{1}{2} \cos(2\omega_0 t)$. The second order term therefore yields a spectral component with twice the frequency of the input signal. Table 2.1 gives an explicit expression for the powers of the input signal for $i = \cos(\omega_0 t)$ and the spectral components that are introduced.

Table 2.1: Explicit expression for i^n for $i = \cos(\omega_0 t)$. The spectral components that are generated by the higher order terms are listed. The frequency is given as $f = \frac{\omega}{2\pi}$ and DC corresponds to a frequency $f = 0$

n	i^n	spectral components
1	$\cos(\omega_0 t)$	f_0
2	$\frac{1}{2} \cos(2\omega_0 t) + \frac{1}{2}$	DC, $2f_0$
3	$\frac{1}{4} \cos(3\omega_0 t) + \frac{3}{4} \cos(\omega_0 t)$	$f_0, 3f_0$
4	$\frac{1}{8} \cos(4\omega_0 t) + \frac{1}{2} \cos(2\omega_0 t) + \frac{3}{8}$	DC, $2f_0, 4f_0$
5	$\frac{1}{16} \cos(5\omega_0 t) + \frac{5}{16} \cos(3\omega_0 t) + \frac{5}{8} \cos(\omega_0 t)$	$f_0, 3f_0, 5f_0$

As indicated in Table 2.1, the higher order terms introduce frequencies that are an integer multiple of the original frequency. These frequencies are commonly referred to as harmonics, hence the name harmonic distortion. It is suggested by [14] that harmonic distortion does not sound so bad. Unfortunately, the nonlinear system introduces another type of distortion known as intermodulation distortion (IMD) when two or more frequencies are played simultaneously. Supposing the input now consists of two sinusoids with the same amplitude, but different frequency: $i = \cos(\omega_0 t) + \cos(\omega_1 t)$. The second order term now yields: $i^2 = \cos((\omega_0 + \omega_1)t) + \cos((\omega_0 - \omega_1)t) + \cos^2(\omega_0 t) + \cos^2(\omega_1 t)$. The cosine squared terms produce harmonic distortion as seen before. However, additional spectral components with frequencies $f_0 + f_1$ and $f_0 - f_1$ are created also. These are the intermodulation frequencies, which may be perceived as unpleasant according to [14]. Table 2.2 lists the additional spectral components that are introduced for all nonzero order terms.

Table 2.2: Inter harmonic spectral components that are introduced by the i^n term of the nonlinear transfer function, if the input is defined as: $i = \cos(\omega_0 t) + \cos(\omega_1 t)$.

n	spectral components
1	f_0, f_1
2	$f_0 + f_1, f_0 - f_1$
3	$2f_0 + f_1, 2f_0 - f_1, 2f_1 + f_0, 2f_1 - f_0$
4	$2f_0 + 2f_1, 2f_0 - 2f_1, 3f_0 + f_1, 3f_0 - f_1, 3f_1 + f_0, 3f_1 - f_0, f_0 + f_1, f_0 - f_1$
5	$4f_0 + f_1, 4f_0 - f_1, 4f_1 + f_0, 4f_1 - f_0, 3f_0 + 2f_1, 3f_0 - 2f_1, 3f_1 + 2f_0, 3f_1 - 2f_0, 2f_0 + f_1, 2f_0 - f_1, 2f_1 + f_0, 2f_1 - f_0$

Sixth or higher order terms in the nonlinear transfer function may generate additional spectral components, but usually these components are negligibly small. In [27], the higher order spectral components can be seen, but they are below the measurement uncertainty and may therefore be neglected. In the measurements of e.g. [12], the third harmonic component is the most dominant. This implies that the odd terms of Equation 2.39 contribute significantly to the non-linearity. Intuitively, this means that the nonlinear function is more or less odd symmetric. It is stated in [18] that asymmetrical non-linearities generate primarily even-order distortion. The graph in Figure 2.6 shows the frequency spectrum when distortion is introduced.

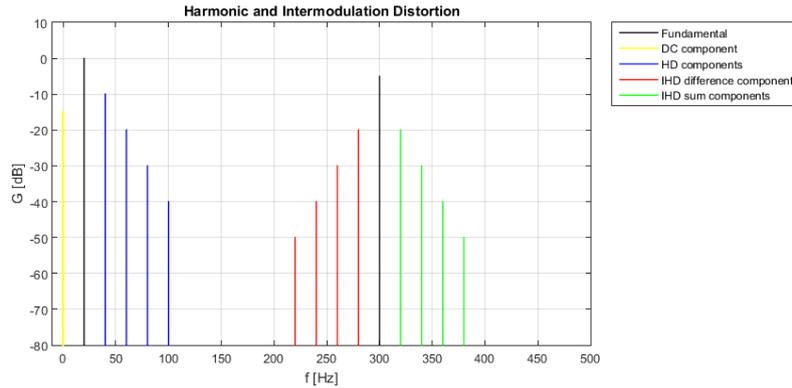


Figure 2.6: Frequency domain visualisation of harmonic (HD) and intermodulation distortion (IMD). Note that the frequency axis is linear. Additional spectral components may arise at higher frequencies, e.g. around 600 Hz but these are not indicated here.

Measure of Performance

As discussed in the previous section, the non-linear behaviour of a loudspeaker introduces additional spectral components to the reproduced signal, which is expressed as harmonic distortion. A quantitative measure of the amount of distortion is the Total Harmonic Distortion (THD). The expression for the THD is given by Equation 2.40, where A_n is the amplitude of the n th harmonic [2].

$$THD = \frac{\sqrt{\sum_{n=2}^{\infty} A_n^2}}{A_1} \quad (2.40)$$

The above definition for the THD is used by e.g. [18], but unfortunately it is not the only definition around in the field. A different definition used by e.g. [12] is given in Equation 2.41. In [1], both definitions are used and the definition of Equation 2.40 is considered favourable.

$$THD_R = \frac{\sqrt{\sum_{n=2}^{\infty} A_n^2}}{\sqrt{\sum_{n=1}^{\infty} A_n^2}} \quad (2.41)$$

Besides the harmonic distortion, there is intermodulation distortion. The Total Intermodulation Distortion (TIMD) is a measure of the performance of the system in this regard, similar to the THD. Equation 2.42 gives the expression for the TIMD as found in [18], where $Y(f)$ is the output signal.

$$TIMD = \frac{\sqrt{\sum_{n=-\infty}^{-1} |Y(nf_0 + f_1)|^2 + \sum_{n=1}^{\infty} |Y(nf_0 + f_1)|^2}}{Y(f_1)} \quad (2.42)$$

Referring back to figure 2.6, the TIMD is regarded as the ratio of the signal in the IMD difference and sum components (red and green graphs) and the fundamental component (300 Hz component). Note that not all distortion components are addressed in the TIMD quantification; components as $Y(2f_0 + 2f_1)$ are neglected. Unfortunately, there are again two slightly different definitions of the TIMD. The definition as used in e.g. [12] is given in Equation 2.43.

$$TIMD_R = \frac{\sqrt{\sum_{n=-\infty}^{-1} |Y(nf_0 + f_1)|^2 + \sum_{n=1}^{\infty} |Y(nf_0 + f_1)|^2}}{\sqrt{\sum_{n=-\infty}^{\infty} |Y(nf_0 + f_1)|^2}} \quad (2.43)$$

In the latter case, the $TIMD_R$ is the ratio between the intermodulation components adjacent to the fundamental frequency and the sum of the fundamental and the intermodulation components. It common practice to express the distortion as a percentage.

3

Non-linear Modelling of the Loudspeaker

In order to simulate the performance of the speaker, a non-linear state space model is derived and implemented in Simulink. The differential equations describing the derived speaker model are then solved numerically. The non-linear model is restricted to the three most prominent non-linearities: the force factor $Bl(x)$, the stiffness of the suspension $K_m(x)$ and the voice coil inductance $L_e(x)$. The non-linearities are determined empirically using a method described in chapter 4 and their fitted Taylor expansion approximations are implemented in the model.

3.1. Free-Body Analysis

In figure 3.1 a mechanical model of the loudspeaker is represented with a mass, spring and damper and representing the mass of the cone m , stiffness of the surround K_m and the acoustic and frictional losses b respectively.

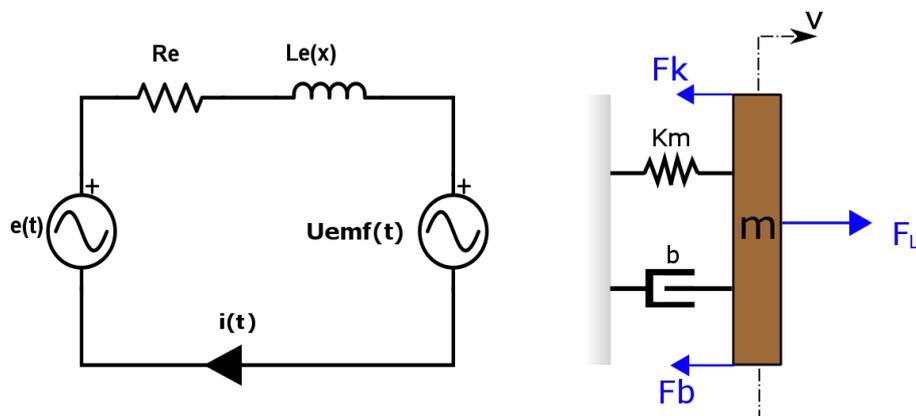


Figure 3.1: Electrical model of the motor (left) and mechanical model of speaker spring-mass-damper system (right)

Three forces are acting on the mass, the damping force $F_b = bv$, the restoration force $F_k = K_m x$, and the Lorentz force generated by the electric motor. The force balance of the system is given below:

$$ma(t) = F_L - F_k - F_b = Bl(x)i(t) - K_m(x)x(t) - bv(t) \quad (3.1)$$

The electric motor, shown on the left in figure 3.1 consists of a voltage source $e(t)$ and the coil resistance R_e and Inductance $L_e(x)$. The back EMF generated by the movement of the the coil is represented by a voltage source. Simple circuit analysis shows that:

$$e(t) = R_e i(t) + L_e \frac{dx(t)}{dt} \tag{3.2}$$

$Bl(x)$ describes the coupling between the electrical and mechanical components of the system. F_L is proportional to the current through the voice coil and u_{emf} is proportional to its velocity. This functionality can be modelled by a gyrator. The bond graph of the loudspeaker motor and spring-mass-damper system is given by causality in figure 3.2. The whole system can be modelled with the electrical equivalent circuit in figure 3.3.

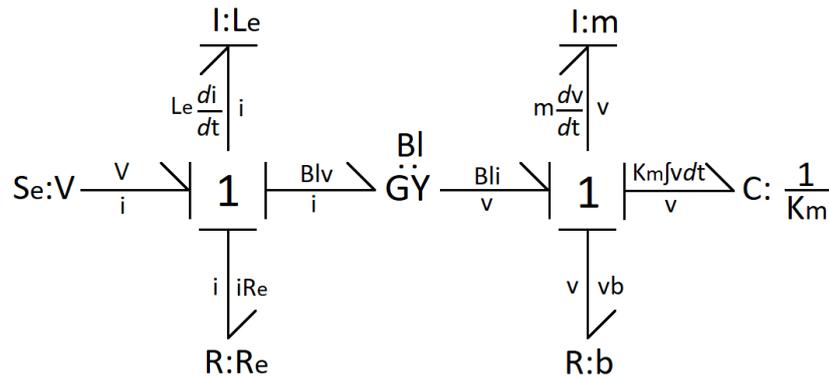


Figure 3.2: Bond Graph representation of the loudspeaker

3.2. State Space Model

For the state space model of the loudspeaker the equivalent electrical circuit in figure 3.3 is considered. This model is based on a model proposed by Klippel[18]. It has several advantages over the Thiele-Small model (figure 2.3):

- The electrical and mechanical forces and flows are clearly separated. On the electrical side (left) the driving force is the input voltage e and the flow is the input current i . On the mechanical side (right), the driving force is the Lorentz force F_L and the flow is the velocity of the moving speaker cone $v = dx/dt$
- Electrical components map directly to the mechanical properties of the driver; the mass m , damping factor b and Stiffness $K_m(x) = 1/C_m(x)$. The force factor $Bl(x)$ is represented by a Gyrator.

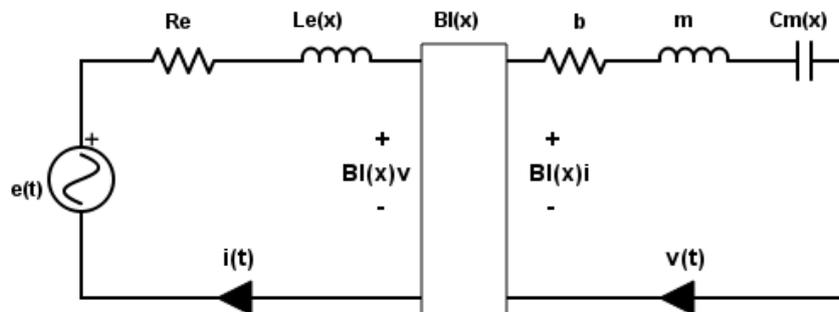


Figure 3.3: Electrical equivalent circuit of loudspeaker driver

Here, the acoustic load of the speaker is not modelled. This is because it is negligible, as explained in section 2.2.1, and because when using motional feedback the acceleration of the speaker cone will be measured and the acoustic load will not be taken into account. Other effects which are sometimes considered when modelling the speaker, like eddy currents losses of the magnet, have not been included for simplicity and because they have a more prominent effect at higher frequencies.

From the equivalent circuit in figure 3.3, several useful parameters can be extracted. The Lorentz Force $F_L = Bl(x)i(t)$ is given as a voltage induced on the mechanical side of the Gyrator. The back EMF generated by the moving coil $u_{emf} = Bl(x)v(t)$ is the voltage induced on the electrical side of the circuit. The acceleration $a(t)$ of the speaker cone, which as explained in section 2.2 is related to the acoustic pressure generated by the moving cone, is the derivative of the velocity of the cone. It is also equivalent and equal to the voltage across the inductor V_m divided by the mass m . This follows from Newton's second Law and the electro-mechanical equivalence of force and voltage and of mass and inductance [11]. The displacement x of the speaker cone is the integral of the velocity of the cone. The velocity and acceleration of the speaker cone will henceforth be referred to in terms of the displacement x as follows:

$$v(t) \equiv \frac{dx}{dt} \equiv \dot{x} \quad (3.3)$$

$$a(t) \equiv \frac{d^2x}{dt^2} \equiv \ddot{x} \quad (3.4)$$

The equations that describe the electrical and mechanical functionality of the speaker are:

$$e = R_e i + L_e(x) \frac{di}{dt} + Bl(x) \frac{dx}{dt} \quad (3.5)$$

$$Bl(x)i = m \frac{d^2x}{dt^2} + b \frac{dx}{dt} + K_m(x)x \quad (3.6)$$

Taking \dot{x} , x and i as state variables, the system is given in state space representation:

$$\begin{pmatrix} \ddot{x} \\ \dot{x} \\ i \end{pmatrix} = \begin{pmatrix} -\frac{b}{m} & -\frac{K_m(x)}{m} & \frac{Bl(x)}{m} \\ 1 & 0 & 0 \\ -\frac{Bl(x)}{L_e(x)} & 0 & -\frac{R_e}{L_e(x)} \end{pmatrix} \begin{pmatrix} \dot{x} \\ x \\ i \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{L_e(x)} \end{pmatrix} e \quad (3.7)$$

The non-linear parameters $Bl(x)$, $K_m(x)$ and $L_e(x)$ can be represented by their polynomial series expansions. This is a convenient way of dealing with them, as will become clear in chapter 5.

$$Bl(x) = \sum_{n=0}^{\infty} b_n x^n \quad (3.8)$$

$$K_m(x) = \sum_{n=0}^{\infty} k_n x^n \quad (3.9)$$

$$L_e(x) = \sum_{n=0}^{\infty} l_n x^n \quad (3.10)$$

To validate the model above, it is implemented in Simulink and solved numerically. The Simulink implementation of the model and the current results of model verification are discussed in chapter 5.

3.3. Possibilities in non-linear control

The possession of a non-linear model of the loudspeaker using the methods described in chapters 3 and 4 opens the possibility for improvement on the linear feedback controller designed by the digital and analog subgroups, making the system more stable and robust and potentially improving overall performance and suppression of linear and non-linear distortion.

As discussed in the thesis of the "digital" sub-group[4], the use of feed-forward when only a linear model of the loudspeaker is available does not offer any advantages over using feedback. This is because a controller based on a linear model cannot predict non-linear distortion. Having a non-linear model of the system allows for the use of pre-distortion to compensate for non-linearities. Using pure feed-forward when a non-linear model is available is however also likely to result in a worse controller since, in addition to lack of the inherent suppression of distortion offered by a feedback system, the limited accuracy of the model and its insensitivity to parameter variation can easily make the system unstable.

Other methods which do implement feedback and have been used in the literature include adaptive feed-forward[23] and observer based feedback[5][26]. These methods incorporate some sort of feedback, which improves the performance of the controller.

Depending on the complexity of the model, a delay due to the pre-processing of the input signal might be introduced in the system. Since delay in the reproduction of sound is typically not a big problem, especially for the reproduction of music in a home audio system, this is for most applications a desirable trade-off against the possibility of an unstable system. In some applications however, like in music production or the monitoring of a live music performance, a low latency is required. Therefore, the latency of the system must be kept as low as possible while not sacrificing on robustness and stability.

4

Impedance Measurement

As indicated in chapter 2, the loudspeaker is a non-linear device, which results in audible distortion. The motional feedback controller will be designed with the requirements regarding distortion in mind, i.e. the THD should not exceed 0.1% in the closed loop system. Therefore, the effect of the controller on the THD should be determined. The THD can be measured experimentally, but this means that many controllers should be made and tested. A more practical approach is to create a non-linear model of the loudspeaker and determine that effectiveness of various controllers via computer simulations. Note that it is not possible to determine the THD analytically, due to the nonlinear character of the system. The most common way to investigate the nonlinearities of a loudspeaker is by means of a Doppler laser system [27] [18] [6]. Such a system can measure position, velocity and acceleration of the diaphragm at various frequencies. The resulting data is sufficient to measure the distortion and the causes of the distortion in detail. However, even with highly precise equipment, it is difficult to accurately determine the nonlinearities [28]. In addition, the setup is limited by the poor signal to noise ratio. The method employed here is a variation on the commonly used small signal identification method, that is based on the impedance measurement. By measuring the impedance as a function of frequency, it is possible to identify the linear parameters related to the voice coil and the loudspeaker suspension and cone. This measurement can be repeated at various cone equilibrium positions to find the parameters as a function of position. The method for measuring the impedance as a function of the position is explained in section 4.1. The results are discussed in section 4.2 and section 4.3 contains a discussion of the results, including the downsides that are related to this method of nonlinear system identification.

4.1. Method for Impedance Measurement

The impedance of the loudspeaker was measured using the setup that is schematically depicted in figure 4.1, with the offset voltage source set to zero. The signal that is generated by the computer is passed on via the USB sound card to a series network of the loudspeaker and a resistor. The voltage at the loudspeaker terminals and the input voltage are both measured via the stereo input of the sound card. The unknown impedance Z_l can be computed using equation 4.1. The input voltage is V_i and the loudspeaker voltage is denoted by V_l . R_f is the series resistor, which is assumed to be accurately known.

$$Z_l = R_f \frac{V_l}{V_i - V_l} \quad (4.1)$$

The measurements were initially carried out using software that was used during the EPO-1 course of the Electric Engineering Bachelor [15]. The program in question is called LS_Measure and can be run using Matlab. The software was used with permission from the author. The test signals that are used by the software are pseudo-noise sequences. These are pseudo-random signals with a nearly flat spectrum, except for a DC error. Since the software was slightly limited in the capability of processing the data, a new Matlab program was written, which can be found in the appendix A.2. This code was used for the majority of the measurement data that is given in section 4.2.

4.1.1. Mass Measurement

The moving mass of the loudspeaker can be measured by measuring the impedance. At low frequencies, the effect of the voice coil self inductance is negligible and the loudspeaker is approximately a second order system. Furthermore, if the impedance is measured with a relatively low cone excursion, the system is approximately linear. The linearized phasor equation for this low frequency approximation is given by equation 4.2.

$$e = \frac{R_e K_m}{Bl} x + j\omega \left(Bl + \frac{R_e b}{Bl} \right) x + (j\omega)^2 \frac{R_e m}{Bl} x \quad (4.2)$$

Assuming that the system is underdamped, i.e. $\zeta < 1$, the homogeneous solution to this differential equation is: $x(t) = X e^{-\zeta \omega_n t} \sin(\omega_R \sqrt{1 - \zeta^2} t + \chi)$. The damping ratio can be computed by: $\zeta = \frac{b}{2\sqrt{mK_m}} + \frac{Bl^2}{2R_e \sqrt{mK_m}}$. The first term is caused by the mechanical damping and the second term is caused by electrical damping. The resonance frequency can be computed by equation 4.3.

$$\omega_R = \sqrt{\frac{K_m}{m}} \quad (4.3)$$

In the impedance measurements, the resonance peak is clearly visible both in the amplitude and phase spectra. The resonance peak occurs when the ratio between voltage and current reaches a local maximum. This is the case when the ratio between the Lorentz force F_L and the velocity $\frac{d}{dt}x$ is at an absolute minimum. And so, in the low frequency approximation, the impedance maximum can be found dividing both sides of equation 4.2 by $j\omega x$. The right hand side of the resulting expression can be differentiated with respect to ω . The roots of the final expression are $\pm\omega_R$, with ω_R the resonance frequency that is given in equation 4.3. It is clear that the resonance peak can be shifted by varying the mass or the stiffness. If the stiffness is kept constant, which is the case if the small signal assumption is valid, the mass of the cone can be measured by adding a small mass Δm . The original mass can now be expressed as a function of the added mass and the ratio of the resonance frequencies, as is the case in equation 4.4. The original resonance frequency is ω_R and with the inclusion of additional mass, the (shifted) resonance frequency is ω_Δ .

$$m = \frac{\Delta m}{\left(\frac{\omega_R}{\omega_\Delta}\right)^2 - 1} \quad (4.4)$$

4.1.2. Measurements with Offset

In order to obtain a nonlinear characterization of the loudspeaker, the small-signal model was established at various cone excursion positions. As described in section 2.3, the parameters that cause the strongest non-linear distortion are the force factor Bl , the suspension stiffness K_m and the voice coil self inductance L_e . The model that is employed is the same model as given in chapter 3. An offset is generated with respect to the position by applying an offset voltage to the loudspeaker terminals. On top of that, a small AC-signal is superimposed on the offset signal, which is used to measure the impedance. The setup that was used to measure the impedance with offset is given in figure 4.1.

The voltage signal can be written as $e = V_0 + \tilde{v}$, where V_0 is the voltage offset and \tilde{v} is the small signal voltage. It is assumed that \tilde{v} is of sufficiently low amplitude that the system can be considered locally linear. Consequently, the current can be written as: $i = I_0 + \tilde{i}$ and the position: $x = X_0 + \tilde{x}$. The position offset X_0 was measured with a caliper ruler. Obviously, the small signal position \tilde{x} cannot be measured without specialist equipment such as a Doppler laser. The differential equations 3.5 and 3.6 have been restated below with the aforementioned signals.

$$V_0 + \tilde{v} = R_e I_0 + R_e \tilde{i} + L_e(x) \frac{d}{dt} \tilde{i} + Bl(x) \frac{d}{dt} \tilde{x} \quad (4.5)$$

$$Bl(x) I_0 + Bl(x) \tilde{i} = K_m(x) X_0 + K_m(x) \tilde{x} + b \frac{d}{dt} \tilde{x} + m \frac{d^2}{dt^2} \tilde{x} \quad (4.6)$$

A typical mistake would be to replace the x -dependent variables, such as $L_e(x)$ by $L_e(X_0)$. This may be justified sometimes, but unless care is taken, it can give to the wrong answers. In our case, routinely filling

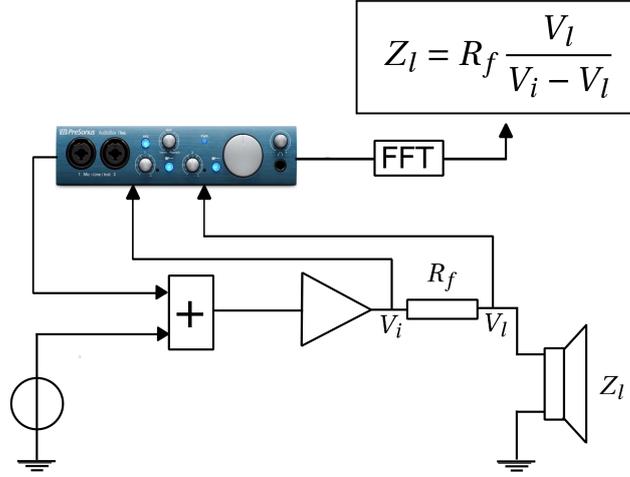


Figure 4.1: Diagram showing the setup used to for the impedance measurements with offset

in $Bl(X_0)$ and $K_m(X_0)$ actually leads to a faulty model, because the linearization is not correct and the Fourier transform (which will be used later) cannot be applied. A more rigorous approach is to linearize the nonlinear parameters in the point X_0 . The linearized functions are given below.

$$L_e(\tilde{x}) = L_e(X_0) + \frac{dL_e}{dx}(X_0)\tilde{x} \quad (4.7)$$

$$Bl(\tilde{x}) = Bl(X_0) + \frac{dBl}{dx}(X_0)\tilde{x} \quad (4.8)$$

$$K_m(\tilde{x}) = K_m(X_0) + \frac{dK_m}{dx}(X_0)\tilde{x} \quad (4.9)$$

At this point, it will be assumed that the small signals \tilde{v} , \tilde{i} and \tilde{x} can be written as $\tilde{v} = Ve^{j\omega t}$, $\tilde{i} = I(\omega)e^{j(\omega t + t(\omega))}$ and $\tilde{x} = X(\omega)e^{j(\omega t + \chi(\omega))}$ respectively. The amplitude V , I and X are assumed to be small. Consequently, any term which contains the product $\tilde{i}\tilde{x}$ or $\tilde{x}\tilde{x}$ can be safely neglected¹. The differential equations can now be fully linearized:

$$V_0 + \tilde{v} = I_0 R_e + R_e \tilde{i} + L_e(X_0) \frac{d}{dt} \tilde{i} + Bl(X_0) \frac{d}{dt} \tilde{x} \quad (4.10)$$

$$Bl(X_0)I_0 + I_0 \frac{dBl}{dx}(X_0)\tilde{x} + Bl(X_0)\tilde{i} = K_m(X_0)X_0 + X_0 \frac{dK_m}{dx}\tilde{x} + K_m(X_0)\tilde{x} + b \frac{d}{dt} \tilde{x} + m \frac{d^2}{dt^2} \tilde{x} \quad (4.11)$$

The equations 4.10 and 4.11 can be separated into the time dependent and time independent components. Supposing that the small signals \tilde{v} , \tilde{i} and \tilde{x} are set to zero initially, the remaining terms can be equated. The resulting expressions are given below:

$$V_0 = I_0 R_e \quad (4.12)$$

$$Bl(X_0)I_0 = K_m(X_0)X_0 \quad (4.13)$$

Neither of the above equations should come as a surprise; the former merely represents Ohm's law and the latter equates the time independent Lorentz force to the time independent spring force. The time independent terms will now be removed from the differential equations, which will be further analyzed in

¹The terms that would contain the product of \tilde{x} and \tilde{i} or \tilde{x} are also nonlinear

the phasor domain. For convenience, we write: $\Lambda(X_0, I_0) = X_0 \frac{dK_m}{dx} + K_m(X_0) + I_0 \frac{dBl}{dx}(X_0)$. The remaining equations are given in 4.14 and 4.15.

$$\tilde{v} = R_e \tilde{i} + j\omega L_e(X_0) \tilde{i} + j\omega Bl(X_0) \tilde{x} \quad (4.14)$$

$$Bl(X_0) \tilde{i} = \Lambda(X_0, I_0) \tilde{x} j\omega b \tilde{x} + m(j\omega)^2 \tilde{x} \quad (4.15)$$

The final equations are very similar to the linear differential equations, with the exception of the factor $\Lambda(X_0, I_0)$. The small signal impedance \tilde{z} will now be defined as $\tilde{z} = \frac{\tilde{v}}{\tilde{i}}$. Since both the small signal voltage and current can be measured, the small signal impedance can be computed also. The parameters R_e and $L_e(X_0)$ can be readily found by means of a least squares fit or by inspection of the data. Since the damping factor b is not considered one of the main nonlinearities, a single measurement is sufficient to find this parameter, e.g. using a least squares fit. For the remaining parameters, the case is, unfortunately, a bit more difficult. Application of a least squares algorithm is possible, but may be difficult due to the large number of parameters. There is, however, another method to determine Bl and K_m . In the low frequency range, the effect of L_e is very small, so it can be assumed that $L_e = 0$. This reduced system is just of second order, for which the resonance frequency is defined as:

$$\omega_R(X_0, I_0) = \sqrt{\frac{\Lambda(X_0, I_0)}{m}} \quad (4.16)$$

Since the mass can be measured rather accurately, this means that $\Lambda(X_0, I_0)$ can be measured as a function of the resonance frequency. As discussed in section 4.1.1, the resonance frequency can be found very easily in the measurement data. Solving for $\Lambda(X_0, I_0)$ gives: $\Lambda(X_0, I_0) = m\omega_R^2(X_0, I_0)$, so there arises a new differential equation:

$$m\omega_R^2(X_0, I_0) = X_0 \frac{dK_m}{dx} + K_m(X_0) + I_0 \frac{dBl}{dx}(X_0) \quad (4.17)$$

The equations 4.13 and 4.17 form a set of coupled differential equations. The solutions for $K_m(X_0)$ and $Bl(X_0)$ are: $Bl(X_0) = m\omega_R^2 \frac{dX_0}{dI_0}$ and $K_m(X_0) = m\omega_R^2 \frac{I_0}{X_0} \frac{dX_0}{dI_0}$. The mass m and the offset currents and positions I_0 and X_0 can be measured and the resonance frequencies $\omega_R(X_0)$ can be found by inspection of the impedance curves. Consequently, all information that is necessary to find the nonlinear force factor and stiffness can be readily found.

4.2. Results

The first impedance measurements were conducted using the LS_Measure Matlab tool. The results are given in figure 4.2a. It has to be noted, however, that the software filters the data so as to suppress irregularities in the data. This notwithstanding, the results are adequate for identifying the basics of the system.

The impedance curve that is shown in figure 4.2a is characteristic of a loudspeaker that is mounted in a bass reflex cabinet. Such a cabinet is not sealed, as is usually the case, but there is a vent at the back. The goal of this modification is to allow the acoustic radiation from the back of the cone to be added in phase with the primary source at the front. As such, there is better impedance matching between the mechanical domain and the acoustic domain. Kinsler [19] states that a bass-reflex system is more difficult to design, but capable of producing higher quality audio. The system can be modeled as an equivalent electrical circuit by the network in figure 4.3 [19].

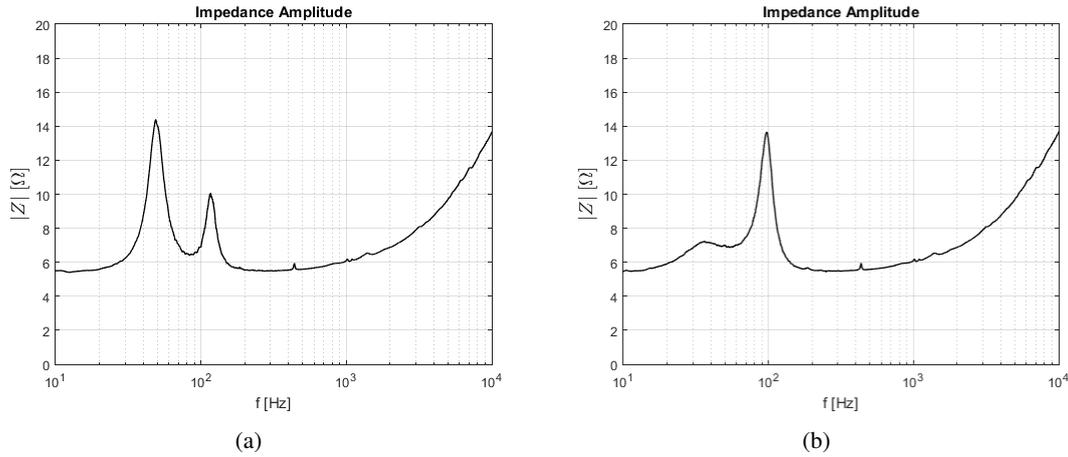


Figure 4.2: Impedance measurement with LS_Measure. (a) shows the initial data, revealing that the loudspeaker is mounted in a bass-reflex cabinet. The impedance measurement with closed vent is given in (b).

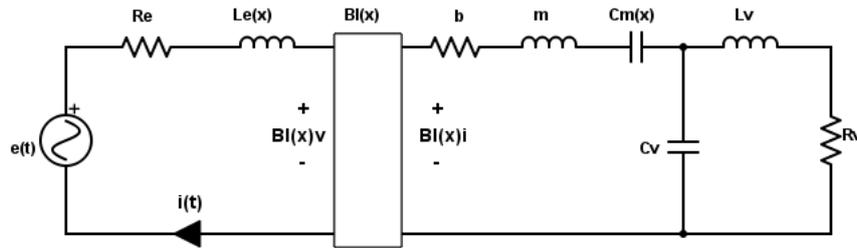


Figure 4.3: Electric equivalent circuit of a bass reflex loudspeaker system. The components L_v and R_v model the inductance and resistance of the loudspeaker vent and C_v models the compliance of the cabinet [19].

Upon sealing the vent of a bass reflex system, the inductance L_v becomes very large and the loudspeaker model reduces to the familiar closed cabinet model that is presented in chapter 3. The bass reflex model is a fifth order model, whereas the closed cabinet model represents a more manageable third order system. On top of that, the motion of the air inside the vent is difficult to measure and control, which means that motional feedback is more suitable for closed cabinet system. Hence, we decided to seal the vent and treat the system as a closed cabinet loudspeaker. The impedance of the closed system has been measured also and the result is given in figure 4.2b. The graph is very similar to the impedance of a closed cabinet loudspeaker, except for a small increase in impedance at 30 [Hz]. Apparently, there is still some radiation "leaking" via the vent, even though it was sealed. The impedance peak at 30 [Hz] is not a major concern, but would have been easier to apply a least squares fit to the data, if an actual sealed enclosure had been used in the first place. It might be possible to use the model of figure 4.3, but in practice it was difficult to apply a least squares fit to the data with seven or more variables. This is partially caused by the fact that there is a relatively small amount of data available at low frequencies and the high frequency data is a bit noisy. Therefore, the software may not find a local minimum near the physically sensible parameter values. Considering that bass reflex systems are not within the scope of this report anyway, the small resonance peak at 30 [Hz] will be ignored.

4.2.1. Mass Measurement Results

As discussed in section 4.1.1, the moving mass of the loudspeaker can be conveniently identified by measuring the impedance. The advantage of measuring the mass in this way is that the loudspeaker does not need to be disassembled. Furthermore, not all of the mass of the spider and surround is moving with the same velocity as the cone, so only a fraction of the respective masses should be added to the effective mov-

ing mass. This fraction should be approximately $\frac{1}{2}$, assuming that the average velocity of the suspension is half the velocity of the cone. However, if velocity of the suspension is not uniformly distributed, this rule of thumb does not apply. Fortunately an analysis of this problem is not necessary if the mass is analyzed using the impedance measurement. The masses that were added were either small batteries or permanent magnets. The permanent magnets were arranged such that the influence on the magnetic flux density of the loudspeaker itself is assumed to be negligibly small. It is recommended to use a non-magnetic mass to avoid complications altogether, but the mass needs to be accurately known and in our case, the mass of the magnets was nearly five times as precise as the mass of the batteries.

Mass Measurement with Batteries

Initially, the mass was measured with batteries of 2.8 [g] a piece. The accuracy of the mass was 0.2 [g]. The tape that was used to attach the mass to the cone of the loudspeaker was 0.4 [g] with 0.1 [g] accuracy. Thus, the total mass of one battery with tape is 3.2 [g] with an accuracy of 0.3 [g]. The mass of the loudspeaker cone can be computed using equation 4.4. Using this equation, the accuracy of the loudspeaker can be determined. Letting um and $u\Delta m$ denote the errors of the cone mass m and the added (battery) mass Δm , the accuracy of the mass is given by equation 4.18, where it is assumed that the errors with respect to the resonance frequencies ω_R and ω_Δ is small.

$$um = \frac{\partial \Delta m}{\partial m} u\Delta m = \frac{1}{\left(\frac{\omega_R}{\omega_\Delta}\right)^2 - 1} u\Delta m \quad (4.18)$$

Equation 4.18 reveals that the accuracy of the mass of the cone will be poor, unless the ratio of the two resonance frequencies is large. However, in order to achieve a high value for the ratio $\frac{\omega_R}{\omega_\Delta}$, the added mass should be large, which introduces additional uncertainty with respect to Δm . This illustrates the need for accurately defined mass elements, which is why the measurement was repeated with small magnets with a much more accurately defined mass. The results of the impedance measurements with the batteries is shown in figure 4.4.

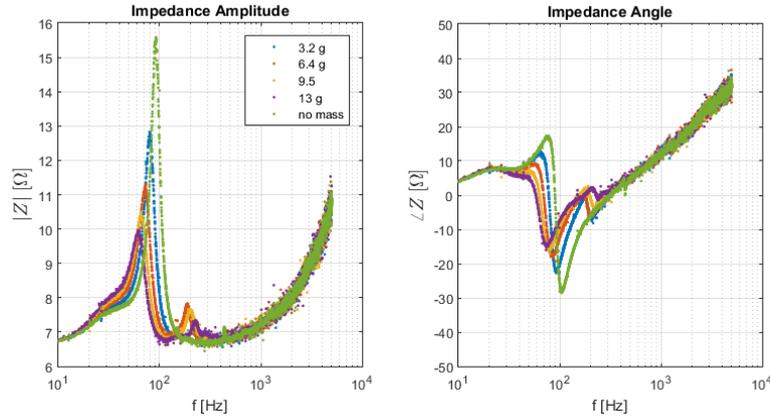


Figure 4.4: Impedance measurement of loudspeaker with added mass elements (small batteries). The impedance amplitude $|Z|$ is plotted on the left, with the phase $\angle Z$ on the right.

Upon closer inspection of the graphs in figure 4.4, it becomes apparent that there are two ways to determine the resonance frequency. Firstly, the peak of the amplitude spectrum occurs at the resonance frequency. Secondly, the phase of the impedance is zero at the resonance frequency, because the combination of the mass and compliance act like a "short circuit" at this frequency. In practice, however, these two methods give a slightly different result, due to the fact that in reality the voice coil reactance also affects the impedance. Since the phase is more affected by a small reactive component at the resonance frequency than the amplitude, it is more accurate to look at the peak in the amplitude spectrum, rather than the phase characteristics. The results are summarized in table 4.1.

Table 4.1: Resonance frequencies and corresponding original mass m as calculated with equation 4.4. The resonance frequency ω_R that corresponds to the situation where no mass is added, is 91 [Hz].

Δm [g]	ω_Δ [Hz]	m [g]
3.2	80	11
6.4	72	11
9.5	66	11
13	62	11

The measurements indicate that the moving mass of the loudspeaker is 11 [g]. As mentioned previously, the accuracy of this measurement is rather low, with an error of 1 [g] in the final value. If the error with respect to the frequency is also included the accuracy would be even lower, but this is not necessary since the accuracy with respect to the frequency can be increased easily by the use of longer pseudo-noise sequences, and by the use of a larger reference resistor. This will be discussed in more detail in section 4.3.

Mass Measurement with Magnets

A key limitation that was encountered when the mass of the loudspeaker was investigated was the poor accuracy of the reference mass. Fortunately, a more accurate reference mass was available to us: small magnets with a mass of 0.50 [g] each with an uncertainty of 0.02 [g]. As a result, the uncertainty of the mass is now primarily due to the uncertainty of the mass of the tape that was used to attach the masses. The same tape was used as in section 4.2.1. The results of the measurements are shown in figure 4.5.

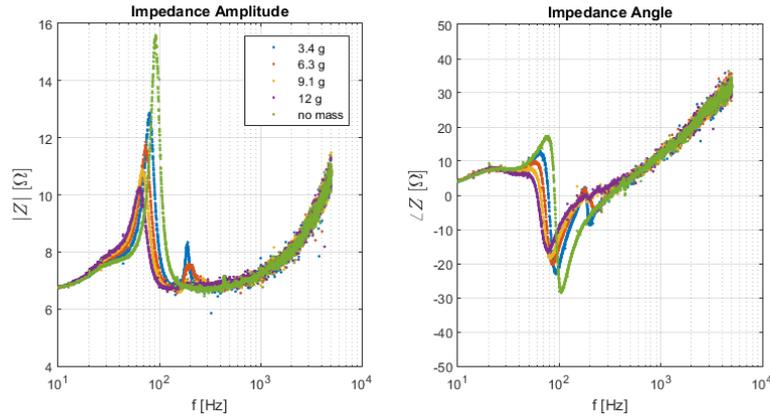


Figure 4.5: Impedance measurement of loudspeaker with added mass elements (small magnets). The impedance amplitude $|Z|$ is plotted on the left, with the phase $\angle Z$ on the right.

A potential downside that is related to the use of magnets as weight elements, is the fact that the magnets may disturb the magnetic field inside the loudspeaker. The magnets were arranged such that the effects were expected to be small. Nevertheless, for future work it is recommended to use accurately defined non-magnetic mass elements. The results are summarized in table 4.2.

Table 4.2: Resonance frequencies and corresponding original mass m as calculated with equation 4.4. The resonance frequency ω_R that corresponds to the situation where no mass is added, is 91.7 [Hz].

Δm [g]	ω_Δ [Hz]	m [g]
3.4	79.3	10.5
6.3	73.3	11.1
9.1	68.3	11.4
12	63.6	11.4

The measurements confirm the data from the previous section. However, the mass appears to be larger

when the added mass is increased. Since the lowest uncertainty was achieved in the measurement with 3.4 [g] extra mass, the adopted value for the loudspeaker mass is 10.5 ± 0.2 [g].

4.2.2. Results of Impedance Measurements with Offsets

In this section, the results of the measurements described in section 4.1.2 will be presented. A schematic depiction of the the setup is given in figure 4.1. The reference resistor has a resistance of $R_f = 1.09$ [Ω]. Table 4.3 lists the offset voltage V_0 that is added to the pseudo-noise sequence. The offset voltage that is measured across the loudspeaker terminals V_{ls} is also given, as well as the voltage across the reference resistor V_f .

Table 4.3: Input offset voltage V_0 with the corresponding loudspeaker offset voltage V_{ls} and the reference resistor offset V_f . The amplifier gain is roughly 20 [dBW].

V_0 [mV]	V_{ls} [V]	V_f [mV]
-1000	-9.7	-1290
-900	-8.6	-1250
-800	-7.6	-1120
-700	-6.6	-1000
-600	-5.7	-907
-500	-4.7	-706
-400	3.7	-635
-300	-2.8	-456
-200	-1.8	-345
-100	-0.89	-180
0	0	0
100	0.92	175
200	1.8	323
300	2.6	426
400	3.8	542
500	4.8	676
600	5.6	882

Measurements with more than 600 [mV] were performed, but did not result in meaningful data, most likely because the system is not stable with such a large positive offset. There were stability issues with the measurement with -1000 [mV] as well. The results of the measurements with -1000 [mV] and 600 [mV] offset, as well as the measurement with no offset are given in figure 4.6.

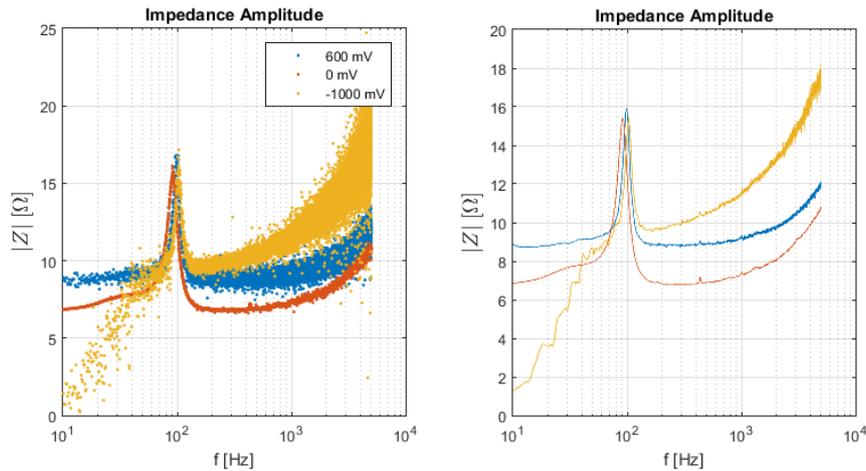


Figure 4.6: Impedance measurement of loudspeaker with several input offset voltages. The left-hand graph is the raw data. The right-hand graph shows the same data, but noise reduction has been applied.

Both of the graphs in figure 4.6 that were created when an offset voltage was applied, exhibit peculiar behaviour at low frequencies. For the -1000 [mV] measurement this seems to be caused by low-frequency instability of the measurement setup, because this type of behaviour occurs in all measurements with large negative offset. As for the 600 [mV] measurement, the high DC impedance cannot be explained. Obviously, the voice coil will heat up as a consequence of the power that is dissipated in the voice coil resistance. This will increase the voice coil resistance, but this alone cannot explain the data. Potentially, the system is not locally stable at this offset. For the -1000 [mV] measurement, the limit of stability seemed to be reached as well, since the amplifier was not able to sustain the offset voltage for a long enough time to complete the measurement. Another observation related to the -1000 [mV] data is that the impedance is increasing rather sharply in the $200 - 500$ [Hz] range, whereas the impedance curve without offset remains flat for much longer. This is not surprising, since the inductance is expected to increase for a negative offset as explained in section 2.3.1. However, at high frequencies, the inductance seems to decrease again. This is caused by the effects of Eddy currents [18]. Eddy currents are induced by a time varying magnetic fields in conductors. In this case the time varying magnetic field is generated by the AC-current in the voice coil and the conductor is the iron inside the loudspeaker. The flux that is generated by the Eddy currents opposes the incoming flux from the voice coil. It is demonstrated in [9] that this phenomenon results in an (nonlinear) impedance that depends on the square root of the frequency. Fortunately, according to Klippel [18], it is adequate to model the Eddy current losses by the circuit shown in figure 4.7.

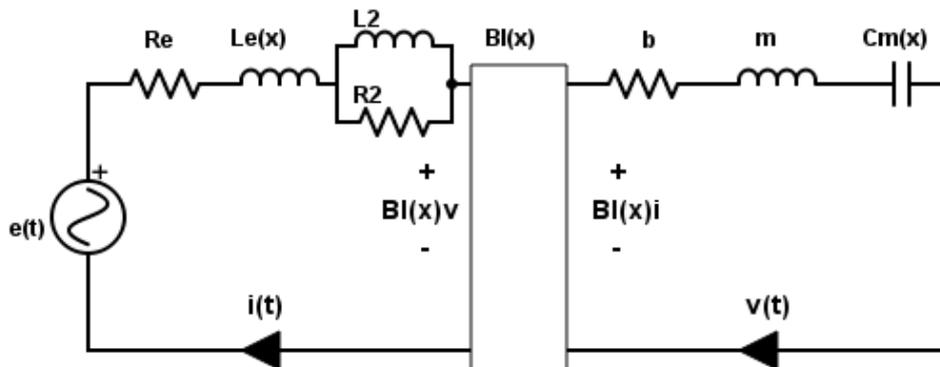


Figure 4.7: Electrical equivalent model of the loudspeaker which takes eddy current losses into account.

The components L_2 and R_2 do not have a physical meaning, and their values depend on the frequency range wherein the impedance is measured or fitted. These components are commonly assumed to be linear [18] [12]. In order to make this assumption valid for our loudspeaker, the practical operation range of the loudspeaker has to be limited, as will be discussed shortly. The graphs of the remaining measurements are given in figure 4.8.

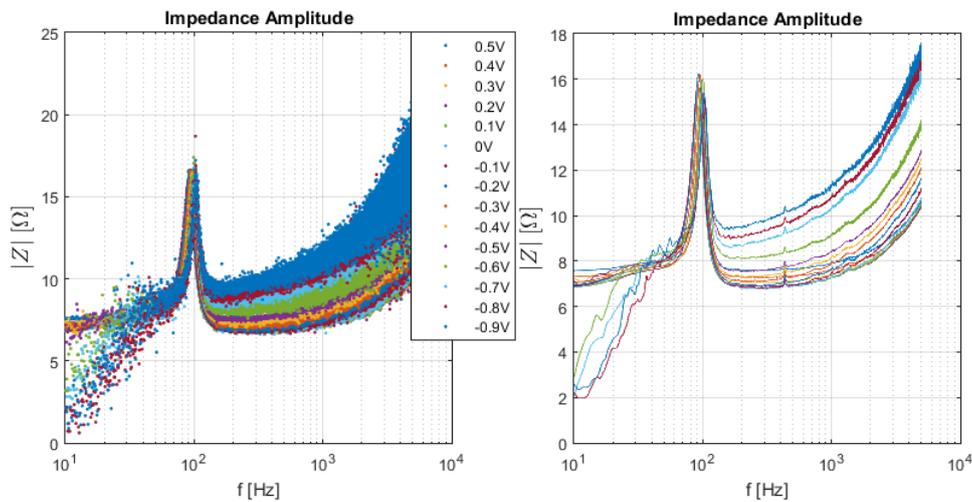


Figure 4.8: Measurements with offset that provide useful data. The left-hand graph is the raw data. The right-hand graph shows the same data, but noise reduction has been applied.

The graphs of figure 4.8 indicate that the effect of Eddy currents is very limited for offset voltages of 500 [mV] or less. The low frequency stability is also good if the offset is limited to 500 [mV] or less. It seems, therefore, that the practical operation range is limited to the cone excursions that are related to these voltages. It turns out that the operation range is -2.5 [mm] to 2.0 [mm] if the effects of Eddy currents are neglected. The most important information that is used from the graphs of the impedance in figure 4.8 is the resonance frequency. A least squares fit has been performed also to find the value of L_e as function of the position. The offset voltage results in an offset in position, that was measured with a caliper ruler. Also, the current can be determined with the data from table 4.3. The data is given in table 4.4. This is the data that will be used in chapter 5 for establishing a nonlinear model of the loudspeaker.

Table 4.4: Measurements of offset position, current and resonance frequency f_R . The voice coil inductance L_e was measured using a least squares fit.

X_0 [mm]	I_0 [mA]	f_R [Hz]	L_e [μ H]
-3.6	-1150	102.9	453
-3.4	-1030	102.7	445
-3.2	-919	101.8	431
-2.9	-833	100	356
-2.5	-648	98.5	323
-2.1	-583	96.5	315
-1.8	-419	95.3	305
-1.3	-317	92.8	291
-0.6	-165	91.0	275
0	0	90.9	260
0.8	161	92.6	250
1.3	297	93.7	243
1.6	392	95.2	237
1.7	498	96.9	234
2.0	621	97.8	232

4.3. Discussion of Impedance Measurements

The concept of identifying the nonlinearities of a loudspeaker by means of impedance measurements with offsets is never mentioned in literature. Having used this concept ourselves, it becomes clear that this method has a number of intrinsic limitations. Firstly, the surround of a loudspeaker cone is often made of material with visco-elastic properties [16]. One of the consequences of this is that when a DC voltage is applied, the cone will quickly assume an offset position, but then "creep" a little further. This is known as the creep effect, which is completely ignored by the method employed in this report. Other effects of the visco-elasticity of the material include frequency dependent damping and compliance [17]. Secondly, the position measurement is rather inaccurate. We used a caliper ruler that can measure the position up to 0.1 [mm] accuracy, which is quite poor considering that the diaphragm excursion is only a few millimeters. Thirdly, the linearization of the nonlinear model turned out to be more complicated than expected and there are still (nonlinear) differential equations that have to be solved if the large signal parameters are to be recovered. This also means that extending the model is difficult, since the linearization process needs to be repeated and a new set of differential equations results. Fourthly, the voice coil resistance will dissipate power if there is a large DC voltage across the loudspeaker terminals. The heat that is generated will increase the voice coil resistance. This increase can be by as much as 15%. The current will therefore drop and the offset position will move slightly towards the equilibrium. The heat development is a key limitation to identifying the loudspeaker parameters. Normally, the loudspeaker is driven by a time-varying signal, which means that maximum cone excursion can be higher for the same amount of power that is dissipated compared to the case with a DC offset signal. And finally, the current, voice coil inductance and resonance frequency need to be characterized as function of position in order to identify the main nonlinear parameters that were discussed in section 2.3. It is desirable that these parameters are expressed as a polynomial of reasonably high order, because a low order polynomial can only produce the low order distortion. As mentioned in section 2.3, fifth order harmonics are often measurable in the output, which means that the current and resonance frequency need to be identified as a sixth order polynomial. In order for this to be possible, there need to be much more data points than currently available, otherwise that results from fitting the model is very large. The measurements already take rather long, so in order for this method to be plausible, the measurements alone could take several days. Nevertheless, the method can be applied to a loudspeaker with low creep effect. For future work however, it is recommended that for the impedance measurement, the loudspeaker will be driven by a current rather than a voltage. In this way, the heat that may be developing will not reduce the current and the position will be more stable. Furthermore, it is recommended that a larger reference resistor is used. The resistor needs to be able to dissipate the extra power and the same applies to the amplifier. The voltage across the resistor will, however, be higher, which

results in a better signal-to-noise ratio. Increasing the total voltage has the same effect, but this means the loudspeaker voltage is higher and therefore, the small signal approximation is not valid.

5

Model Verification

The results of the impedance measurements as summarized in table 4.4 can be used to establish a model of the loudspeaker with nonlinear force factor, voice coil inductance and compliance. A least squares fit can be applied to the data to find $I_0(X_0)$, $\omega_R^2(X_0)$ and $L_e(X_0)$. The latter function represents one of the three dominant nonlinearities in loudspeakers. The other functions will be used to find Bl and K_m . In section 5.1, a method will be presented to find the parameters of interest based on the known functions $I_0(X_0)$ and $\omega_R^2(X_0)$. Section 5.2 describes the method and results of the simulation of distortion with the nonlinear model. It must be noted, however, that the model that was used for the simulations was not correct. The linearization of the equations 4.5 and 4.6 had not been explicitly linearized. The resulting model considers the small signal behaviour at various positions, but cannot accurately predict the large signal behaviour of Bl and K_m . At the time of writing, the proper method for determining the nonlinearities has not been applied to the data yet. The distortion of the loudspeaker has also been measured experimentally. The results from the distortion simulation using the model and the measurements will be compared.

5.1. Solving for $K_m(x)$ and $Bl(x)$

For convenience, the offset positions and currents X_0 and I_0 , will be replaced by the regular position x and current i , now that we are dealing with the large signal domain only. Furthermore, it is assumed that for every I_0 , there is a well defined X_0 , which implies that the current can be written as a function of position: $i = i(x)$. The two equations that were found for $K_m(x)$ and $Bl(x)$ in the previous chapter are repeated in equations 5.1 and 5.2.

$$Bl(x)i = Km(x)x \quad (5.1)$$

$$m\omega_R^2(x) = x \frac{dKm}{dx} + Km(x) + i \frac{dBl}{dx}(x) \quad (5.2)$$

Fortunately the coupled equations can be solved quite easily, resulting in: $Bl(x) = m\omega_R^2 \frac{dx}{di}$ and $K_m(x) = m\omega_R^2 \frac{i}{x} \frac{dx}{di}$. The functions $i(x)$, $m\omega_R^2(x)$ and $mi(x)\omega_R(x)$ are not stated explicitly, but we can use Taylor's theorem to express these functions as polynomials. For the current, we find thus: $i(x) = i_0 + i_1x + i_2x^2 + \dots$ and for the derivative of the current: $\frac{di}{dx} = i_1 + 2i_2x + 3i_3x^2 + \dots$. For the function $m\omega_R^2(x)$ we will write $m\omega_R^2(x) = \lambda_0 + \lambda_1x + \lambda_2x^2 + \dots$ and for the product of i and $m\omega_R^2$ we have $im\omega_R^2 = \phi_0 + \phi_1x + \phi_2x^2 + \dots$. $Bl(x)$ and $K_m(x)$ can be solved directly, if they are also expanded as polynomials. For $Bl(x)$ we have $Bl(x) = Bl_0 + Bl_1x + Bl_2x^2 + \dots$ and in similar fashion we obtain $K_m(x) = K_0 + K_1x + k_2x^2 + \dots$. The solutions for the coefficients is given in equations 5.3 and 5.4.

$$Bl_n = \frac{1}{i_1} \left(\lambda_n - \sum_{v=2}^{n+1} vK_{n-v+1}i_v \right) \quad (5.3)$$

$$K_n = \frac{1}{i_1} \left(\phi_{n+1} - \sum_{v=2}^{n+1} v K_{n-v+1} i_v \right) \quad (5.4)$$

Actually, for K_0 and Bl_0 the above expressions are not valid, since the summation expression cannot be computed, but these coefficients can be computed by: $K_0 = \frac{\phi_1}{i_1}$ and $Bl_0 = \frac{\lambda_0}{i_1}$. Since $L_e(x)$ can be measured directly, as explained in section 4.1.2, the coefficients for a polynomial of L_e can be directly evaluated with a least squares fit.

5.2. Simulating Large Signal Behaviour of the Loudspeaker

The block diagram of the state-space model above is given in figure 5.1, as implemented in Simulink. The nonlinearities are implemented using lookup tables of the fitted polynomial series expansions. The constant parameters are given as gains. The parameters (shown in red) are determined using the method explained in chapter 4. The MATLAB code used to generate the values for these parameters can be found in appendix A.3. Using Simulink, the non-linear behaviour of the speaker for a given input signal is solved numerically and simulated and the input current $i(t)$, acceleration $a(t)$, velocity $v(t)$ and displacement $x(t)$ are monitored. The simulation runs with a fixed step size of $1/48000$ s.

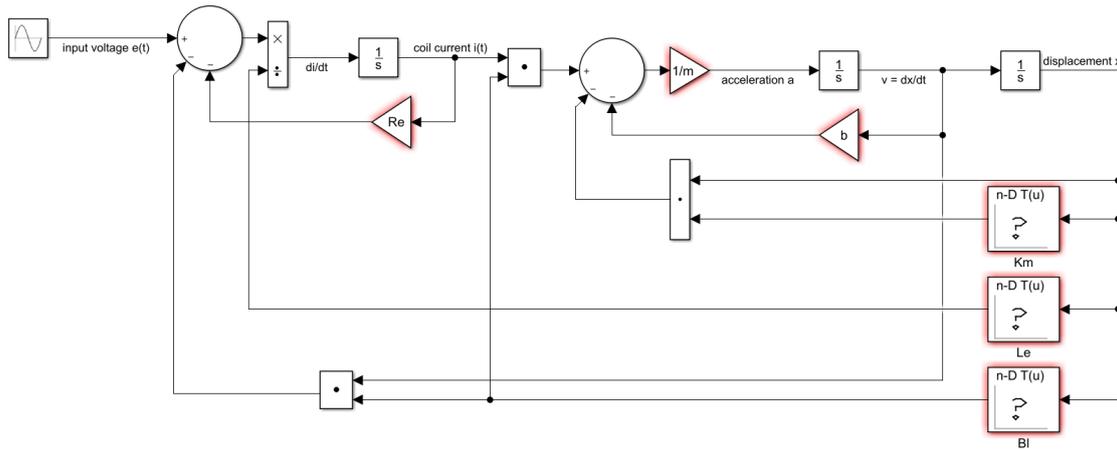


Figure 5.1: State space model of loudspeaker implemented in Simulink

To validate the model, THD measurements are simulated at different amplitudes of the input signal and compared to the measurements using the physical speaker. The simulated measurements must comply with the requirements given in section 1.1. The THD measurement code can be found in Appendix A.1. MATLAB code to measure TIMD has also been developed but not used.

5.3. Discussion of Simulation Results

Figure 5.2 shows the characteristic curves of the non-linearities $L_e(x)$, $Bl(x)$ and $K_m(x)$ as calculated initially. At first consideration, they seem plausible. The form of the curves is consistent with the literature[18][8] and so is the order of magnitude of the calculated values with $L_e(0) = 260\mu H$, $Bl(0) = 3.16 \frac{N}{A}$ and $K_m(0) = 3.59 \frac{N}{mm}$. Klippel [18] explains that this form of the force factor curve suggests an equal-length configuration of the voice coil. This means that the width of the voice coil (w_{coil} in figure 2.5) is the same as that of the pole gap, making the $Bl(x)$ non-linear even at small displacements.

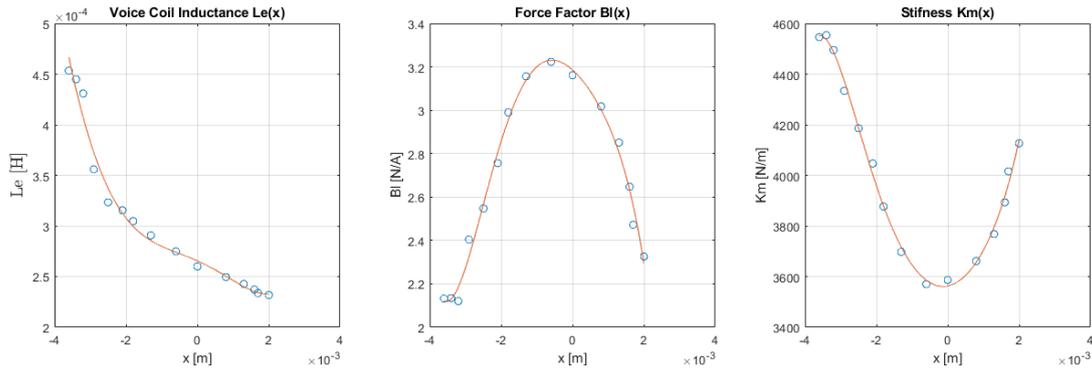


Figure 5.2: The calculated characteristic curves of the non-linearities $L_e(x)$, $Bl(x)$ and $K_m(x)$. The blue circles represent the values found at the corresponding x -offset and the red line represents their 5-th order polynomial fit.

However, these values are not correct. The values of the non-linearities at each x -offset are acquired from impedance measurements using the small signal identification method and are, as discussed in 4, not suitable for predicting large signal behaviour of the speaker. This can also be seen when comparing the Total Harmonic Distortion measured with the physical speaker with the simulated Total Harmonic Distortion measurement using the model (figure 5.3). The figure shows that for all frequencies between 20-180 Hz, the simulated THD is much lower than in the speaker measurements. The inaccuracy of the results is also likely due to the simplicity of the model. Effects like eddy currents losses have much more influence on the behaviour of the loudspeaker than what was initially expected (see section 4.3), and including them in the model would significantly improve the accuracy of the model.

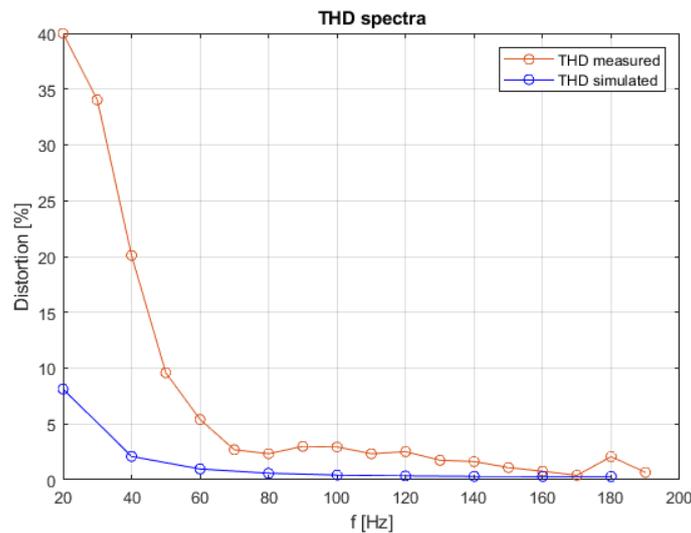


Figure 5.3: Comparison of THD measurement from the speaker using a microphone and the simulated THD using the Simulink model an input signal amplitude of 5V. The measurements were made using the matlab codes 'plotTHD.m' and 'simTHD.m' (see Appendix A.1)

At the time of submission of this thesis, validation of the model has not given good results. This is due to a mistake in the methodology used for the interpretation of the impedance measurement results, resulting in characteristics for the non-linearities $L_e(x)$, $K_m(x)$ and $Bl(x)$ which seem plausible but are incorrect. This leads to an underestimation of the THD in the model compared with THD on the physical speaker for the same input amplitude. A new methodology for the extraction of non-linearity characteristics from impedance measurements is currently being investigated. This methodology is explained in detail in section 4.1.2.

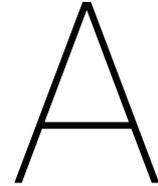
6

Conclusion

Non-linear behaviour of a loudspeaker was investigated and a non-linear model was derived and solved numerically in Simulink. MATLAB code used for the parameterization and validation of the the loudspeakers and MFB system's performance was provided. Initial validation of the model showed underestimation of the Total Harmonic Distortion generated by the speaker, caused by an error in the methodology used for the extraction of speaker parameters from impedance measurements. The nonlinear system identification based on impedance measurement suffers from poor accuracy. The proposed measurement technique is also very time-consuming. A new methodology was proposed for the estimation of the non-linear parameters. A lack of time has, however, prevented the application and validation of this method.

6.1. Recommendations

For future work we recommend that a nonlinear model is created using the Klippel method [18], with a doppler laser. The effects of Eddy currents were visible in the measurement, but were not taken into consideration in the data processing in this report. It is recommended that this phenomenon is also assessed. Further research is needed to identify the benefits of various types of controllers. It is also recommended that the possibility of a current driven system is investigated.



MATLAB code

A.1. Model Validation

A.1.1. plotTHD.m

```
breaklines
1 %=====
2 % plotTHD.m :   Performs nonlinear distortion measurements on loudspeaker
3 %               using USB soundcard. Uses the pawavplayw file from author
4 %               Gerard Janssen .
5 %-----
6 % Author      :   Aart-Peter Schipper & Alexandros Skourtis-Cabrera
7 % Date        :   17/6/2018
8 %=====
9 function plotTHD()
10 % set basic measurement paramters
11 Fs = 48000;
12 N = 5*Fs;
13 df = 10;
14 fMin = 20;
15 fMax = 200;
16 margin = 50;
17 duration_sec = N/Fs;
18 M = 5;
19
20 % obtain impulse response H
21
22
23 % execute the THD function to find the THD
24 A = 1.0;
25 [f0 , THDa, THD] = fnTHD(Fs , N, df , fMin , fMax , A, margin);
26
27 % use the plotTHD function to generate a plot of the result
28 THDa = 100*THDa;
29 THD = 100*THD;
30 plotTHD(f0 , THDa, THD);
31
32 %-----
33 % list of nested functions :
34 %=====
35 % function THD computes the THD spectrum with resolution df
36 % | list of inputs:      || definition:                                     |
```

```

37 % |-----||-----|
38 % | Fs           || sample frequency           |
39 % | N           || number of samples           |
40 % | df         || THD resolution           |
41 % | fMin       || lower bound of frequency sweep |
42 % | fMax       || upper bound of frequency sweep |
43 % | A          || amplitude of input signal      |
44 % | H          || impulse response of loudspeaker |
45 % | margin     || look for maximum between +- margin |
46 %
47 % | list of outputs || definition:           |
48 % |-----||-----|
49 % | f0         || distorted frequencies     |
50 % | THD        || Total Harmonic Distortion |
51 % | THDa       || accoustic THD           |
52 %=====
53 function [f0, THDa, THD] = fnTHD(Fs, N, df, fMin, fMax, A, margin)
54 % find number of loop iterations
55 K = floor((fMax -fMin)/df);
56
57 % preallocate for speed
58 f0 = zeros(K, 1);
59 THDa = zeros(K, 1);
60 THD = zeros(K, 1);
61
62 for k =0:K-1
63     f0(k+1) = fMin + k *df;
64     x = gen_freq(f0(k+1), A, Fs, duration_sec, N); % play f0
65     y = record_freq(x, Fs);% obtain data of recording and fourier transform
66
67     spectrum = fftshift(fft(y(:,2)'));
68 % %         spectrum = spectrum /H;
69
70 % sort out the frequency axis
71 if ceil(N/2) > N/2
72     f = 1/(2*N) * [0 : 2 : N-1];
73     spectrum = spectrum(N/2:end);
74 else
75     f = 1/(2*N) * [0 : 2 : N-1];
76     spectrum = spectrum((N/2+1):end);
77 end
78
79 F = f *Fs;
80 %     figure
81 %     plot(F, abs(spectrum))
82 %     xlim ([0 500])
83 %     hold on;
84
85
86 % select data point where distortion is expected
87 fDistorted = zeros(M, 1);
88 nDistorted = zeros(M, 1);
89 peak = zeros(M,1);
90
91 % find the amplitude of the maximum distortion component
92 for m = 1:M
93     fDistorted(m) = m *f0(k+1);
94     nDistorted(m) = floor((N/Fs) *fDistorted(m));

```

```

95
96         %plot (nDistorted(m)-margin:nDistorted(m)+margin , spectrum (nDistorted(m)-margin:
97
98         selection = abs(spectrum (nDistorted(m)-margin:nDistorted(m)+margin));
99         %plot((nDistorted(m)-margin:nDistorted(m)+margin)./(N/Fs) , selection)
100         peak(m) = max(selection);
101         %disp(peak(m))
102     end
103
104 %         stem(fDistorted , peak);
105
106         distortion = sum(peak(2:M).^2);
107         THDa(k+1) = sqrt(distortion/(distortion +peak(1)^2));
108         THD(k+1) = sqrt(distortion)/peak(1);
109
110     end
111 end
112
113 %=====
114 % function plotTHD plots the THD in percent as function of frequency
115 % | list of inputs:      || definition:
116 % |-----||-----|
117 % | f0                  || distorted frequencies
118 % | THDa                || acoustic THD
119 % | THD                 || standard THD
120 %
121 % | list of outputs:
122 % |-----|
123 % | graph of THDa and THD versus f0
124 %=====
125 function plotTHD(f0 , THDa, THD)
126     figure
127     plot (f0 , THDa, 'o')
128     hold on
129     plot (f0 , THD, 'o')
130     hold off
131     title ('THD spectra')
132     xlabel ('f [Hz]')
133     ylabel ('Distortion [%]')
134     grid on
135     legend ('THDa', 'THD')
136 end
137 end

```

A.1.2. simTHD.m

```

breaklines
1 %=====
2 % simTHD.m :   Measures THD by numerically solving the theoretical model
3 %             of the loudspeaker in Simulink. Works similarly to plotTHD.
4 %-----
5 % Author      :   Aart-Peter Schipper & Alexandros Skourtis-Cabrera
6 % Date        :   17/6/2018
7 %=====
8
9
10
11 % set basic measurement paramters

```

```

12 Fs = 48000;
13 N = 5*Fs;
14 df = 20;
15 fMin = 20;
16 fMax = 200;
17 margin = 50;
18 duration_sec = N/Fs;
19 M = 5;
20 source = 1;           % 0 for soundcard, 1 for Simulink model
21
22
23
24
25 % obtain impulse response H
26
27
28 % execute the THD function to find the THD
29 A = 5.0; % Amplitude of sinewave (in Volts if simulating)
30
31 %-----
32 % list of nested functions:
33 %=====
34 % function THD computes the THD spectrum with resolution df
35 % | list of inputs:      || definition:                |
36 % |-----||-----|
37 % | Fs                   || sample frequency          |
38 % | N                     || number of samples        |
39 % | df                    || THD resolution           |
40 % | fMin                  || lower bound of frequency sweep |
41 % | fMax                  || upper bound of frequency sweep |
42 % | A                     || amplitude of input signal  |
43 % | H                     || impulse response of loudspeaker |
44 % | margin                || look for maximum between +- margin |
45 % | source                || 0 for soundcard, 1 for Simulink model |
46 %
47 % | list of outputs      || definition:                |
48 % |-----||-----|
49 % | f0                   || distorted frequencies      |
50 % | THD                  || Total Harmonic Distortion  |
51 % | THDa                 || acoustic THD               |
52 %=====
53
54 % find number of loop iterations
55 K = floor((fMax -fMin)/df);
56
57 % preallocate for speed
58 f0 = zeros(K, 1);
59 THDa = zeros(K, 1);
60 THD = zeros(K, 1);
61
62 for k =0:K-1
63     f0(k+1) = fMin + k *df;
64     freq = f0(k+1);
65     model = 'speaker_nonlin_ss';
66
67     load_system(model)
68     cs = getActiveConfigSet(model);
69     model_cs = cs.copy;

```

```

70         start_time = 0;
71         stop_time = N/Fs;
72         Ts = 1/Fs;
73         set_param(model_cs, 'FixedStep', num2str(Ts));
74     %         'StartTime', num2str(start_time),...
75     %         'StopTime', num2str(stop_time),...
76
77     % create timeseries obj from sequence
78     %     Freq = [0:length(Seq)-1]./Fs;
79     %     global sim_in;
80     %     sim_in = timeseries(Seq', Freq');
81     %     busSignal.busElement_1 = seq_ts;           % load to inport 1
82
83     % Run simulation and extract output
84     sim_out = sim(model, model_cs);
85     y_out = sim_out.sim_a.data(1:N);
86
87     spectrum = fftshift(fft(y_out'));
88     % %     spectrum = spectrum /H;
89
90     % sort out the frequency axis
91     if ceil(N/2) > N/2
92         f = 1/(2*N) * [0 : 2 : N-1];
93         spectrum = spectrum(N/2:end);
94     else
95         f = 1/(2*N) * [0 : 2 : N-1];
96         spectrum = spectrum((N/2+1):end);
97     end
98
99     F = f *Fs;
100    %     figure
101    %     plot(F, abs(spectrum))
102    %     xlim ([0 500])
103    %     hold on;
104    %
105
106    % select data point where distortion is expected
107    fDistorted = zeros(M, 1);
108    nDistorted = zeros(M, 1);
109    peak = zeros(M, 1);
110
111    % find the amplitude of the maximum distortion component
112    for m = 1:M
113        fDistorted(m) = m *f0(k+1);
114        nDistorted(m) = floor((N/Fs) *fDistorted(m));
115
116    %     plot(nDistorted(m)-margin : nDistorted(m)+margin, spectrum(nDistorted(m)-margin : N));
117
118        selection = abs(spectrum(nDistorted(m)-margin : nDistorted(m)+margin));
119    %     plot((nDistorted(m)-margin : nDistorted(m)+margin)./(N/Fs), selection)
120        peak(m) = max(selection);
121    %     disp(peak(m))
122    end
123
124    %     stem(fDistorted, peak);
125
126    distortion = sum(peak(2:M).^2);
127    THDa(k+1) = sqrt(distortion / (distortion + peak(1)^2));

```

```

128         THD(k+1) = sqrt(distortion)/peak(1);
129     end
130     %=====
131     % function plotTHD plots the THD in percent as function of frequency
132     % | list of inputs:      || definition:      |
133     % |-----||-----|
134     % | f0                  || distorted frequencies |
135     % | THDa                || accoustic THD    |
136     % | THD                 || standard THD    |
137     %
138     % | list of outputs:
139     % |-----|
140     % | graph of THDa and THD versus f0
141     %=====
142     % use the plotTHD function to generate a plot of the result
143     THDa = 100*THDa;
144     THD = 100*THD;
145     plotTHD(f0, THDa, THD);
146
147     function plotTHD(f0, THDa, THD)
148         figure
149         plot (f0, 100*THDa, 'o')
150         hold on
151         plot (f0, 100*THD, 'o')
152         hold off
153         title ('THD spectra')
154         xlabel ('f [Hz]')
155         ylabel ('Distortion [%]')
156         grid on
157         legend ('THDa', 'THD')
158     end

```

A.2. Impedance Measurement

A.2.1. nl_imp_meas.m

```

breaklines
1 %=====
2 % nl_imp_measure      :   script used to generate impedance curves from
3 %                     speaker measurements. Based on 'LS_measure' program
4 %                     written by Gerard Janssen
5 %-----
6 %   Author      :   Alexandros Skourtis-Cabrera
7 %   Date        :   17/6/2018
8 %=====
9
10 %----VARIABLES----%
11
12 Rref = 1.088;      % value of reference resistor in Ohm
13 R_dc = 5.6;       % rest (f = 0) resistance of speaker in Ohm
14 Ri = 16.2e3;      % input whatever
15
16 L = 18;           % generator length
17
18 Amin = 0.5;       % minimum Amplitude
19 Amax = 2;         % maximum Amplitude
20 dA = 0.2;         % Amplitude increment
21
22 Fs = 48000;       % Sampling frequency in Hz
23
24 nreps = 15;       % number of playback repetitions
25 %-----%
26
27 %----PLAYBACK AND RECORD----%
28 % Settings for play-record device
29   playdevice=3;
30   recdevice=1;
31   samplerate=Fs;
32   recnsamples=0;
33   recfirstchannel=1;
34   reclastchannel=2;
35   devicetype='win';
36 %-----%
37
38
39 df = Fs/(2^L);    % frequency increment per sample
40
41 Z2_f = zeros(1,2^L);
42
43 Seq = choose_ml_length(L); % load white noise snippet to use
44 len_seq = size(Seq,2); % length of sequence
45
46 rawdataout = zeros(len_seq , 2, nreps);
47
48 for idx = 1:nreps
49
50 %Q = ceil(len_seq*rand(1));
51 %Seq = -[ml_seq(1,Q:len_seq) ml_seq(1,1:Q-1)]; % Randomize Sequence
52 %Seq = -2*ml_seq +1;
53   playbuffer=[Seq',Seq'];
54

```

```

55
56 % play and record
57 RX = pa_wavplayrecord(playbuffer,[playdevice],[samplerate],[recnsamples],[recfirstchannel]
58
59 rawdataout(:, :, idx) = RX;
60
61 RX = RX';
62
63
64
65 % Filter Reference and Measured signal to BW
66 Ref = (rcos_window(0.01, size(RX, 2))) .* RX(1, :);
67 Meas = (rcos_window(0.01, size(RX, 2))) .* RX(2, :);
68
69 % Get freq domain Reference and Measured Signal
70 Ref_f = fft(Ref);
71 Meas_f = fft(Meas);
72 %-----%
73
74 % Measurement cable resistance
75 R_k = 0.05;
76 R_m = 0.05;
77 R_g = R_k + R_m;
78
79 % Determine the impedance
80 Z_f = Rref*Meas_f./(Ref_f-Meas_f);
81 %Z_f = Z_f.*Ri./(Ri-Z_f);
82 %Z_f = Z_f - R_g;
83
84 % Plot each measurement
85
86 % Freq = df*[1:size(Z_f,2)]; % Define frequency axis
87 % figure
88 % title('Complex Speaker impedance')
89 %
90 % subplot(2, 2, 1) % Impedance Amplitude
91 % semilogx(Freq, abs(Z_f))%,'LineWidth',2);
92 % title('Amplitude');
93 % xlabel('Frequency (Hz)');
94 % ylabel('|Z| [Ohm]');
95 % grid on
96 % axis([10 20000 0 50])
97 %
98 % subplot(2, 2, 2) % Impedance Phase
99 % semilogx(Freq, 180*unwrap(angle(Z_f))/pi)%,'LineWidth',2);
100 % title('Phase');
101 % xlabel('Frequency (Hz)');
102 % ylabel('Phase shift [degr]');
103 % grid on
104 %
105 % subplot(2, 2, 3) % Real and Imaginary impedance
106 % semilogx(Freq, real(Z_f), Freq, imag(Z_f))%,'LineWidth',2);
107 % title('Real- and Imaginary part');
108 % xlabel('Frequency (Hz)');
109 % ylabel('real(Z), imag(Z) [Ohm]');
110 % grid on
111
112 if idx == 1

```

```

113     Z2_f = Z_f;
114 else
115     %Z2_f = 0.75*Z2_f+0.25*Z_f;
116     Z2_f = (idx-1)*Z2_f/idx + Z_f/idx;
117 end
118
119 end
120
121 [B,A] = butter(1,0.05);
122 [Z2_f] = filtfilt(B,A,Z_f);
123
124
125 Freq = df*[1:size(Z2_f,2)]; % Define frequency axis
126
127
128
129
130 %----PLOTS----%
131
132
133 figure
134 title('Complex Speaker impedance')
135
136 subplot(2, 2, 1) % Impedance Amplitude
137 semilogx(Freq, abs(Z2_f))%, 'LineWidth', 2);
138 title('Amplitude');
139 xlabel('Frequency (Hz)');
140 ylabel('|Z| [Ohm]');
141 grid on
142 axis([10 20000 0 50])
143
144 subplot(2, 2, 2) % Impedance Phase
145 semilogx(Freq, 180*unwrap(angle(Z2_f))/pi)%, 'LineWidth', 2);
146 title('Phase');
147 xlabel('Frequency (Hz)');
148 ylabel('Phase shift [degr]');
149 grid on
150
151 subplot(2, 2, 3) % Real and Imaginary impedance
152 semilogx(Freq, real(Z2_f), Freq, imag(Z2_f))%, 'LineWidth', 2);
153 title('Real- and Imaginary part');
154 xlabel('Frequency (Hz)');
155 ylabel('real(Z), imag(Z) [Ohm]');
156 grid on
157
158
159
160 % FOR LEAST SQUARES FIT
161 % lsqcurvefit

```

A.2.2. nl_imp_extract.m

```

breaklines
1 %=====
2 % nl_imp_extract      :  script that extract impedance data from figure and
3 %                      crops it to certain frequency range
4 %-----
5 % in:      fig =  figure to extract from

```

```

6 %
7 % out:   Z   =   impedance vector
8 %       f   =   cropped frequency axis vector
9 %-----
10 % Author   :   Alexandros Skourtis-Cabrera
11 % Date     :   17/6/2018
12 %=====
13 function [Z, f] = nl_imp_extract(fig)
14 %-VARS-%
15
16 fmin = 10; % min freq to CROP
17 fmax = 20000; % max freq to CROP
18
19
20 %-EXTRACT-%
21 open(fig);
22
23 h = gcf; %current figure handle
24 % ..(1) for |Z|, ..(2) for angle(Z), ..(3) for Im(Z) & Re(Z)
25 axesObjs = get(h(1), 'Children'); %axes handles
26 dataObjs = get(axesObjs(1), 'Children'); %handles to low-level graphics objects in axes
27 % objTypes = get(dataObjs, 'Type'); %type of low-level graphics object
28
29 f = cell2mat(get(dataObjs, 'XData')); % data from low-level graphics objects
30 Z = cell2mat(get(dataObjs, 'YData'));
31
32 f = f(2,:); % choose correct row for |Z|
33 Z = Z(2,:); % choose correct row for |Z|
34
35 close;
36 % % plot
37 % figure;
38 % semilogx(f(1,:), Z(1,:))
39 % hold on;
40 % semilogx(f(2,:), Z(2,:))
41 % grid on;
42 % axis([10 20000 0 50])
43
44 %-CROP-%
45 N = length(f); % number of samples
46 n_min = round(N*fmin./f(end)); % 1st sample of cropped data
47 n_max = round(N*fmax./f(end)); % last sample of cropped data
48
49 f = f(n_min:n_max); % crop f
50 Z = Z(n_min:n_max); % crop Z
51
52 semilogx(f, Z);
53 %-SAVE-%
54 end

```

A.2.3. nl_imp_analysis.m

```

breaklines
1 %=====
2 % nl_imp_analysis.m : Script that (1) uses the Least Squares method to fit
3 % impedance measurement data and approximate value of
4 % of the Linear model components and (2) Tries to fit
5 % model for nonlinear component values. Based on the

```

```

6 %                                     'LS_measure' program written by Gerard Janssen
7 %-----
8 %   in      :   Z_f
9 %   out     :   par = [Re, Le, Cp, Lp, Rp]
10 %-----
11 %   Author  :   Alexandros Skourtis-Cabrera
12 %   Date    :   17/6/2018
13 %=====
14 function par = nl_imp_analysis(Z_f)
15
16
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18 % Variables %
19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20 j = sqrt(-1);
21
22 Fs = 48000;
23 L = 18;
24 df = Fs/(2^L); % frequency increment per sample
25
26 Freq = df*[1:size(Z_f,2)];
27
28 % offset_units = nl_imp_offsets; % matrix with conversions values
29 %                                     % Voff_in(mV) | Voff_speaker(V) | x(cm)
30 % disp_mm = 10*offset_units(:,3); % Speaker displacement in mm
31
32 plot(abs(Z_f))
33 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
34 % Parameter Calculation & Fitting %
35 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
36
37 Re = 5.6; % Coil resistance (should be constant)
38
39 [peaks, n_r] = findpeaks( abs(Z_f( round(50/df) : round(150/df) )) );
40 [R_res, idx_n] = max(peaks); % Impedance value at resonance peak
41 n_res = n_r(idx_n) + round(50/df);
42 Rp = R_res - Re; % R_res = Re + Rp
43 w_0 = 2*pi*Freq(n_res); % resonance frequency
44
45 % Calculate peak bandwidth
46 bw_err = 0.1; % acceptable error margin for localization
47 i = 1; % sample index
48 while 1 % left freq
49     if abs( abs(Z_f(n_res - i)) - (R_res/sqrt(2)) ) < bw_err; % check if imp at sample within
50         n_lf = i; % difference between peak and left bound in # samples
51         break
52     end
53     i=i+1;
54 end;
55 i = 1;
56 while 1 % right freq
57     if abs( abs(Z_f(n_res + i)) - (R_res/sqrt(2)) ) < bw_err; % check if imp at sample within
58         n_rg = i; % difference between peak and right bound in # samples
59         break
60     end
61     i=i+1;
62 end
63

```

```

64 Bw = 2*pi*df*(n_rg + n_lf + 1); % peak bandwidth in rad/s
65
66 Cp = 1/(Rp*Bw);
67
68 Lp = 1/(w_0^2 * Cp);
69
70 Le = 0;
71 for idx = round(2000/df):round(18000/df)
72     Le_temp = sqrt(abs(Z_f(idx))^2 - Re^2)/(2*pi*Freq(idx));
73     Le = (idx-1)*Le/idx + Le_temp/idx;
74
75 end
76
77 % par[] = [Re, Le, Cp, Lp, Rp]
78 % Z_abs_model = @(par,omegadata)abs((par(1) + 1./(1/par(5) + par(5)*(omegadata*par(3) - 1./om
79 % par0 = [5.6 149e-6 1.1e-3 2.5e-3 8.7]; % [10 149e-6 0.4e-3 8e-3 15]; % zero-displacement
80 %
81 % [par, resnorm, ~, exitflag, output] = lsqcurvefit(Z_abs_model, par0, 2*pi*f_disp(1,:), Zabs_disp(1
82
83 Z_model = abs((Re + 1./(1/Rp + Rp*((2*pi*Freq)*Cp - 1./((2*pi*Freq)*Lp)).^2))+ j*((2*pi*Freq)*
84
85 semilogx(Freq, abs(Z_f), Freq, Z_model);
86 axis([10 20000 0 50])
87
88 par = [Re, Le, Cp, Lp, Rp];

```

A.2.4. imp_datatoimp.m

```

breaklines
1 function Z2_f = imp_datatoimp(rx)
2 %=====
3 % imp_datatoimp.m : Script that calculates impedance Z(f) of speaker
4 %                   using raw data generated by "nl_imp_meas.m". Based
5 %                   on 'LS_measure' program written by Gerard Janssen
6 %-----
7 % in   : rx = raw recording data->   dim 1 = sequence length
8 %                                           dim 2 = channel
9 %                                           dim 3 = repetition number
10 %
11 % out  : Z2_f= impedance as a fuction of frequency averaged from all
12 %         measurements.
13 %-----
14 % Author   : Alexandros Skourtis-Cabrera
15 % Date     : 17/6/2018
16 %=====
17
18
19 %-----VARIABLES-----%
20
21 Rref = 1.088; % value of reference resistor in Ohm
22 R_dc = 5.6; % rest (f = 0) resistance of speaker in Ohm
23
24 Fs = 48000; % Sampling frequency in Hz
25
26 L = log2(size(rx, 1));
27 df = Fs/(2^L); % frequency increment per sample
28 %-----%
29

```

```

30 Z2_f = zeros(1,2^L);
31
32 for idx = 1:size(rx,3)
33   RX = rx(:, :, idx)';
34
35
36
37   % Filter Reference and Measured signal to BW
38   Ref = (rcos_window(0.01, size(RX,2))).*RX(1, :);
39   Meas = (rcos_window(0.01, size(RX,2))).*RX(2, :);
40
41   % Get freq domain Reference and Measured Signal
42   Ref_f = fft(Ref);
43   Meas_f = fft(Meas);
44   %-----%
45
46   % Measurement cable resistance
47   R_k = 0.05;
48   R_m = 0.05;
49   R_g = R_k + R_m;
50
51   % Determine the impedance
52   Z_f = Rref*Meas_f./(Ref_f-Meas_f);
53   %Z_f = Z_f.*Ri./(Ri-Z_f);
54   %Z_f = Z_f - R_g;
55
56   % Plot each measurement
57
58   % Freq = df*[1:size(Z_f,2)]; % Define frequency axis
59   % figure
60   % title('Complex Speaker impedance')
61   %
62   % subplot(2, 2, 1) % Impedance Amplitude
63   %   semilogx(Freq, abs(Z_f))%,'LineWidth',2);
64   %   title('Amplitude');
65   %   xlabel('Frequency (Hz)');
66   %   ylabel('|Z| [Ohm]');
67   %   grid on
68   %   axis([10 20000 0 50])
69   %
70   % subplot(2, 2, 2) % Impedance Phase
71   %   semilogx(Freq, 180*unwrap(angle(Z_f))/pi)%,'LineWidth',2);
72   %   title('Phase');
73   %   xlabel('Frequency (Hz)');
74   %   ylabel('Phase shift [degr]');
75   %   grid on
76   %
77   % subplot(2, 2, 3) % Real and Imaginary impedance
78   %   semilogx(Freq, real(Z_f), Freq, imag(Z_f))%,'LineWidth',2);
79   %   title('Real- and Imaginary part');
80   %   xlabel('Frequency (Hz)');
81   %   ylabel('real(Z), imag(Z) [Ohm]');
82   %   grid on
83
84   if idx == 1
85     Z2_f = Z_f;
86   else
87     %Z2_f = 0.75*Z2_f+0.25*Z_f;

```

```

88     Z2_f = (idx-1)*Z2_f/idx + Z_f/idx;
89     end
90
91     end
92
93     [B,A] = butter(1,0.05);
94     [Z2_f] = filtfilt(B,A,Z_f);
95     Z2_f = Z2_f';
96
97     Freq = df*[1:size(Z2_f,2)]; % Define frequency axis
98
99
100
101
102     %----PLOTS----%
103
104
105     % figure
106     % title('Complex Speaker impedance')
107     %
108     % subplot(2, 2, 1) % Impedance Amplitude
109     %     semilogx(Freq, abs(Z2_f))%,'LineWidth',2);
110     %     title('Amplitude');
111     %     xlabel('Frequency (Hz)');
112     %     ylabel('|Z| [Ohm]');
113     %     grid on
114     %     axis([10 20000 0 50])
115     %
116     % subplot(2, 2, 2) % Impedance Phase
117     %     semilogx(Freq, 180*unwrap(angle(Z2_f))/pi)%,'LineWidth',2);
118     %     title('Phase');
119     %     xlabel('Frequency (Hz)');
120     %     ylabel('Phase shift [degr]');
121     %     grid on
122     %
123     % subplot(2, 2, 3) % Real and Imaginary impedance
124     %     semilogx(Freq, real(Z2_f), Freq, imag(Z2_f))%,'LineWidth',2);
125     %     title('Real- and Imaginary part');
126     %     xlabel('Frequency (Hz)');
127     %     ylabel('real(Z), imag(Z) [Ohm]');
128     %     grid on
129
130
131
132     % FOR LEAST SQUARES FIT
133     % lsqcurvefit
134
135
136     end

```

A.3. Parameter Estimation

A.3.1. pars_nl.m

```

breaklines
1 %=====
2 % pars_nl.m      :      Code used to calculate characteristic curves of Km(x),
3 %                  Bl(x) and Le(x) using method described in chapter 4 of
4 %                  the thesis. this script is NOT COMPLETE and is
5 %                  currently being developed.
6 %-----
7 % Author       :      Aart-Peter Schipper & Alexandros Skourtis-Cabrera
8 % Date        :      17/6/2018
9 %=====
10
11 load pars.mat
12 load offset.mat
13 load Vref_offset.mat
14 %offset(:,3) = flipud(offset(:,3));
15
16 m = 0.011025; % mass of moving parts
17 n_pol = 5; % order of polynomial approximations
18 xPlot = linspace(-3.6e-3, 2e-3, 100); % x axis used for polynomial approximations
19 dfPlot = (2e-3 + 3.6e-3)/100;
20
21
22 Rref = 1.088;
23
24 %
25 Rp_0 = pars(6,3); %8.3; % 0 offset Rp for determination of constant pars
26 Cp_0 = pars(6,5); %7.5e-4; % 0 offset Cp for determination of constant pars
27 Bl_0 = sqrt(m/Cp_0); % 0 offset Rp for determination of constant pars
28 % "const." components (approx)
29 b = Bl_0^2/Rp_0;
30 Re = 5.6;
31
32
33 % % % non linear gyrator model (INCORRECT
34 % % Bl_meas = sqrt(m./ pars(:,5));
35 % % Le_meas = pars(:,2);
36 % % Cm_meas = pars(:,4)./((Bl_meas).^2);
37
38 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39 % Polynomial Approximations %
40 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
41
42 %current i(x) = i(1)*x + i(2)*x^2 + ... + i(n)*x^n
43 i_meas = Vref_offset/Rref;
44
45 [pI, ~, muI] = polyfit(offset(:,3), i_meas, n_pol);
46 %pI(end) = 0;
47 IPlot = polyval(pI, xPlot, [], muI);
48 didxPlot= diff(IPlot)./ dfPlot;
49
50 plot(offset(:,3), i_meas, 'o')
51 plot(xPlot, IPlot)
52 title('i(x)');
53 grid on
54

```

```

55     figure;
56     plot(xPlot(1:end-1), didxPlot);
57     title('di/dx');
58     grid on
59
60
61 % Q and Q*i
62 omega_0 = sqrt(1./(pars(:,4).*pars(:,5))); % Resonance frequency omega_0 = sqrt(1/(LpCp))
63 Q_meas = (omega_0.^2)*m;
64 [pQ, ~, muQ] = polyfit(offset(:,3), Q_meas, n_pol);
65 QPlot = polyval(pQ, xPlot, [], muQ);
66     figure;
67     plot(offset(:,3), Q_meas, 'o')
68     hold on
69     plot(xPlot, QPlot)
70     title('Q')
71     grid on
72
73 Qi_meas = Q_meas.*i_meas;
74 [pQi, ~, muQi] = polyfit(offset(:,3), Qi_meas, n_pol);
75 QiPlot = polyval(pQi, xPlot, [], muQi);
76     figure;
77     plot(offset(:,3), Qi_meas, 'o')
78     hold on
79     plot(xPlot, QiPlot)
80     title('Q*i')
81     grid on
82
83
84 % BI = Q*dx/di
85 BIPlot = QPlot(1:end-1)./didxPlot;
86
87     figure
88     plot(xPlot(1:end-1), BIPlot)
89     title('BI(x)')
90     grid on
91
92
93 % K*x = (Q*i)/(di/dx)
94 FkPlot = QiPlot(1:end-1)./didxPlot;
95     figure
96     plot(xPlot(1:end-1), FkPlot, xPlot(1:end-1), BIPlot.*IPlot(1:end-1));
97     title('Fk & BI')
98     grid on
99
100 KmPlot = FkPlot./xPlot(1:end-1);
101     figure
102     plot(xPlot(1:end-1), KmPlot);
103     title('Km(x)')
104     grid on
105
106 %% Polynomial approximation of Km = 1/Cm
107 % Q_meas = m./(pars(:,4).*pars(:,5)); % Q(x) = m/Lp(x)Cp(x) factor for diff eq. of Km
108 % pQ = polyfit(offset(:,3), Q_meas, n_pol); % fit polynomial to Q
109 % pKm = zeros(1, n_pol);
110 % for N = 0:n_pol % k_N = q_N/(N+1)
111 %     pKm(n_pol+1-N) = pQ(n_pol+1-N)/(N+1);
112 % end

```

```

113 % KmPlot = polyval(pKm, xPlot);
114
115
116 %% plot the large signal parameters for greenlight twopage
117 % load pars
118 % load offsets
119 %
120 % pars = abs(pars);
121 %
122 % x = [-3.6; -3.4; -3.2; -2.9; -2.5; -2.1; -1.8; -1.3; -.6; 0; .8; 1.3; ...
123 %      1.6; 1.7; 2];
124 % x = flipud(x);
125 %
126 %% derive small signal equivalent mechanical values
127 % B12 = 1./pars(:,5);
128 % Cm = pars(:,4)./B12;
129 % B1 = sqrt(B12);
130 % Le = pars(:,2);
131 %
132 %% do a polynomial approximation of the data
133 % pLe = polyfit(offset(:,3), Le_meas, n_pol);
134 % pB1 = polyfit(offset(:,3), B1_meas, n_pol);
135 % pCm = polyfit(offset(:,3), Cm_meas, n_pol);
136
137 % pLp = polyfit(offset(:,3), Lp_meas,5);
138
139
140 %% create data for plotting a curve
141 % LePlot = polyval(pLe, xPlot);
142 % B1Plot = polyval(pB1, xPlot);
143 % CmPlot = polyval(pCm, xPlot);
144
145
146 % find the spring force
147 % Fk = 1e-5*x./Cm_meas;
148 % FkPlot = 1e-5*xPlot./CmPlot;
149 %
150 %% plot the graphs of the values
151 % figure
152 % subplot(131)
153 % plot(x, Le_meas*1e6, 'o')
154 % hold on
155 % plot(xPlot, LePlot*1e6)
156 % hold off
157 % title ('Voice Coil Inductance ')
158 % xlabel ('x [mm] ')
159 % xlim ([-4 4])
160 % y = ylabel ('$L_e\;\lbrack\mu H \rbrack$ ');
161 % set(y,'interpreter','Latex','FontSize',14)
162 % grid on
163 %
164 % subplot(132)
165 % plot(x, B1_meas, 'o')
166 % hold on
167 % plot(xPlot, B1Plot)
168 % hold off
169 % title ('Force Factor ')
170 % xlabel ('x [mm] ')

```

```

171 % xlim ([-4 4])
172 % ylabel ('Bl')
173 % grid on
174 %
175 % subplot(133)
176 % plot(x, Cm_meas, 'o')
177 % hold on
178 % plot(xPlot, CmPlot)
179 % hold off
180 % title ('Spider Force')
181 % xlabel ('x [mm]')
182 % xlim ([-4 4])
183 % ylabel ('F')
184 % grid on

```

A.3.2. pars_nl_old.m

```

breaklines
1 %=====
2 % pars_nl_old.m      :   Code used for the initial calculation of the chara-
3 %                    :   cteristic curves of Km(x), Bl(x) and Le(x). The
4 %                    :   method used in this code is INCORRECT and has been
5 %                    :   replaced by 'pars_nl.m'
6 %-----
7 %   Author      :   Aart-Peter Schipper & Alexandros Skourtis-Cabrera
8 %   Date        :   17/6/2018
9 %=====
10
11 % measure model parameters using small
12 %
13 load pars.mat
14 load offset.mat
15 load Vref_offset.mat
16 %offset(:,3) = flipud(offset(:,3));
17
18 m = 0.011025; % mass of moving parts
19 n_pol = 5; % order of polynomial approximations
20 xPlot = linspace(-3.6e-3, 2e-3, 100); % x axis used for polynomial approximations
21 dfPlot = (2e-3 + 3.6e-3)/100;
22
23
24 Rref = 1.088;
25
26 %
27 Rp_0 = pars(6,3); %8.3; % 0 offset Rp for determination of constant pars
28 Cp_0 = pars(6,5); %7.5e-4; % 0 offset Cp for determination of constant pars
29 Bl_0 = sqrt(m/Cp_0); % 0 offset Rp for determination of constant pars
30 % "const." components (approx)
31 b = Bl_0^2/Rp_0;
32 Re = 5.6;
33
34
35 % non linear gyrator model (INCORRECT!)
36 Bl_meas = sqrt(m./pars(:,5));
37 Le_meas = pars(:,2);
38 Cm_meas = pars(:,4)./((Bl_meas).^2);
39
40

```

```

41 % plot the large signal parameters for greenlight twopage
42 pars = abs(pars);
43
44 %% derrive small signal equivalent mechanical values
45 % B12 = 1./pars(:,5);
46 % Cm = pars(:,4)./B12;
47 % B1 = sqrt(B12);
48 % Le = pars(:,2);
49
50 % do a polynomial approximation of the data
51 [pLe, ~, muLe] = polyfit(offset(:,3), Le_meas, n_pol);
52 [pB1, ~, muB1] = polyfit(offset(:,3), B1_meas, n_pol);
53 [pCm, ~, muCm]= polyfit(offset(:,3), Cm_meas, n_pol);
54
55 % pLp = polyfit(offset(:,3), Lp_meas,5);
56
57
58 % create data for plotting a curve
59 LePlot = polyval(pLe, xPlot, [], muLe);
60 B1Plot = polyval(pB1, xPlot, [], muB1);
61 CmPlot = polyval(pCm, xPlot, [], muCm);
62
63
64 % find the spring force
65 % Fk = offset(:,3)./Cm_meas;
66 % FkPlot = xPlot./CmPlot;
67
68 % plot the graphs of the values
69 figure
70 subplot(131)
71 plot(offset(:,3), Le_meas, 'o')
72 hold on
73 plot(xPlot, LePlot)
74 hold off
75 title ('Voice Coil Inductance Le(x)')
76 xlabel ('x [m]')
77 xlim ([-4 4]*10e-4)
78 y = ylabel ('Le [H]');
79 set(y, 'interpreter', 'Latex', 'FontSize', 14)
80 grid on
81
82 subplot(132)
83 plot(offset(:,3), B1_meas, 'o')
84 hold on
85 plot(xPlot, B1Plot)
86 hold off
87 title ('Force Factor B1(x)')
88 xlabel ('x [m]')
89 xlim ([-4 4]*10e-4)
90 ylabel ('B1 [N/A]')
91 grid on
92
93 subplot(133)
94 plot(offset(:,3), 1./Cm_meas, 'o')
95 hold on
96 plot(xPlot, 1./CmPlot)
97 hold off
98 title ('Stifness Km(x)')

```

```
99 xlabel ('x [m]')
100 xlim ([-4 4]*10e-4)
101 ylabel ('Km [N/m]')
102 grid on
```

A.4. Other

A.4.1. `fourier.m`

```

breaklines
1 %=====
2 % function fourier generates a shifted fourier transform
3 % | list of inputs:      || definition:                |
4 % |-----||-----|
5 % | x                    || signal to be converted (column vector) |
6 % | Fs                   || sample frequency                |
7 % | N                    || number of samples                |
8 %
9 % | list of outputs:    || definition:                |
10 % |-----||-----|
11 % | X                    || fourier transformed signal    |
12 % | dBX                  || X in dB                      |
13 % | f                    || frequency axis                |
14 % | M                    || number of samples in transformed signal |
15 %
16 % date   : 11/5/2018
17 % author : Aart-Peter Schipper
18 %=====
19 function [X, dBX, f, M] = fourier (x, Fs, N)
20     % find number of samples if N is not specified
21     if N < 1
22         N = size(x, 1);
23     end
24
25     % sort out the frequency axis
26     if ceil(N/2) > N/2
27         M = 0.5 + N/2;
28         f = 1/(2*N) * [0 : 2 : N-1];
29     else
30         M = N/2;
31         f = 1/(2*N) * [0 : 2 : N-1];
32     end
33
34     % apply the digital fourier transform and convert do decibel
35     %window = hamming(N);
36     %X = fft(x.*window);
37     X = fft(x);
38     X = X(1:M, :);
39     dBX = 10 * log10(abs(X));
40
41     % convert normalized frequency to actual frequency
42     if Fs > 0
43         f = transpose(Fs * f);
44     end
45 end

```

A.4.2. `playRec.m`

```

breaklines
1 %=====
2 % function playRec plays frequency f0 and returns the recorded signal
3 % | list of inputs:      || definition:                |
4 % |-----||-----|
5 % | f0                  || input frequency 1            |

```

```

6 % | f1          || input frequency 2          |
7 % | Fs          || sample frequency          |
8 % | N           || number of samples          |
9 % | A0          || amplitude of the first frequency |
10 % | A1         || secondary amplitude, if A1 = 0, A1 = A0|
11 %
12 % | list of outputs:
13 % |-----|
14 % | x : signal that is played through the speaker
15 % | y : (Nx2) vector containing the recorded signal
16 %
17 % this function uses the pawavplayw file from author Gerard Janssen.
18 %
19 % date : 11/5/2018
20 % author : Aart-Peter Schipper
21 %=====
22 function [x, y] = playRec (f0, f1, Fs, N, A0, A1, x)
23 % find out what signal to play
24 if x ~= 0 % x already defined
25 elseif (f0 <= 0) && (f1 <= 0) % create white noise
26 flatSpectrum();
27 elseif f0 <= 0 % play one frequency
28 getFrequency1(f1);
29 elseif f1 <= 0 % play one frequency
30 getFrequency1(f0);
31 else % play two frequencies
32 getFrequency2();
33 end
34
35 % set device parameters
36 buf = [x, x]; % dual channel signal
37 playDev = 3; % use pawavplayw in Command Window to list audio
38 recDev = 1; % devices
39 n = 0; % number of samples for recording, if n = 0, n = N
40 recChan = [1, 2]; % first and last channel for recording
41
42 % play and record using pawavplayw and convert to double
43 % y = pawavplaya(buf, playDev, Fs, recChan(1), recChan(2), n, recDev);
44 % y = pawavplayx(buf, playDev, Fs, recChan(1), recChan(2), n, recDev);
45 y = pawavplayw(buf, playDev, Fs, recChan(1), recChan(2), n, recDev);
46 y = double(y);
47
48 %-----
49 % list of nested functions:
50 %=====
51 % function whiteNoise generates a random signal with zero offset
52 % | list of inputs: || definition: |
53 % |-----||-----|
54 % | N           || number of samples |
55 % | A           || amplitude of the input signal |
56 %
57 % | list of outputs:
58 % |-----|
59 % | x : (approximately) white noise signal
60 %=====
61 function flatSpectrum ()
62 % find the number of shift registers
63 m = log2(N +1);

```



```

122         g = [0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0; ...
123              0;0;0;0;0;1];
124     case 29
125         g = [0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0; ...
126              0;0;0;0;0;0;1];
127     case 30
128         g = [1;0;0;0;0;0;0;0;0;0;0;0;0;1;1;0;0;0;0;0; ...
129              0;0;0;0;0;0;0;1];
130     case 31
131         g = [0;0;1;0;0;0;0;0;0;0;0;0;0;0;1;1;0;0;0;0;0; ...
132              0;0;0;0;0;0;0;1];
133     case 32
134         g = [1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0; ...
135              0;0;0;0;1;1;0;0;0;1];
136     otherwise
137         error('signal size not supported')
138     end
139
140     a = ones(1,m);
141     x = zeros(N,1);
142
143     % create pseudo random signal
144     for k = 1:N
145         tmp = mod(sum(a*g), 2); % find new least significant bit
146         a(2:m) = a(1:m-1); % shift left
147         a(1) = tmp; % shift new bit in
148         x(k) = a(1); % set pseudo noise signal
149     end
150     else % use white noise otherwise
151         x = round(rand(N,1));
152     end
153
154     % conditioning of the sequence
155     x = -2*x + 1;
156     x = A0*x;
157 end
158
159 %=====
160 % function getFrequency1 generates a signal at one frequency
161 % | list of inputs:      || definition:                |
162 % |-----||-----|
163 % | f2                    || desired frequency        |
164 % | Fs                    || sample frequency      |
165 % | N                     || number of samples   |
166 % | A                     || amplitude of the input signal |
167 %
168 % | list of outputs:
169 % |-----|
170 % | x : signal with frequency f2
171 %=====
172 function getFrequency1 (f2)
173     dt = transpose(0:N-1);
174     x = A0*sin(2*pi*f2/Fs .* dt);
175 end
176
177 %=====
178 % function getFrequency1 generates a signal at one frequency
179 % | list of inputs:      || definition:                |

```

```

180 % |-----||-----|
181 % | Fs          || sample frequency          |
182 % | N          || number of samples          |
183 % | A          || amplitude of the input signal |
184 %
185 % | list of outputs:
186 % |-----|
187 % | x : signal with frequencies f0 and f1
188 %=====
189 function getFrequency2
190     % find out if amplitude A1 is set, if not, set it equal to A0
191     if A1 == 0
192         A1 = A0;
193     end
194
195     dt = transpose(0:N-1);
196     x = A0*sin(2*pi*f0/Fs .* dt) +A1*sin(2*pi*f1/Fs .* dt);
197 end
198 end

```

A.4.3. gen_freq.m

```

breaklines
1 %=====
2 % gen_freq.m : Short script that generates a sine wave Sequence 'seq'
3 %             of frequency 'freq', amplitude 'amplitude', sampling
4 %             rate 'Fs' and sample count 'N'.
5 %-----
6 % Author      :   Alexandros Skourtis-Cabrera
7 % Date        :   17/6/2018
8 %=====
9 function seq = gen_freq(freq, amplitude, Fs, duration_seconds, N)
10     % wavelength in # of samples
11
12     t_single = (0:1:N-1)/Fs;
13     seq = amplitude.*sin((2*pi*freq).*t_single); % generate sigle wavelegth sine wave
14
15     lambda_nsamples = Fs/freq; % wavelength in # of samples
16 %
17     t_single = 0:1:lambda_nsamples-1;
18     seq_single = amplitude.*sin((2*pi/lambda_nsamples).*t_single); % generate sigle wavelegth
19 %
20     nrepetitions = duration_seconds*freq; % numberof repetitio
21     seq = repmat(seq_single, 1, nrepetitions); % replicate single tone 2^ml_
22
23
24     % Length of signal in samples
25
26     %PLOT x_in
27     %plot(t, Seq);

```

A.4.4. record_freq.m

```

breaklines
1 %=====
2 % record_freq.m : Short script that records the playback of a Sequence
3 %               'Seq' on the loudspeaker with a sampling rate of 'Fs'
4 %               using USB soundcard. Uses the pawavplayw file from

```

```

5 %                               author Gerard Janssen .
6 %-----
7 % Author      :   Alexandros Skourtis-Cabrera
8 % Date       :   17/6/2018
9 %=====
10 function y_out = record_freq( Seq, Fs) % NOTE: amplitude has to be non-integer
11     % generate one tone signal
12
13
14 % % % REPLACED BY gen_freq
15 % %     lambda_nsamples = Fs/freq;     % wavelength in # of samples
16 % %
17 % %     t_single = 0:1:lambda_nsamples-1;
18 % %     seq_single = amplitude.*sin((2*pi/lambda_nsamples).*t_single); % generate sigle wavele
19 % %
20 % %     nrepetitions = Fs*duration_seconds/freq;
21 % %     Seq = repmat(seq_single , 1, nrepetitions); % replicate single tone 2^m
22 % %     x_in = [Seq' , Seq'];
23     %PLOT x_in
24 % %     plot(t, Seq);
25 % % [x_in, seq_length] = gen_freq( freq, amplitude, Fs, duration_seconds); % generate repeated
26
27     % time axis
28
29     %PLOT x_in
30     %plot(t, x_in(:,1));
31
32
33     %Settings for play-record device
34     playbuffer = [Seq' , Seq'];
35     playdevice= 4;
36     recdevice= 1;
37     samplerate= Fs;
38     recnsamples= length(Seq);
39     recnsamples= 0;
40     recfirstchannel=1;
41     reclastchannel= 2;
42     devicetype= 'win';
43
44
45
46     %Play and record
47     y_out = pa_wavplayrecord(playbuffer , playdevice , samplerate , recnsamples , recfirstchannel ,
48
49 % PLOT
50     % plot(Seq);
51     % hold off;
52     % plot(y_out(:,2));

```

A.4.5. sim_freq.m

```

breaklines
1 %=====
2 % sim_freq.m :   Script that simulates the playback of a sine wave
3 %               of Specified frequency 'freq', amplitude 'A', sample
4 %               rate Fs and sample count. Parameters are into Simulink
5 %               loudspeaker model.

```

```
6 %-----
7 % Author    :   Alexandros Skourtis-Cabrera
8 % Date      :   17/6/2018
9 %=====
10 function y_out = sim_freq(freq , A, Fs, N)
11 % Set Simulink parameters
12     model = 'speaker_nonlin_ss';
13     load_system(model)
14     cs = getActiveConfigSet(model);
15     model_cs = cs.copy;
16     start_time = 0;
17     stop_time = N;
18     Ts = 1/Fs;
19     set_param(model_cs , ...
20         'StartTime', num2str(start_time) , ...
21         'StopTime', num2str(stop_time));
22
23     % create timeseries obj from sequence
24     %     Freq = [0:length(Seq)-1]./Fs;
25     %     global sim_in;
26     %     sim_in = timeseries(Seq', Freq');
27     %     busSignal.busElement_1 = seq_ts;           % load to inport 1
28
29     % Run simulation and extract output
30     sim_out = sim(model, model_cs);
31     y_out = sim_out.yout{1}.Values.Data;
32 end
```




Measurement Equipment

The speaker being modelled is the JAMO D115 passive loudspeaker system. The measurement setup used consists of the following equipment :

- Behringer ECM8000 measurement microphone
- AudioBox iTwo USB Audio interface

Bibliography

- [1] Khalid Mohammad Al-Ali. *Loudspeakers: Modeling and Control*. PhD thesis, University of California at Berkeley, 1999.
- [2] W. Merwe B. Ferreira. *Electronic and Electromechanic Power Conversion*. Wiley-IEEE Press, 2014.
- [3] R.T. Beyer. *Nonlinear Acoustics*. Naval Ship Systems Command, 1974.
- [4] Sybold Hijlkema Bishwas Regmi. Motional feedback in a bass loudspeaker, digital implementation. Technical report, TU Delft, 2018.
- [5] Pascal Brunet. *Nonlinear System Modeling and Identification of Loudspeakers*. PhD thesis, Northeastern University Boston, Massachusetts, apr 2014.
- [6] Y.C. Shiah J.H. Huang C. Chang, C. Wang. Numerical and experimental analysis of harmonic distortion in a moving-coil loudspeaker. *Communications in Nonlinear Science and Numerical Simulation*, 18:1902–1915, 2013.
- [7] Chiu George Cheng Chi-Cheng Peng Huei Chen, C-Y. Passive voice coil feedback control of closed-box subwoofer systems. *Journal of Acoustical Society of America Proceedings of The Institution of Mechanical Engineers Part C-journal of Mechanical Engineering Science*, 214(7):995–1005, July 2000.
- [8] Y.C. Shiah-Jin H.Huang Chun Chang, Chi-Chang Wang. Numerical and experimental analysis of harmonic distortion in a moving-coil loudspeaker. *Communications in Nonlinear Science and Numerical Simulation*, 19(7):1902–1915, Jul 2013.
- [9] J.J. Feeley. A simple dynamic model for eddy currents in a magnetic actuator. *IEEE Transactions on Magnetics*, 32(2):453–458, March 1996.
- [10] Iain Forgusson. Loudspeaker corss section, May 2010. URL <http://en.wikipedia.org/wiki/File:Speaker-cross-section.svg>.
- [11] Emami-Naeni Franklin, Powell. *Feedback Control of Dynamic Systems*. Pearson, seventh edition, 2014.
- [12] R. Hilmisson. Feedback linearisation of low frequency loudspeakers. Master’s thesis, Technical University of Denmark, sep 2009.
- [13] S.H. de Koning J.A. Klaassen. Motional feedback with loudspeakers. *Phillips Technical Review*, 29(5):148–157, 1968.
- [14] M. Jakobsson, D. Larsson. Modelling and compensation of nonlinear loudspeakers,. Master’s thesis, Chalmers University of Technology, 2010.
- [15] Dr.ir. G.J.M. Janssen, Dr.ir. J.F. Creemer, Dr.ir. D. Djairam, Dr.ir. M. Gibescu, Dr.ing. I.E. Lager, Dr.ir. N.P. van der Meijs, Dr.ir. S. Vollebregt, Dr.ing. B. Roodenburg, Dr. J. Hoekstra, and Ing. X. van Rijnsoever. *Lab Courses EE Semester 1, Student Manual*. TU Delft, 2017-2018.
- [16] F.T. Agerkvist K. Thorborg, C. Tinggaard. Frequency dependence of damping and compliance in loudspeaker suspensions. *Journal of the Audio Engineering Society*, 58(6):472–486, 2010.

- [17] W. Klippel. Extended creep modeling. https://www.klippel.de/fileadmin/_migrated/content_uploads/AN_49_Extended_Creep_Modeling.pdf, 2018. Accessed: 18/6/2018.
- [18] Wolfgang Klippel. Loudspeaker nonlinearities - causes, parameters, symptoms. *Journal of Audio Engineering Society*, 54(10):907–939, October 2006.
- [19] A.B. Coppins J.V. Sanders L.E. Kinsler, A.R. Frey. *Fundamentals of Acoustics*. John Wiley and Sons, fourth edition, 2000.
- [20] Yaoyu Li and G. T. C. Chiu. Control of loudspeakers using disturbance-observer-type velocity estimation. *IEEE/ASME Transactions on Mechatronics*, 10(1):111–117, February 2005.
- [21] M.J. Lighthill. On sound generated aerodynamically. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 211(1107):564–587, March 1952.
- [22] Chau-Min Huang Mingsian R. Bai. Expert diagnostic system for moving-coil loudspeakers using nonlinear modeling. *Journal of Acoustical Society of America*, 125(2):819–830, February 2009.
- [23] Robert-H Munnig Schmidt. Motional feedback theory in a nutshell. Technical report, RMS Acoustics & Mechatronics, 2017.
- [24] D.V. Schroeder. *An Introduction to Thermal Physics*. Pearson, first edition, 2014.
- [25] C. Sean. A direct pwm loudspeaker feedback system. Master’s thesis, Massachusetts Institute of Technology, 1996.
- [26] Paolo La Torraca. Feedback control of a dynamic loudspeaker with embedded sensor coil. Master’s thesis, Politecnico di Milano, 2015.
- [27] R. Valk. Control of voicecoil transducers. Master’s thesis, Delft University of Technology, 2013.
- [28] J.H. Huang Y. Tsai, C. Wang. Inverse determination of the nonlinear force factor of moving-coil loudspeaker motor systems. *Noise Control: Theory, Application and Optimization in Engineering*, pp. 167-188, 2014.