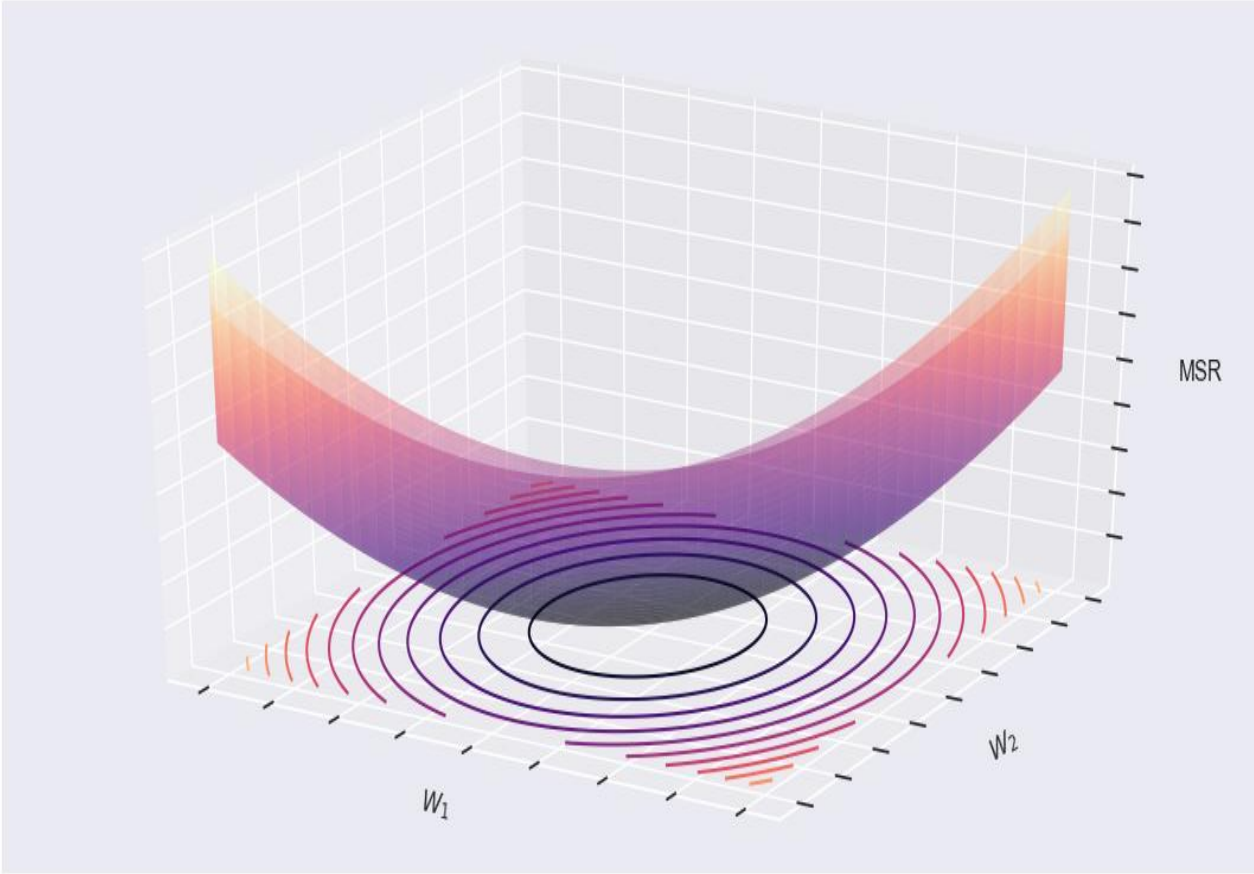
CHALMERS
UNIVERSITY OF TECHNOLOGY



Prediction of the gain values produced by a Bone Anchored Hearing Aid (BAHA) using Machine Learning

Master's Thesis in Biomedical Engineering

RYAN THOMAS SEBASTIAN

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019
Master's Thesis 2019

MASTER'S THESIS 2019

Prediction of the gain values produced by a Bone Anchored Hearing Aid (BAHA) using Machine Learning

Master's Thesis in Biomedical Engineering

RYAN THOMAS SEBASTIAN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2019

Master's Thesis in Biomedical Engineering

RYAN THOMAS SEBASTIAN

Supervisors: Thomas Rylander, Department of Electrical Engineering
Armin Azhirnian, Cochlear Bone Anchored Solutions AB
Examiner: Thomas Rylander
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
Image of a 3D plot of the Mean Squared Residual (MSR) of a predictive model as a function of two of its weights. The contours of the 3D plot on a 2D co-ordinate plane of the weights is also shown.

Prediction of the gain values produced by a Bone Anchored Hearing Aid (BAHA) using Machine Learning

Master's thesis in Biomedical Engineering

RYAN THOMAS SEBASTIAN

Department of Electrical Engineering
Chalmers University of Technology

Abstract

Hearing loss is a very prevalent sensory disorder and hearing aids provide the most common means to alleviate this problem. The effectiveness of hearing aids heavily depends on the device's tuning or "fitting" to the patient's particular hearing deficit. Hearing aid fitting is a process where the hearing aid is tuned by adjusting some inbuilt parameters to produce the desired output. The effect of changing these parameters on the response of the hearing aid can often be complex and this response needs to be modelled. This thesis focuses on modelling the output response of a bone anchored hearing aid (BAHA) using machine learning based on regression methods. Linear regression using polynomials of different orders are used and the effects of ridge and LASSO regularization are studied. This study produced several predictive models and their capabilities were evaluated using a test dataset. It was shown that regularization can be beneficial when producing higher order predictive models and LASSO regularization proved to be particularly effective. This thesis was carried out in collaboration with Cochlear Bone Anchored Solutions (CBAS) AB and the performance of the models proposed in this thesis was found to be superior in most cases when benchmarked against the methods currently used by CBAS.

Key words:

Machine learning, Linear regression, Regularization, Ridge, LASSO, Hearing aids, Bone anchored hearing aids.

Acknowledgements

I would like to express my most heartfelt gratitude towards my supervisor and examiner Thomas Rylander, who always guided me in the right direction with his extensive knowledge and pedagogical skills. I would also like to sincerely thank Armin Azhirnian from Cochlear Bone Anchored Solutions (CBAS), who was very approachable and knowledgeable and educated me on the inner workings of hearing aids, which provided critical understanding of the problem tackled in this thesis. In addition to supervising me, he acted as my liaison with CBAS which made the administrative and collaborative processes easy. The knowledge I acquired from you both will be invaluable for my engineering career.

Special thanks need to be expressed towards CBAS for providing the required data and Chalmers University of Technology needs to be thanked for its literature resources. In addition, both organisations, dedicated some its employees' working hours towards this thesis and I am very thankful for that.

The moral support I received during my work for this thesis was invaluable and I would like to thank my family and friends for that. Special thanks are expressed towards my peers, who are the following: Alberto Nieto, Asta Danauskienė, Mauricio Machado, Laura Guerrero, Isabelle Montoya, Tryggvi Kaspersen, Berglind Þorgeirsdóttir, Alexander Doan, Abdulrahman Alsaggaff and Wilhelm Råbergh. I would not have reached the endgame without your "whatever it takes" attitude. Finally, I would like to express my gratitude towards my girlfriend Monisha Chakravarthy for her boundless support.

Ryan Thomas Sebastian, Gothenburg, 2019

Contents

Notations

Notation	Meaning
$\mathbf{G}$	Matrix named $\mathbf{G}$
$\mathbf{g}$	Vector named $\mathbf{g}$
$\boldsymbol{\mathcal{G}}$	Domain named $\mathcal{G}$
$\mapsto$	"maps to" e.g. $\mathcal{P} \mapsto \mathcal{G}$
$\in$	"element of"
$\frac{\boldsymbol{\partial L_1}(\mathrm{W})}{\boldsymbol{\partial \mathbf{W}_{1,1}}}$	Partial derivative of $L_1(\mathbf{W})$ with respect to $\mathrm{W}_{1,1}$

1. Introduction

Hearing loss is a very prevalent sensory disorder, which was estimated to effect 466 million people or 5% of the world's population in 2012 by the World Health Organisation [1]. Hearing loss can have drastic effects on the quality of life of the victims as sound is primarily used by humans for communication and therefore it can lead to social isolation [2].

Hearing loss can be divided into two types: *conductive hearing loss* and *cochlear hearing loss*. Conductive hearing loss occurs when a decline occurs in the transfer efficiency of sound through the outer or middle ear. This can occur due to multiple causes such as obstacles in the sound transmission pathway (presence of ear wax or fluid due to infection), damage to the *tympanic membrane* (eardrum), or change in the mechanical properties of the ossicles (tiny bones) in the middle ear, where these structures don't vibrate adequately to transmit sound. The end result of conductive hearing loss is an attenuation of the sound reaching the cochlea. The cochlea is a fluid filled component in the inner ear that vibrate with the incident sound waves and convert the vibrations to nerve impulses which travels along the auditory nerve. The sound attenuation can be frequency dependant and this means that in addition to overall attenuation of incoming sounds, distortions in the sound may also occur [3].

Cochlear hearing loss occurs when the cochlea is damaged, and this damage can be caused by loud sounds and ototoxic chemicals. In addition, cochlear hearing loss can also be caused by genetic factors, autoimmune diseases and infections. Sometimes these factors can also cause damage to neural structures. When both the cochlea and the neural structures are damaged, the name *sensorineural hearing loss* is used to describe the condition. Sometimes patients can also have *mixed hearing loss* where both conductive and sensorineural hearing losses are present [3].

While partial or full hearing restoration is sometimes possible through surgical or pharmaceutical interventions, the use of hearing aids remains the main utility to combat hearing loss. Hearing aids are medical devices which are essentially packages that house microphones, preamplifiers, analogue to digital converters, digital signal processors, transducers and batteries. Hearing aids come in different styles based on its placement on the patient's head. In-the-ear (ITE) type hearing aids fit in the concha of the ear and are externally visible, unlike the completely-in-the-canal (CIC) hearing aids which fit entirely in the ear canal, which cannot be seen from the outside. However, the most common hearing aid style is the behind-the-ear (BTE) hearing aid which fits behind the patient's ear and the sound is then transmitted into the ear canal through a thin acoustic tube[4].

The most common reason for patients to adopt the use of hearing aids is due to poor speech perception [5]. Even though, hearing aids do help overcome hearing loss to some extent, normal hearing is not fully restored. This is because hearing aids function well to increase audibility of sounds, but this doesn't guarantee comprehensibility of the sound such as speech. This is especially true in environments with abundant background noise [6].The main reason behind this particular shortcoming of hearing aids is due to the fact that a decrease in sensitivity to sounds is only one part of hearing loss. A healthy cochlea adds distortions to an incoming sound from the environment and these distortions along with the original incoming

sound create a neural response in the auditory nerve. When the cochlea is damaged (hearing loss) and a hearing aid is used in an attempt to alleviate the problem, the hearing aid's output doesn't incorporate the distortions that a healthy cochlea would produce naturally. Thus, the neural activity that results from the hearing aid's output is different to that caused by a healthy cochlea and therefore the sound is deciphered differently by the brain. The transformation of sound in the cochlea with respect to these distortions is not very well understood and therefore it cannot be mimicked successfully by hearing aids currently [7], [8] .

It should be noted that the scope of this thesis will be limited to the use of machine learning for hearing aid fitting. This means that the hearing aid itself will often be treated as a "black-box", where it is simply considered as a system with inputs, outputs and tuneable parameters. The inner workings of hearing aids and the function of human hearing will not be explored in great detail.

2. Bone Anchored Hearing Aid (BAHA)

BAHA is a type of hearing aid which transfers sound to the cochlea through the skull bone, this is known as *bone conduction* as opposed to *air conduction* which is utilised by conventional hearing aids [9]. BAHAs requires a titanium screw known as a *fixture* to be surgically implanted into the mastoid bone of the skull. A detachable sound processor with a transducer is attached to the fixture via a piece which protrudes though the skin known as an *abutment.* Sometime after implantation, the fixture becomes strongly attached to the bone through a process known as *osseointegration* [10].

There are also other bone conduction hearing aids which have transducers placed against the skin covering the mastoid bone using a head band, however these are not as effective as they are somewhat free to move around. In addition, the presence of skin and hair in the conduction pathway can have undesired effects on the transmission to the cochlea [11][10].

The use of a BAHA is well motivated when an air conduction hearing aid is simply not enough to alleviate the patient's hearing loss. Such patients are affected by conductive or mixed hearing loss where the sound conduction mechanism that carries sound to the cochlea is no longer functional. This means that if air conduction hearing aids are used, the output sound from the hearing aid will not reach the cochlea effectively. Bone conduction solves this problem by bypassing the air conduction pathway and the sound travels directly from the mastoid bone to the cochlea. The use of a BAHA is sometimes required by hearing loss patients who encounter fluid discharge from the ear due to infections. In such cases, the fluid will interfere with the function of the air conduction hearing aids [9] [12].

2.1 Overview

A simplified schematic of the functioning of a BAHA can be seen in Figure 1.

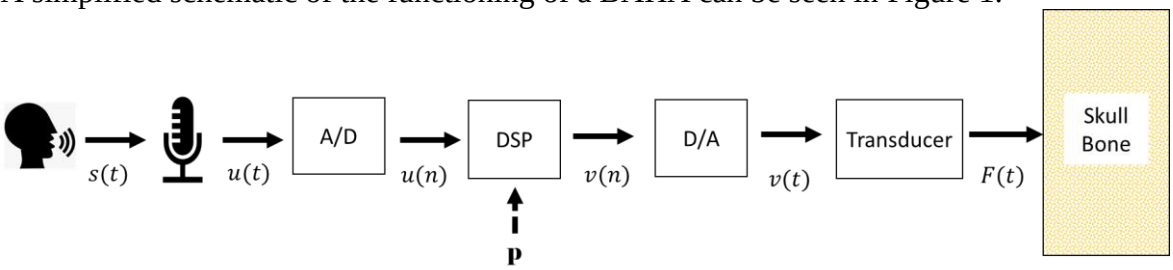


Figure 1: A schematic view of the signal processing chain of a BAHA showing a Microphone, Analog to Digital Converter (A/D), Digital Signal Processor (DSP), Digital to Analog converter (D/A) and the Transducer. The DSP can be tuned by adjusting the parameters represented in the parameter vector $\boldsymbol{p}$ *[4].*

The incoming acoustic signal $s(t)$ is received by the microphone and fed to the Analog to Digital converter (A/D) as a voltage $u(t)$. This continuous analogue signal $u(t)$ is converted to a discrete digital signal $u(n)$ by the A/D. The Digital Signal Processor (DSP) then applies some operations on $u(n)$ based on the parameters set in the parameter vector $\mathbf{p}$. The output signal $v(n)$ from the DSP is then converted to an analogue signal $v(t)$ by the Digital to Analogue converter (D/A). $v(t)$ is the input voltage to the transducer which vibrates on the skull bone with the force $F(t)$ [4]. Given a time-harmonic input sound signal with the frequency ω and the amplitude $\hat{s}(\omega)$, we define the power gain for the system as:

$$G_s(\omega) = 20\ \log_{10}\left(\frac{\frac{\left|\hat{F}(\omega)\right|}{\left|\hat{s}(\omega)\right|}}{\frac{1\ \mu N}{20\ \mu Pa}}\right) \quad (1)$$

where $\hat{F}(\omega)$ is the output force signal's amplitude associated with the fundamental frequency component. Thus, all the harmonics of the output force are discarded. The power gain $G_s(\omega)$ has the unit decibels (dB) relative to $\frac{1\ \mu N}{20\ \mu Pa}$ [13].

It should be noted that the RMS of the incoming acoustic signal $s(t)$ is taken and it is then expressed in dB SPL (decibel Sound Pressure Level) which is equivalent to expressing the RMS of sound pressure in Pascal (Pa) as $20\ \log_{10}\frac{s_{RMS}}{20\ \mu\ Pa}$ [14].

Similarly, the RMS of the output force $F(t)$ is expressed as dB OFL (decibel Output Force Level), dB OFL can be obtained by expressing the RMS of the output force in Newton (N) as $20\ \log_{10}\frac{F_{RMS}}{1\ \mu\ N}$ [13]. The RMS of $F(t)$ is taken after the signal has reached steady state[14].

2.2 Digital Signal Processor (DSP)

The DSP as indicated in Figure 1 is a critical component of the BAHA. It functions as a multi-band dynamic range compressor in a BAHA. This is illustrated in Figure 2 [4].

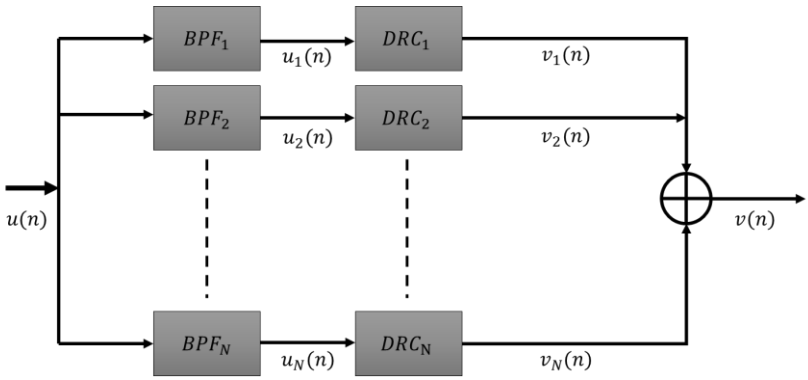


Figure 2: Block diagram of DSP where the incoming signal is split into N- frequency bands by band-pass filters, compressed by dynamic range compressors and then recombined [4].

2.2.1 Band Pass Filter (BPF)

It is seen from Figure 2 that the incoming signal $u(n)$ is spilt into N channels. The splitting of the signal into these channels is based on Band Pass Filters (BPF) which allows signals within a certain frequency range to pass through. The frequency response for an typical BPF is shown in Figure 3 [15].

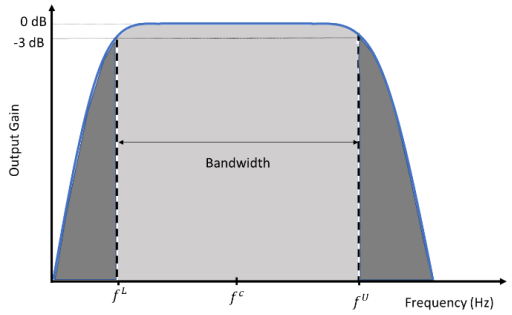


Figure 3: Frequency response of a typical bandpass filter [15].

There are three characteristic frequencies associated with a BPF, namely, *lower cut-off frequency*(f^L), *upper cut-off frequency* (f^U) and *centre frequency* (f^C). The lower and upper cut-off frequencies determine the *passband* (the frequencies that can pass through) of the filter. The span of this passband is the *bandwidth* of the filter and the frequencies that are outside this passband are attenuated. The *centre frequency* is defined as being halfway between f^L and f^U. The attenuated frequencies are known as the *stopband* of the filter. It is seen from Figure 3 that the transition between the passband and the stopband is not instantaneous. This is known as the *roll-off* of the filter and it is not present in the case of the ideal band pass filter [15].

The *N* BPF filters shown in Figure 2 has different lower and upper cut-off frequencies and therefore each filter passes a different frequency range. Thus, the incoming signal is split into *N* frequency bands[4].

Some applications where band-pass filters are implemented in a DSP include image processing, radar signal processing and telephone line modems [16].

2.2.2 Dynamic Range Compressor (DRC)

After the signal is divided into frequency bands, its dynamic range is reduced by the Dynamic Range Compressor (DRC). Dynamic range compression means that the effect of an increase in the input sound pressure level (SPL) will result in a lower (compressed) increase in the output SPL. This operation is illustrated in Figure 4 where the input SPL is plotted against the output SPL and the gain [4], [13].

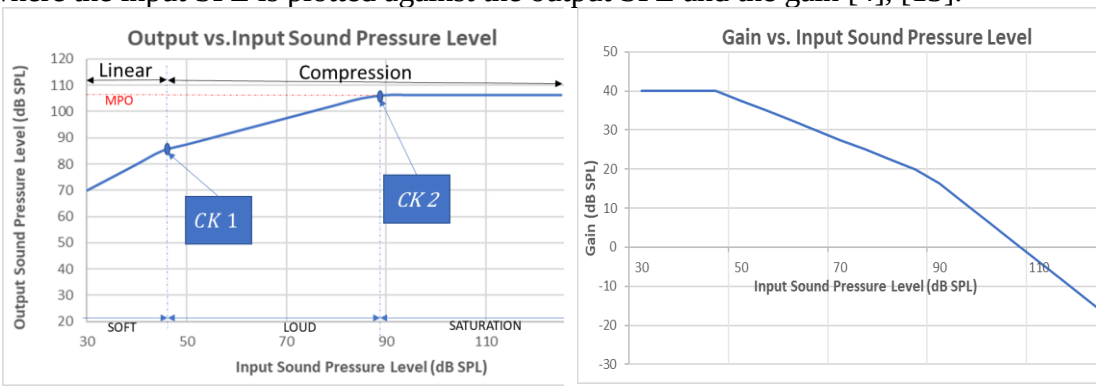


Figure 4: The effect of compression on the output SPL and on the gain [4], [13].

From these plots, gain can be defined as:

$$\text{Gain [dB]} = \text{ output SPL [dB SPL]} - \text{input SPL [dB SPL]}$$

The *compression ratio* is defined as $\frac{\Delta\, input}{\Delta\, output}$. Therefore, when this ratio is unity, no compression occurs, and this results in the linear region shown in the plot and the gain is constant. Any compression ratio larger than unity implies that compression is activated and that the dynamic range of the output SPL is reduced. A compression ratio of ∞ means that the output SPL will not increase further for any increase in the input SPL and this is known as *saturation*. The point at which compression is activated, i.e. where the compression ratio changes is known as the *compression knee-point* (CK). The output SPL when the input SPL is in the saturation region in point is known as the *maximum power output* (MPO). Compression is an important feature, since a person with hearing loss typically has normal hearing at higher input levels. Thus, a sound processor with no compression is uncomfortable at higher input. It should be noted that CK 1 is set as a function of the input sound level while CK 2 is set as a function of the MPO [4], [13].

Input SPL can be split into three regions as seen in Figure 4 and these regions are referred to as: (i) soft region – input SPL lower than the CK 1; (ii) loud region – input SPL between the CK 1 and CK 2; and (iii) saturation region – input SPL is sufficiently large to make the output SPL saturate [4], [13].
The DRC has other properties such as attack times and release times, which govern how much time it takes for the gain to change when the input SPL increases or decreases abruptly. These changes in gain should not occur instantaneously as this will cause distortions in the output sound's waveform [4].

2.3 The power gain and gain settings

The power gain for each frequency band at each input sound level is controlled by parameters known as *gain settings.* Two vectors, **p** and **g** can be used to store the power gain and the gain setting for each frequency band at each input sound. The structure of the **p** and **g** vectors are shown below:

$$\mathbf{p} = \begin{bmatrix} p_{1,1} \\ \vdots \\ p_{J,1} \\ p_{1,2} \\ \vdots \\ p_{J,2} \\ \vdots \\ \vdots \\ p_{1,S} \\ \vdots \\ p_{J,S} \end{bmatrix} \qquad \mathbf{g} = \begin{bmatrix} g_{1,1} \\ \vdots \\ g_{J,1} \\ g_{1,2} \\ \vdots \\ g_{J,2} \\ \vdots \\ \vdots \\ g_{1,S} \\ \vdots \\ g_{J,S} \end{bmatrix} \tag{2}$$

where, $p_{j,i}$ or $g_{j,i}$ is the gain setting or power gain corresponding to the j–th frequency band and i – th input sound level. J is the total number of frequency bands and S is the total number of input sound levels. The indexing of the elements of **p** and **g** vectors can be simplified by discarding information about the frequency band and the input sound level and simply re-indexing based on the position of the element in the vector. Note that only the indices are changed, and the elements of these vectors remain the same. The re-indexed **p** and **g** vectors are as shown:

$$\mathbf{p} = \begin{bmatrix} p_1 \\ \vdots \\ p_M \end{bmatrix} \qquad \mathbf{g} = \begin{bmatrix} g_1 \\ \vdots \\ g_M \end{bmatrix} \tag{3}$$

where, $M = J \times S$.

2.3.1 Constraints on p – vector

The **p** vector that is input into the DSP must obey certain constraints to be accepted by the DSP. These constraints and their motivations are as follows:

1. **Compression constraint** which specifies that $p(f_i, L_j) \geq p(f_i, L_{j+1})$ for $j = 1$. This means that when considering two gain settings for a given frequency band but for the "soft" and "loud" input SPL, the gain setting for the "soft" input SPL must have a higher value. The motivation for this constraint is to

ensure that sounds with a large amplitude are not amplified too much so that it may cause discomfort or further hearing impairment to the patient.

2. **Compression rule constraint** which states that $p(f_i, L_j) - p(f_i, L_{j+1}) \leq \alpha$ for $j = 1$ and $\alpha > 0$. This means that two gain settings of the same frequency band, but for "soft" and "loud" input SPL, can only differ by a given constant α at most. This constraint helps to avoid infinite compression of high amplitude sounds.
3. **Band Rule Constraint** which specifies that $|p(f_i, L_j) - p(f_{i\pm1}, L_j)| \leq \beta$. This which means that the difference between two gain settings of neighboring frequency bands of the same input SPL must be less than or equal to a given constant β. This constraint is enforced because if the difference in gain between neighboring frequency bands is too high, sound quality degradation can occur.
4. **Minimum / maximum value constraint** which specifies that $p(f_i, L_j) \in [p_{min}, p_{max}]$ and $p \in \mathbb{Z}$ .This means that the gain settings must be an integer and the possible values any gain setting can have must lie in a given range (regardless of its corresponding frequency band and input SPL region). If a gain setting p is set such that $p < p_{min}$, the output signal will be subject to noise in the measurement setup as this p results in a g that is too low and the output signal will be below the noise floor. On the other hand, if $p > p_{max}$, it will result in a g that is too high that it causes distortion in the output signal.

2.3.2 Transformation between p and g domains

The data for this thesis comes from measurements and each data point consists of pairs of values $(\mathbf{p}_n, \mathbf{g}_n)$ where $\mathbf{p}_n$ is a vector of input gain settings and $\mathbf{g}_n$ is a vector of power gains obtained during the measurement, where $n = 1, \dots, N$ and N is the total number of measurements.

The transformation from $\mathcal{P}$ domain to $\mathcal{G}$ domain is given by an unknown function $\mathbf{g}(\mathbf{p})$. This unknown function can be estimated using a model and this model is obtained by a machine learning technique called *linear regression*. The unknown function $\mathbf{g}(\mathbf{p})$ is estimated by linear regression using the measured data pairs $(\mathbf{p}_n, \mathbf{g}_n)$ and our model is $\mathbf{g}_n = \mathbf{g}(\mathbf{p}_n)$. However, a model cannot predict every $\mathbf{g}_n$ accurately and therefore the accuracy of the prediction must be considered using the residual $\mathbf{r_g}$, which is defined as:

$$\mathbf{r_g} = \hat{\mathbf{g}}_n - \mathbf{g}_n \tag{4}$$

where, $\hat{\mathbf{g}}_n$ is the gain predicted by the model.

Similarly, the transformation from $\mathcal{G}$ domain to $\mathcal{P}$ domain is given by another unknown function $\mathbf{p}(\mathbf{g})$**.** This transformation/ mapping is sometimes referred to as "the inverse mapping". It must be noted that $\mathbf{p} = \mathbf{p}(\mathbf{g})$ can be explicitly computed from the previously obtained model $\mathbf{g}(\mathbf{p})$ in some cases. Similar to the estimation of $\mathbf{g}(\mathbf{p})$, $\mathbf{p}(\mathbf{g})$ can also be estimated with linear regression using the model that $\mathbf{p}_n = \mathbf{p}(\mathbf{g}_n)$. The explicit computation of $\mathbf{p}(\mathbf{g})$ from $\mathbf{g}(\mathbf{p})$ will be compared with estimation of $\mathbf{p}(\mathbf{g})$ using linear regression in this report.

Once again, the accuracy of the predictions made by $\mathbf{p}(\mathbf{g})$ needs to be evaluated and the residual as defined below can be used for this:

$$\mathbf{r_p} = \hat{\mathbf{p}}_n - \mathbf{p}_n \tag{5}$$

It can be argued that $\mathbf{r_p}$ is not an ideal criterion to assess the ability of $\hat{\mathbf{p}}_n$ to produce the required power gain $\mathbf{g}$. This is because the solution to $\mathbf{p}(\mathbf{g})$ is not uniquely defined. In other words, there is possibly more than one $\mathbf{p}$ that can produce a given $\mathbf{g}$**.** Therefore, another criterion should be used to assess $\hat{\mathbf{p}}_n$. One such criterion is $\mathbf{r_g^*}$ where,

$$\mathbf{r_g^*} = \mathbf{g}(\hat{\mathbf{p}}_n) - \mathbf{g}_n \tag{6}$$

$\mathbf{r_g^*}$ evaluates the ability of the predicted parameter vector $\hat{\mathbf{p}}_{\boldsymbol{n}}$ to produce the required gain vector $\mathbf{g}_n$. It should be noted that the use of $\mathbf{r}_\mathrm{g}^*$ is somewhat flawed too as it is essentially using a model to evaluate the performance of another model and the validity of $\mathbf{r}_\mathrm{g}^*$ is only as good as the model $\mathbf{g}(\mathbf{p})$. A better approach to evaluate $\hat{\mathbf{p}}_\mathbf{n}$ would be to enter this predicted parameter vector into the BAHA and to observe the output gain. However, this approach is time consuming as more measurements will have to be performed.

3. Machine Learning

The term "Machine Learning" (ML) was coined by Arthur Samuel in 1959 and it was described as machines being able to learn without being programmed explicitly [17].

Machine learning can be split into four categories namely: *supervised* learning, *unsupervised* learning, *semi-supervised* learning and *reinforcement* learning. Supervised learning is used when labelled data is available and unsupervised learning is used when the available data is unlabelled. The labelling of the data is done by the s*upervisor* and humans usually assume this role. Semi-supervised learning deals with a mix of labelled and unlabelled data. Reinforcement learning is somewhat different to the aforementioned learning. An action is performed by an $agent$ and this action is rewarded or penalised, based on the outcome of the action towards achieving the goal task that reinforcement learning is being used for [18].

As already mentioned, the type of machine learning being used in this thesis is linear regression, which is a subset of supervised learning. Linear regression will be used to create predictive models. These models are developed through *training* where the ML algorithm attempts to find the minimum of a *loss function.* The loss function is a function that takes the model prediction as the input and evaluates the error of that prediction by comparing it to the actual values in the data. The model is then tuned in such a way to produce a lower output from the loss function[19].

A commonly used loss function is the mean squared residual (MSR). Given a matrix of model predictions $\widehat{\mathbf{Y}}$ and a matrix of the true values from the data $\mathbf{Y}$. The MSR can be expressed as:

$$MSR = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(\hat{y}_{i,j} - y_{i,j}\right)^2 \tag{7}$$

where $\hat{y}_{i,j}$ and $y_{i,j}$ are elements of $\widehat{\mathbf{Y}}$ and $\mathbf{Y}$ respectively and $i = 1, \dots, N$ and $j = 1, \dots, M$. In addition to being used as the loss function, the MSR can also be used as a metric to assess the prediction accuracy of a model [19].

3.1 Linear Regression

In the context of machine learning, the elements of $\mathbf{g}$ and $\mathbf{p}$ are known as the *dependant* variables and *independent* variables, respectively. Linear regression models attempt to express the dependant variable as a linear combination of the independent variables [19].

3.1.1 Linear regression for linear model

The first linear regression model to be presented is the linear model and it is defined as follows:

$$\mathbf{g} = \mathbf{W}^{\mathbf{T}}\mathbf{p} \tag{8}$$

where $\mathbf{W}$ is a matrix known as the weight matrix and it dictates the effect or the "weight" of each of the parameters on the gain.

When given N measurements, which result in one data pair $(\mathbf{g_n}, \mathbf{p_n})$ for each measurement, the model can be re-written as:

$$\mathbf{g_n}=\mathbf{W^T p_n} \tag{9}$$

where, $n = 1 \ldots . N$. This model is generalised for N measurements by transposing and stacking $\mathbf{p}_n$ and $\mathbf{g}_n$ vectors as shown below:

$$\begin{bmatrix} \leftarrow \mathbf{g_1^T} \rightarrow \\ \vdots \\ \leftarrow \mathbf{g_N^T} \rightarrow \end{bmatrix} = \begin{bmatrix} \leftarrow \mathbf{p_1^T} \rightarrow \\ \vdots \\ \leftarrow \mathbf{p_N^T} \rightarrow \end{bmatrix} \begin{bmatrix} \mathrm{W}_{1,1} & \cdots & \mathrm{W}_{1,M} \\ \vdots & \ddots & \vdots \\ \mathrm{W}_{M,1} & \cdots & \mathrm{W}_{M,M} \end{bmatrix} \tag{10}$$

This generalised model can now be expressed as:

$$\mathbf{G} = \mathbf{PW} \tag{11}$$

The matrix $\mathbf{W}$ must be solved for in order to use this model and this can be done using the *normal equations*, derived below.

In order to derive the normal equations, one must begin at the loss function for linear regression, which is the least squares objective function, defined as [19], [21]:

$$L(\mathbf{W}) = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (\mathrm{G}_{i,j} - (\mathrm{P}_{i,1}\mathrm{W}_{1,j} + \cdots + \mathrm{P}_{i,M}\mathrm{W}_{M,j}))^2 \tag{12}$$

Note that $L(\mathbf{W})$ is the MSR, which was previously defined in equation 7.
The optimal $\mathbf{W}$ in equation 12 will minimise $L(\mathbf{W})$ and it is found by setting the partial derivative of $L_j(\mathbf{W})$ with respect to each element of $\mathbf{W}$ to zero[17]. $L_j(\mathbf{W})$ is $L(\mathbf{W})$ for j-th column of $\mathbf{W}$, where $j = 1, \ldots M$. Thus, $L_j(\mathbf{W})$ can be expressed as [19]:

$$L_j(\mathbf{W}) = \frac{1}{N \times M} \sum_{i=1}^{N} (\mathrm{G}_{i,j} - (\mathrm{P}_{i,1}\mathrm{W}_{1,j} + \cdots + \mathrm{P}_{i,M}\mathrm{W}_{M,j}))^2 \tag{13}$$

The partial derivative of $L_j(\mathbf{W})$ with respect to each element of $\mathbf{W}$ is put into a matrix $\mathbf{J}$ which has the same dimensions as $\mathbf{W}$. $\mathbf{J}$ is shown below [21]:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial L_1(\mathbf{W})}{\partial \mathrm{W}_{1,1}} & \cdots & \frac{\partial L_M(\mathbf{W})}{\partial \mathrm{W}_{1,M}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L_1(\mathbf{W})}{\partial \mathrm{W}_{M,1}} & \cdots & \frac{\partial L_M(\mathbf{W})}{\partial \mathrm{W}_{M,\mathrm{M}}} \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \tag{14}$$

Let's examine an arbitrary element $\mathbf{J}_{k,j}$ where $k = 1, \ldots, M$ and $j = 1, \ldots, M$.

$$\mathbf{J}_{k,j} = \frac{\partial L_j(\mathbf{W})}{\partial \mathrm{W}_{k,j}} = \frac{1}{N \times Q} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} 2(\mathrm{G}_{\boldsymbol{i,j}} - (\mathrm{P}_{\boldsymbol{i,1}}\mathrm{W}_{1,j} + \cdots + \mathrm{P}_{\boldsymbol{i},M}\mathrm{W}_{M,j}))(-\mathrm{P}_{\boldsymbol{i},k}) = 0 \tag{15}$$

This can be rearranged as[21]:

$$\sum_{i=1}^{N} \mathrm{G}_{i,j}\mathrm{P}_{i,k} = \sum_{i=1}^{N} (\mathrm{P}_{i,1}\mathrm{W}_{1,j} + \cdots + \mathrm{P}_{i,M}\mathrm{W}_{M,j}))(\mathrm{P}_{i,k}) \tag{16}$$

This left-hand side and the right-hand side of this equation can be stacked into two matrices of the same dimensions as **J** as shown [21]:

$$\begin{bmatrix} \sum_{i=1}^{N} G_{i,1}P_{i,1} & \cdots & \sum_{i=1}^{N} G_{i,M}P_{i,1} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} G_{i,1}P_{i,M} & \cdots & \sum_{i=1}^{N} G_{i,M}P_{i,M} \end{bmatrix} =$$

$$\begin{bmatrix} \sum_{i=1}^{N} (P_{i,1}W_{1,1} + \cdots + P_{i,M}W_{M,1})P_{i,1} & \cdots & \sum_{i=1}^{N} (P_{i,1}W_{1,M} + \cdots + P_{i,M}W_{M,M})P_{i,1} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} (P_{i,1}W_{1,1} + \cdots + P_{i,M}W_{M,1})P_{i,M} & \cdots & \sum_{i=1}^{N} (P_{i,1}W_{1,Q} + \cdots + P_{i,M}W_{M,M})P_{i,M} \end{bmatrix} \quad (17)$$

The equations within these matrices can be condensed using matrix notation as shown [21]:

$$\begin{bmatrix} \mathbf{p}_1^{\mathbf{T}}\mathbf{G} \\ \vdots \\ \mathbf{p}_M^{\mathbf{T}}\mathbf{G} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1^{\mathbf{T}}\mathbf{PW} \\ \vdots \\ \mathbf{p}_M^{\mathbf{T}}\mathbf{PW} \end{bmatrix} \quad (18)$$

$$\mathbf{P^T G = P^T PW} \quad (19)$$

The weight matrix **W** can finally be obtained by solving the following [19], [21]:

$$\mathbf{W} = [\mathbf{P^T P}]^{-1}\mathbf{P^T G} \quad (20)$$

The term $\mathbf{[P^T P]^{-1}P^T}$ is known as the pseudo-inverse of **P** and is denoted as $\mathbf{P}^+$ [22]. If the number of rows in **P** is greater than the number of columns, the system is said to be *over-determined* (more equations than unknowns) and a unique solution can be obtained provided that $\mathbf{P}^+$ is full rank. On the other hand, if the number of columns in **P** is greater than the number of rows, the system is said to be *underdetermined* (more unknowns than equations), a unique solution does not exist[23].

A model where **W** is obtained using this method is called an ordinary least squares (OLS) model as only the least squares criterion was used to evaluate **W** [19].
As previously discussed, residuals are required to gauge the accuracy of the predictions made by a particular model. The residual in this case is the defined as:

$$\mathbf{R_g = PW - G} \quad (21)$$

where $\mathbf{R_g}$ is the residual matrix.

3.1.2 Affine model

It can be seen that the linear model only represents each gain g as a linear combination of **p**. This model doesn't accommodate any bias in g that is not dependant on **p.** This problem can be mitigated by adding a bias term into the linear combination of **p** that results in the prediction of **g**.
This model with the bias term is called an affine model and it is defined as:

$$\mathbf{g} = \mathbf{W}^{\mathbf{T}}\mathbf{p} + \mathbf{b} \tag{22}$$

where, **b** is the bias vector [19]. The bias vector **b** can be incorporated into the weight matrix **W** and the affine model can still be expressed as:

$$\mathbf{G} = \mathbf{PW} \tag{23}$$

The structure of **P** and **W** must be changed slightly to accommodate these changes as shown below:

$$\begin{bmatrix} \leftarrow \mathbf{g}_1^{\mathbf{T}} \rightarrow \\ \vdots \\ \leftarrow \mathbf{g}_N^{\mathbf{T}} \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & \leftarrow \mathbf{p}_1^{\mathbf{T}} \rightarrow \\ \vdots & \vdots \\ 1 & \leftarrow \mathbf{p}_N^{\mathbf{T}} \rightarrow \end{bmatrix} \begin{bmatrix} W_{0,1} & \cdots & W_{0,M} \\ \vdots & \ddots & \vdots \\ W_{M,1} & \cdots & W_{M,M} \end{bmatrix} \tag{24}$$

It can be seen that the modified **W** has an additional row at the top and this row is $\mathbf{b}^{\mathbf{T}}$. A column of ones must be added to **P** to allow the change to **W**[19], [21]**.** The modified **P** matrix is known as the *design* matrix.

This affine model is derived from a modified version of the least squares objective function $L(\mathbf{W})$ with an additional term $W_{0,j}$ as shown below [19], [21]:

$$L(\mathbf{W}) = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (G_{i,j} - (W_{0,j} + P_{i,1} W_{1,j} + \cdots + P_{i,M} W_{M,j}))^2 \tag{25}$$

This system can also be solved as previously shown in equation 20 [19], [21].

3.1.3 Higher-order models

It is seen from previously discussed models that the basis functions only consisted of the **p** vector and a constant. Therefore, all the functions in the basis were linear. Further descriptive power can be added to this approach by adding additional functions to the basis and this is known as *basis expansion.* These additional basis functions are non – linear and have an exponent higher than 1. This approach of adding to non-linear functions to the basis is called *polynomial regression.* Some examples of these basis functions are: p_1^2, p_1^3 and $p_4 p_1^2 p_2^3$ and so on. The highest exponent in the basis is called the order of the polynomial regression model. This new model can be expressed as:

$$\mathbf{G} = \mathbf{BW} \tag{26}$$

Where **B** is the modified design matrix and **W** is the corresponding weight matrix, which can be solved for as shown in equation 20. It must be noted that polynomial regression is considered a subclass of linear regression because the model is still linear with respect to its weights [19].

In order to understand the structure of **B,** it is useful to consider applying a 3rd order polynomial regression model to the simple case when only two parameters are present in the **p** vector, such that

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{27}$$

In this case, the **B** matrix would look like this:

$$\mathbf{B} = \begin{bmatrix} 1 & p_{1_{(1)}} & p_{2_{(1)}} & p_{1_{(1)}}^2 & p_{2_{(1)}}^2 & p_{2_{(1)}}p_{1_{(1)}}^2 & p_{1_{(1)}}p_{2_{(1)}}^2 & p_{1_{(1)}}^3 & p_{2_{(1)}}^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & p_{1_{(N)}} & p_{2_{(N)}} & p_{1_{(N)}}^2 & p_{2_{(N)}}^2 & p_{2_{(N)}}p_{1_{(N)}}^2 & p_{1_{(N)}}p_{2_{(N)}}^2 & p_{1_{(N)}}^3 & p_{2_{(N)}}^3 \end{bmatrix} \tag{28}$$

where, $p_{1_{(n)}}$ is the n – th measurement for p_1 where, $n = 1, \ldots, N$.
This is a rather trivial example with only two parameters in **p** and a relatively low order (3rd order) polynomial regression model. However, *the curse of dimensionality* becomes obvious in this example as more columns will be added to the design matrix **B** as the number of dimensions of **p** is increased. This effect is compounded when the order of the polynomial regression model is increased. The issue with the number of columns increasing in the design matrix is that the system will become underdetermined if the number of columns of the design matrix is greater than the number of rows, unless more rows are added to the design matrix by adding more data [24].

In order to put this problem into perspective, let's examine the number of data points required for the system to be not underdetermined, for different order polynomial regression models, when an 18 dimensional **p** vector is involved. The highest number of dimensions for the **p** vector in this thesis is 18. Each data point consists of a measurement pair $(\mathbf{g}_n, \mathbf{p}_n)$. Assume that all these models also have the constant function in the basis.

Polynomial Order	Number of data points required
1	19
2	190
3	1330
4	7315
5	33649

Table 1: The number of data points required for the system of linear equations to be not underdetermined for varying orders of polynomial regression models.

It is seen from Table 1 that the amount of data required increases rapidly as the polynomial regression order increases due to the high dimensionality of the data. The general formula for the required number of points for a model of given degree including a bias term is given by:

$$\#\, required\ points = 1 + \sum_{i=1}^{D} \frac{(x+i-1)!}{i!\,(x-1)!}$$

where x is the length of the **p** vector and D is the order of the polynomial. The `sci-kit learn` package in Python, which was used for this thesis governs the number of columns in the design matrix according this formula.

3.2 Training, validation and testing data

The purpose of a machine learning dataset can be broadly categorized into training, testing and validation.

3.2.1 Training data

Training in the context of machine learning means, the process from which the model obtains its weights by minimizing the loss function. In the case of linear regression, this involves populating the elements of the **W** matrix. The portion of the available dataset that is allocated to the training of the model's weights is called training data. This is often the largest portion of the available dataset and the training data is picked randomly to avoid bias [19],[24].

3.2.2 Testing data

Once the model is trained, it is essential to quantify the performance of the model on unseen data. This is to give an indication of the accuracy of the model in reality when making predictions given some input. For example, to predict the parameter vector **p** that would give a particular gain vector **g**. In order to get an accurate assessment, it is therefore crucial that no part of the training data is present in the testing data [19],[24].

3.2.3 Validation data

In order to understand the role of the validation data, the concept of *overfitting* must be discussed. Overfitting is a phenomenon that occurs when the model learns information that is too specific to the training data such as measurement noise. This results in the model being *fitted* too closely to the training data instead of learning only the general trends. Overfitting results in a very small residual on the training data, however, the model will fail to generalize well on unseen data [17],[22].

The role of the validation data can be described as somewhat intermediate to training and testing data. It is used as a proxy for test data during training. The validation data is used to prevent overfitting as the ability of the model to generalise well to new data is measured using the prediction accuracy of the model on the validation data. Using this method of testing the accuracy of the model on validation prevents an overly optimistic view on the performance of the model. The performance of a particular model on the validation data can be used as a criteria for model selection [19], [24].

Validation data is also used to tune certain model parameters known as *hyper-parameters*. The values of these parameters usually cannot be directly calculated from the data but determined from heuristic methods such as trial and error. The model is trained with certain values for certain hyperparameters and then the prediction accuracy of the model on the validation dataset is calculated to gauge the suitability of these specific hyper-parameter values. This process is repeated to find the most suitable hyper-parameter values [19], [24].

It is well established that the most important use of the available dataset should be to train the model and therefore only the minimum required amount of data should be reserved for testing and validation. A validation strategy known as *k-fold validation* can be used to maximise the available amount of training data and avoid reserving a proportion of data only for validation [19], [24].

3.2.3.1 K-fold validation

The main benefit of k-fold validation is that it takes a subset of training data as validation data in such a way that this subset can also be used for training. The general procedure for k-fold validation is outlined below:

1. Divide the training data into *k* number of *folds* or subsections.
2. Train the model on $k - 1$ folds.
3. Validate the model on the remaining fold and note the accuracy.
4. Repeat this process until all the folds have been used as training and validation folds.
5. Calculate the overall validation accuracy by averaging the validation accuracy from that of every fold.
6. Repeat the above steps for different hyper-parameter values.

Finally, the model is then trained on all the training data using the most appropriate hyperparameter values chosen through k-fold validation. Training and validating in this manner ensure that all the training data is used for training. Despite this method maximising the available training data, it is computationally expensive as the model is trained and validated k – times for each hyper-parameter value. A commonly used value used for k is 10 [24].

3.3 Regularization

As previously discussed, overfitting occurs when the model learns information that is too specific to the training data, this typically occurs in a linear regression model when the chosen model is too complex to represent the general trend of the data. The complexity of a linear regression model is dictated by its order and overfitting can be mitigated by reducing the order of the model[19].

Another method of controlling the model complexity to reduce the likelihood of overfitting is known as *regularization.* In the context of linear regression, this involves placing a penalty on the magnitude of the weights and the strength of this penalty can be controlled by the regularization parameter λ, which is a hyper-parameter. This penalty is imposed by modifying the loss function presented in equation 12, where an additional term known as the regularization term is added to the loss function. The regularization term is often chosen to be either the L_2 norm or the L_1 norm of the weights[19].

To illustrate the effects of regularization, the MSR of training and validation accuracy of a regularized model is presented as a function of λ in Figure 5.

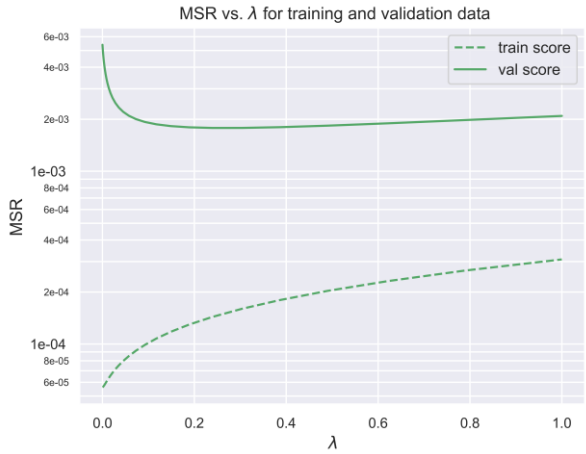


Figure 5: MSR of training and validation data plotted as a function of λ.

It is seen that a lower MSR value is found at an optimal non-zero value of λ and the training MSR increases as λ is increased. This demonstrates that this model was overfitted to the training data at $\lambda = 0$ and that regularization by increasing λ, mitigated this problem and reduced the MSR of the validation data.

3.3.1 L_2 Regularization

Linear regression with L_2 regularization or *ridge* regression uses the L_2 norm to impose a penalty on the magnitude of the weights. A geometric interpretation of ridge regression during training using two weights is shown in Figure 6 [19].

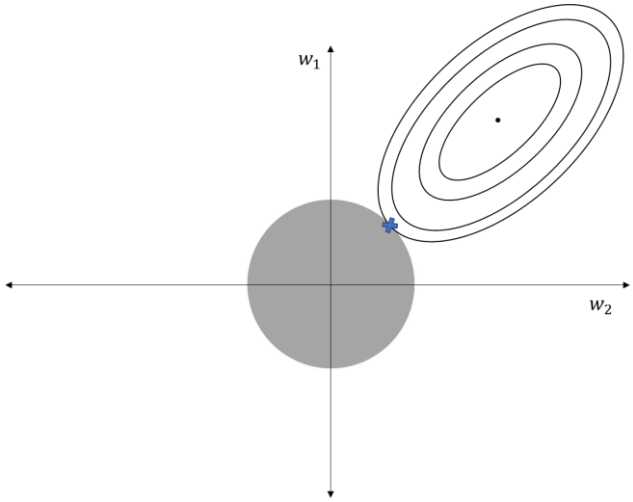


Figure 6: Geometric interpretation of ridge regression using two weights. The ellipses indicate the contours of the OLS MSR, and the grey shaded region indicates the possible combinations of w_1 and w_2. The black dot represents the weights selection made by the OLS criteria alone and the blue cross indicates the weights chosen using ridge regression with the addition of the L_2 norm constraint [19]..

The black dot in this figure indicates the optimal values for w_1 and w_2 according to OLS model, which is determined by the least squares criterion. The ellipses represent the contours produced by the output of the OLS loss function, which is the MSR. On a given ellipse, the MSR is the same for any combination of w_1 and w_2 that lie on that ellipse and the MSR increases for the ellipses that are further away from the black dot. The grey circle on the plot represents the "budget" available for the weights, this budget is governed by the L_2 norm of the weights for ridge regression. The diameter of this circle is inversely proportional to the λ that is chosen. Thus, a large λ will

make the weights budget small and therefore the available magnitudes for the weights will be more constrained. The blue cross on the figure represents the chosen value for the weights, which satisfy both the least squares and the budget criteria. The visual representation presented in Figure 6 holds true in higher dimensions for models with more weights [19].

The OLS loss function needs to be modified to implement ridge regression and this is done with the addition of the regularization term, which is the weight budget. The modified loss function can be seen below:

$$L(\mathbf{W}) = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (\mathrm{G}_{i,j} - (\mathrm{P}_{i,1}\mathrm{W}_{1,j} + \cdots + \mathrm{P}_{i,M}\mathrm{W}_{M,j}))^2 + \lambda \left(\sqrt{\mathrm{W}_{1,j}^2 + \cdots \mathrm{W}_{M,j}^2} \right)^{\mathbf{2}} \quad (29)$$

It is possible to formulate this problem into a system of linear equations which can be solved for **W**. The above equation can be condensed into matrix notation as it was done in section 3.1.1 with the linear model. The condensed equation is expressed as:

$$\frac{1}{N \times M} [\, \mathbf{P^T G}] = \left[\frac{1}{N \times M} [\mathbf{P^T P}] \; + \lambda \boldsymbol{I} \right] \mathbf{W} \quad (30)$$

This equation can then be solved for **W** as follows:

$$\mathbf{W} = \left[\frac{1}{N \times M} \, [\mathbf{P^T P}] + \lambda \boldsymbol{I} \right]^{\mathbf{-1}} \left[\frac{1}{N \times M} \, [\, \mathbf{P^T G}] \right] \quad (31)$$

If the model contains a bias term, the **P** matrix is modified as a column of ones is inserted as previously shown in equation 24. In addition, equation 30 is modified slightly as follows:

$$\frac{1}{N \times M} [\mathbf{P^T G}] = \left[\frac{1}{N \times M} [\mathbf{P^T P}] + \lambda \tilde{\boldsymbol{I}} \right] \mathbf{W} \quad (32)$$

where $\tilde{I}$ is the identity matrix where the element on the intersection of the first row and the first column is zero. This modification has the effect of not penalising the magnitude of the bias term. The bias term is left unpenalized as doing so means that a simple offset in the dependant variable would not result in a similar shift in the predictions of the model as the penalised bias term will not allow it [19].

3.3.2 L_1 Regularization

Linear regression with L_1 regularization or LASSO (Least Absolute Shrinkage and Selection Operator) regression uses the L_1 norm of the weights as the regularization term to control the magnitude of the weights. A geometric interpretation of LASSO regression is shown in Figure 7 [19].

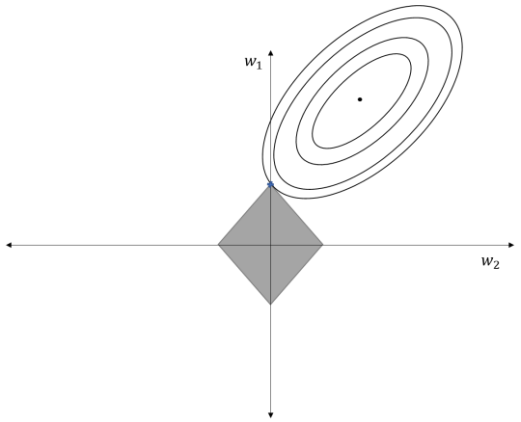


Figure 7: Geometric interpretation of LASSO regression using two weights. The ellipses indicate the contours of the OLS MSR, and the grey shaded region indicates the possible combinations of w_1 and w_2. The black dot represents the weights selection made by the OLS criteria alone and the blue cross indicates the weights chosen using LASSO regression with the addition of the L_1 norm constraint [19].

In contrast to the constrained region of weights for ridge regression, it is seen that the weights region is diamond shaped as the L_1norm is used. This diamond shaped region has pointed corners where one of the weights is exactly zero. If the solution for weights occurs on these corners, one of the coefficients will be set to zero. There are more of these pointed regions in higher dimensions and therefore there is a high chance that an increasing number of the weights are going to be set to zero. This is beneficial as it encourages feature selection where the independent variables that doesn't have a large effect on the dependant variable are eliminated from the model. In addition, a model with a high number of zero weights has the advantage of being computationally efficient when it comes to making predictions [19].

LASSO regression is implemented by modifying the OLS loss function. This loss function to be minimised is shown below:

$$L(\mathbf{W}) = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (\mathrm{G}_{i,j} - (\mathrm{P}_{i,1}\mathrm{W}_{1,j} + +\mathrm{P}_{i,M}\mathrm{W}_{M,j}))^2 + \lambda\left(\left|\mathrm{W}_{1,j}\right| + .. + \left|\mathrm{W}_{M,\mathrm{j}}\right|\right) \quad (33)$$

Unlike ridge regression, the lasso regression loss function cannot be formulated into a set of linear equations, so the solution for **W** is obtained by solving a minimization problem[19]. In the case of this thesis, an algorithm called *coordinate descent* was used which is already implemented in the `sci-kit learn` package for Python.

3.3.3 Choosing the optimal value for λ

The optimal value for λ will result in a model that is not overfitted while preventing *underfitting*. Underfitting occurs when the model's learning is restricted too much and it doesn't learn enough from training data. If the linear regression model chosen for given data requires regularization and the chosen value for λ is too low overfitting will occur as the magnitude of the weights are not being penalised enough. However, on the other hand, if the chosen value of λ is too large, then the magnitude of the weights is being restricted too much and underfitting will occur. Therefore, it can be appreciated that choosing the optimal value of λ is crucial to the performance of a regularised model[19].

A common approach to choosing a value for λ is to train the model with different values of λ and then observe the accuracy of the predictions made on the validation

data by the trained model. This process can be repeated for many arbitrary values of λ and the corresponding **W** that produces the lowest MSR can be used as the optimal model that does not overfit.

However, this approach of arbitrarily varying λ rarely gives the most optimal value for λ. A more refined method would be to use a minimisation algorithm such as the *golden section search* which finds the minimum of a function by varying a function input between a given parameter interval. In this case, the MSR of the trained model's prediction on the validation data is minimised with respect to λ, where λ is varied in the interval $[\lambda_{min}, \lambda_{max}]$ [25].

The values for λ_{min} and λ_{max} were found by trial and error and then golden section search was used to find the optimal λ. The golden section search algorithm was already implemented in the `scipy` library available for Python.

3.4 Scaling and basis transformation

The input data for the model must be scaled so that all the elements of the design matrix and the dependant variable matrix are on the same scale of magnitude. The scaling of the columns for design matrix is important for both OLS and regularized models.
For OLS, this can be understood through an example, consider the values in the design matrix column that correspond to the basis function p_1^3, the magnitudes of these values can orders of magnitudes larger or smaller than that present in p_1 column. This can result in $ill - conditioning$ in the regression problem. An ill-conditioned problem is one where a small change in the input results in a large change in the output. Solving such problems using computers cause inaccuracies in the solution due to issues with numerical precision. In the case of linear regression, this leads to a poor solution for **W** [26].

For regularized models, in addition to conditioning of the design matrix, scaling poses additional problems. If the design matrix has values in varying orders of magnitude, the associated weights will also be present in varying orders of magnitude. This means that weights that are large, simply due to scaling will be penalised under regularization [19].

An arbitrary array **X** can be scaled to produce the scaled array $\mathbf{X_{sc}}$, which contains elements that lie in the interval $[a, b]$. $\mathbf{X_{sc}}$ is obtained as follows:

$$\mathbf{X_{sc}} = \left[(b-a)\tilde{1} * (\mathbf{X} - 1\,\mathbf{x_{min}}) * \frac{1}{1(\mathbf{x_{max}} - \mathbf{x_{min}})}\right] + a\tilde{1} \qquad (34)$$

where, "*" indicates elementwise multiplication of matrices, and the fraction indicates elementwise division, $\mathbf{x_{max}}$ and $\mathbf{x_{min}}$ are row vectors that contain column-wise maximum and column-wise minimum of the unscaled array **X**. The number of elements in $\mathbf{x_{max}}$ and $\mathbf{x_{min}}$ are the same as the number of columns present in **X**. 1 is a column vector of ones with the same number of rows as **X** , whereas, $\tilde{1}$ is a matrix of ones with the same dimensions as **X** [27]**.**

It must be noted that when scaling the test data using the above equation that the column – wise maximum and minimum values are taken from the training data. This is to prevent *data leakage*, where information about test data's distribution is learned

by the model. Data leakage can cause the model to perform better on the test data and therefore cause overly optimistic estimate of the model's performance [28].

In addition to scaling the design matrix, its elements are also transformed using *Legendre polynomial basis.* The effect of this transformation is that it changes the basis to be *orthonormal*, where orthonormal basis functions have the property of being linearly independent to each other. Transforming the basis to an orthonormal basis has the effect of improving the conditioning of the least squares problem as the initial basis, which simply consists of powers of $\mathbf{p}$ will results in an ill-conditioned problem. In addition, some *multi-collinearity*, will start to occur as the order of the model order is increased. Multi – collinearity is where the value of one independent variable in the basis, depends on another. An example of this is the value of p_1^2 as an "independent" variable depends on the value of p_1, which is another independent variable. Multi-collinearity can lead to an ill-conditioned system and there will be considerable errors in the solution for $\mathbf{W}$. The linear independence of the orthonormal basis helps to mitigate multi-collinearity[26][29].

The values of an array scaled according to equation 34 can be restored by applying its inverse scaling using the formula given below:

$$\mathbf{X} = \left[(\mathbf{X_{sc}} - a\tilde{\mathbb{1}}) * (\mathbb{1}(\mathbf{x_{max}} - \mathbf{x_{min}})) * \frac{1}{(b-a)\tilde{\mathbb{1}}} \right] + \mathbb{1}\,\mathbf{x_{min}} \tag{35}$$

Note that this is a simple algebraic rearrangement of equation 34.

The inverse scaling needs to be applied to the predictions made by models where both the dependant and independent variables were scaled. This is because the predictions made by the model will be in the same magnitude as the scaled dependant variables, so in order for the model predictions to make sense, the inverse scaling must be applied.

3.5 Pipeline

Now that all the individual steps for a generalised machine learning approach using linear regression is outlined, the exact methodology used in this thesis can be summarised using the steps given below:

1. The rows of the $\mathbf{P}$ and $\mathbf{G}$ matrices are split into training and testing sets to result in four new matrices: $\mathbf{P}_{tr}$, $\mathbf{G}_{tr}$, $\mathbf{P}_{ts}$ and $\mathbf{G}_{ts}$. 80 % of the data is used for training and 20 % of the data is used for testing.
2. Column-wise scaling is applied to $\mathbf{P}_{tr}$ and $\mathbf{G}_{tr}$ to be in the range of $[-1,1]$ according to equation 34. The scaled training matrices are denoted as $\overline{\overline{\mathbf{P}_{tr}}}$ and $\overline{\overline{\mathbf{G}_{tr}}}$.
3. Column-wise scaling is applied to $\mathbf{P}_{ts}$ and $\mathbf{G}_{ts}$ and the range is specified to be $[-1,1]$, but the column-wise minimum$(\mathbf{x}_{\min})$ and maximum $(\mathbf{x}_{\max})$ used in the scaling process are taken from the training data. This means that all the elements of the scaled test matrices might not exactly lie in the range $[-1,1]$. The scaled test matrices are denoted as $\overline{\overline{\mathbf{P}_{ts}}}$ and $\overline{\overline{\mathbf{G}_{ts}}}$.
4. $\overline{\overline{\mathbf{P}_{tr}}}$ is transformed into Legendre polynomial basis and the weight matrix $\mathbf{W}$ in the linear regression model $\mathbf{PW} = \mathbf{G}$ is calculated by feeding the training data into the model. This obtains the mapping $\mathcal{P} \mapsto \mathcal{G}$. Different models of different

degrees and with/without different regularisation techniques outlined previously are trained here. No validation is performed for unregularized models as there are no hyperparameters to tune. For regularized models, k-fold validation is used to choose the optimal λ. While training regularised models, unscaled training data $\mathbf{P}_{tr}$ and $\mathbf{G}_{tr}$ are fed in. During k-fold validation, different training and validation folds are constructed from the unscaled data and then the training and validation folds are scaled to be within $[-1,1]$. However, during the scaling of the validation fold, $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ are obtained from the training fold. The reason why the scaling of the validation data is done in this fashion is to avoid learning about the probability distribution of the validation data. Therefore, the validation data is treated similarly to the testing data in this regard. Meanwhile, the scaled training data is fed into the model for the training of the unregularized models as there is no validation step involved. Legendre polynomial basis transformation is also done on the scaled **P** matrix during k-fold validation.

5. The weight matrix **W** obtained during training is used with the scaled test data matrix $\overline{\overline{\mathbf{P}_{ts}}}$ to make predictions. These predictions are then scaled using the inverse scaling equation shown in equation 35 and then the residual $\mathbf{R}_{\mathrm{g}}$ is evaluated using $\mathbf{G}_{\mathrm{ts}}$ according to equation 36.
6. In order obtain the mapping $\mathcal{G} \mapsto \mathcal{P}$, the weight matrix $\widetilde{\mathbf{W}}$ is obtained from the linear regression model $\mathbf{G}\widetilde{\mathbf{W}} = \mathbf{P}$ by feeding the training data. The exact same procedure outlined in step 4 is used in this case too.
7. The residual $\mathbf{R}_{\mathrm{p}}$ is then calculated using equation 37. $\widetilde{\mathbf{W}}$ and the scaled test data matrix $\overline{\overline{\mathbf{G}_{ts}}}$ are used to make predictions which are then scaled using the inverse scaling formula and $\mathbf{R}_{\mathrm{p}}$ is evaluated using $\mathbf{P}_{ts}$**.**
8. The predicted values are denoted as $\widehat{\mathbf{P}_{ts}}$ which is used with a weight matrix **W** to evaluate the residual $\mathbf{R}_{\mathrm{g}}^{*}$ according to equation 38. This residual evaluates the ability of a particular predicted **p** vector to produce the required **g** vector.

A visual representation of the aforementioned machine learning approach can be seen in Figure 8.

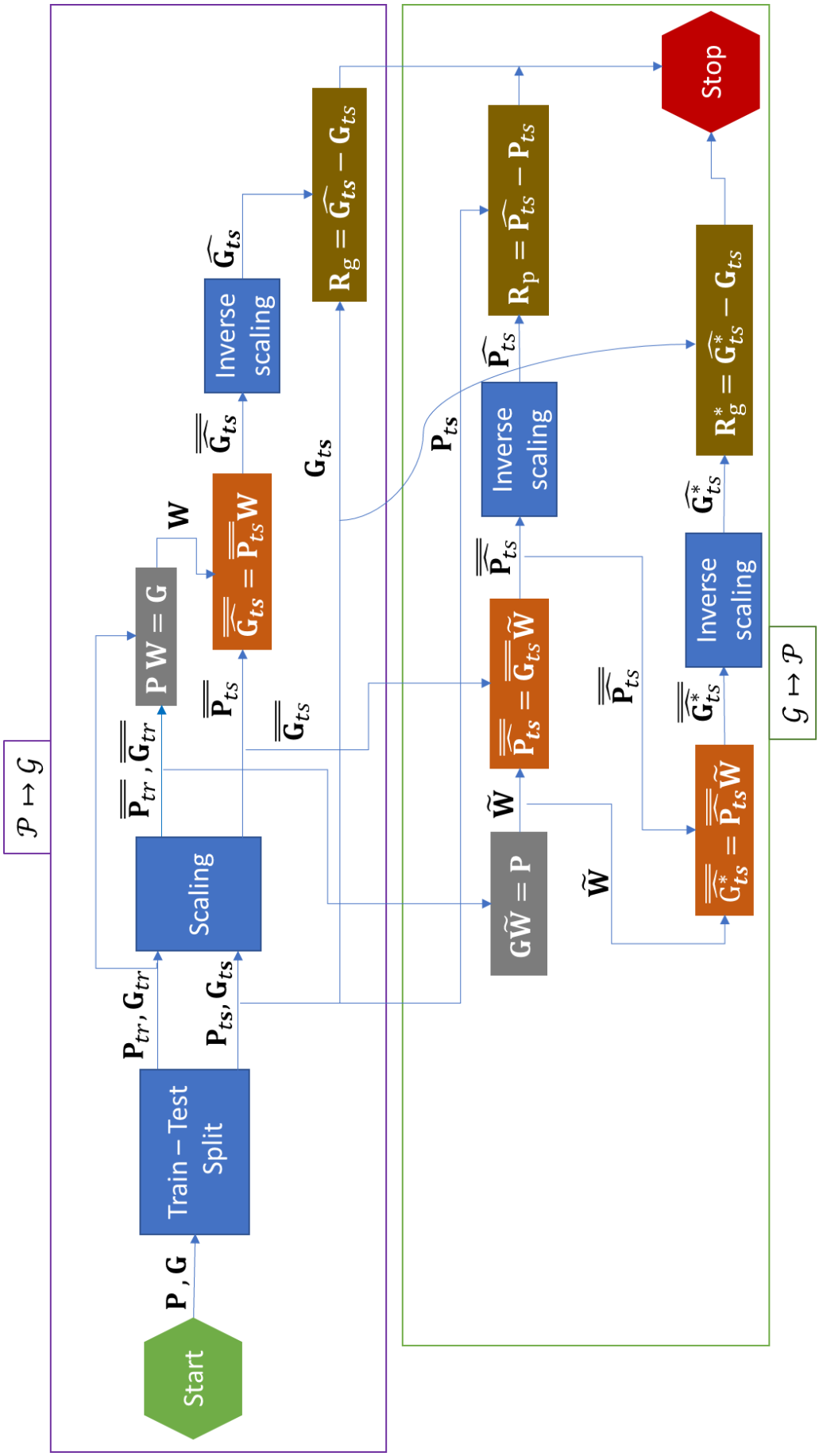


Figure 8: Flowchart illustrating the machine learning approach undertaken in this thesis.

4. Results

This section will present and discuss the results obtained by performing this thesis work.

4.1 Measurement campaign

The data that was used in this thesis comes from measurements performed by CBAS AB. These measurements were automated and performed in a controlled environment. The measurement process involves measuring the response of the transducer to the sound produced by a sinusoidal frequency sweep. This frequency sweep involves 9 different frequency bands at 3 different amplitude levels. These frequency sweeps are repeated, and the data obtained from these frequency sweeps is then processed. The processed data results in data pairs $(\mathbf{p_n}, \mathbf{g_n})$ which can be used for machine learning.

Measurements involved two devices: device A and device B and three different parameter regions: region 1, region 2 and region 3. For the sake of convenience when discussing the results of a particular device / parameter region combination, some notation needs to be employed. For example, when referring to the results for device A in parameter region 3, the results will be referred to as "A3".

In addition, each device/parameter combination was tested under two scenarios:

- **Scenario 1:** The soft and loud gain settings were varied, and the saturation gain settings were kept constant. Each measurement made in this scenario gives rise to $\mathbf{p_n}$ and $\mathbf{g_n}$ vectors with 18 elements each. The acoustic input signal's SPL corresponds to the soft or loud range.

- **Scenario 2:** The soft and loud gain settings were kept constant while the saturation gain settings were varied. Each measurement made in this scenario gives rise to $\mathbf{p_n}$ and $\mathbf{g_n}$ vectors with 9 elements each. The acoustic input signal's SPL is in the saturation region.

The distinction of which scenario a particular result belongs to will also be specified during discussion.

4.1.1 Devices and available data

As mentioned previously, the measurements were performed on two devices, device A and device B. These devices differ mainly in the transducer design they employ. Device A has an electromagnetic transducer while device B has a piezo-electric transducer. The exact design of these transducers or an explanation towards the impact of the transducer design on the results that will be presented is outside the scope of this thesis.

The number of data pairs $(\mathbf{p}_n, \mathbf{g}_n)$ available for each device/parameter region combination for the two aforementioned measurement scenarios is given presented below in Table 2.

	A1	A2	A3	B1	B2	B3
Scenario 1	300	300	300	0	300	0
Scenario 2	100	100	200	0	100	0

Table 2: Number of data points available for each device/parameter region combination for two measurement scenarios

It can be seen from Table 2, that B1 and B3 has no data pairs available and therefore, these combinations were not studied.

Now that the available number of data points is quantified, the required number of data points must also be considered. The number of data points required for the system of linear equations to not be underdetermined for different order models are shown in Table 3.

	Degree 1	Degree 2	Degree 3	Degree 4
Scenario 1	19	190	1330	7315
Scenario 2	10	55	220	715

Table 3: Number of data points required for system of linear equations to be not under-determined for different order models including a bias term.

It is seen that the number of required data points increase significantly as the order of the model increases. Therefore, considering the available data, a model of order 2 is the highest possible model that can be used for both scenarios.

It should be noted that a model of order 2, which is the highest order that fulfils the criterion for not being underdetermined, is quite a low order model and we are quite restricted by the amount of data available. The OLS models where the system of linear equations was underdetermined were also solved using the pseudo-inverse employed in the normal equations as described in section 3.1.1. The ridge regression problems were solved using singular value decomposition as the design matrix was too ill conditioned, especially at very small λ, to be solved accurately using the pseudo-inverse as described in section 3.3.1.

4.1.2 Parameter regions

As previously mentioned, there are three parameter regions being studied. The difference between these parameter regions can be explained in terms of the following aspects:

1. **p -vector variation:** This is the variation between the values of the elements of the **p** vector.
2. **Non −linearity:** This is the amount of non-linearity present in the relationship between a **p** vector in a given parameter region and its corresponding **g** vector.
3. **Ease of mapping**: An indication of how easy it is to produce a mapping between a **p** vector from a given parameter region to its corresponding **g** vector. It is difficult to map a **p** vector to its corresponding **g** vector where there are large non-linearities in the relationship between **p** and its corresponding **g**. Ease of mapping is based on the magnitude of the residual produced by the mapping.

The different parameter regions are presented in terms of the aforementioned aspects in Table 4.

Parameter region	p - vector variation	Non - linearity	Ease of mapping
Region 1	Low	None	Easy
Region 2	Medium	Little	Medium
Region 3	High	Large	Hard

Table 4: Parameter regions described in terms of p *-vector variation, non-linearity and ease of mapping.*

4.2 Quantifying the results

Before presenting the results, it is useful to discuss a few tools that will be used to quantify the results and the reasons for employing them in the discussion.

4.2.1 Residuals

As previously mentioned, residuals will be used to measure the accuracy of the model predictions and three different residuals are used for these. These residuals are outlined as below:

1. $\mathbf{R}_{\mathrm{g}}$**:** This residual evaluates the ability of the model to produce the mapping $\mathcal{P} \mapsto \mathcal{G}$. It is calculated as follows:

$$\mathbf{R}_{\mathrm{g}} = \widehat{\mathbf{G}} - \mathbf{G} = \mathbf{PW} - \mathbf{G} \tag{36}$$

2. $\mathbf{R}_{\mathrm{p}}$**:** This residual evaluates the ability of the model to produce the mapping $\mathcal{G} \mapsto \mathcal{P}$. It is calculated as follows:

$$\mathbf{R}_{\mathrm{p}} = \widehat{\mathbf{P}} - \mathbf{P} = \mathbf{G}\widetilde{\mathbf{W}} - \mathbf{P} \tag{37}$$

3. $\mathbf{R}_{\mathrm{g}}^{*}$ **:** This residual evaluates the ability of the predicted parameters to produce the required gain. It is calculated as follows:

$$\mathbf{R}_{\mathrm{g}}^{*} = \widehat{\mathbf{G}}^{*} - \mathbf{G} = \widehat{\mathbf{P}}\mathbf{W} - \mathbf{G} \tag{38}$$

   The weight matrix $\mathbf{W}$ used to calculate $\mathbf{R}_{\mathrm{g}}^{*}$ is the weight matrix that resulted in the lowest $\mathbf{R}_{\mathrm{g}}$ across all polynomial degrees, with/ without regularization. This was done to ensure that multiple models of different degrees, with/without regularization, that predicts $\widehat{\mathbf{P}}$ is assessed by the same, most accurate model.

Residual percentages can be derived from the residuals and can be defined as below:

$$\text{Residual percentage} = \frac{\text{Predicted value} - \text{True Value}}{\text{True value}} \times 100$$

This formula is applied elementwise to the matrix of predicted values and the matrix containing true values.

Residual percentages provide an elementwise indication of the accuracy of the matrix predicted by the model. A descriptive method of displaying residual percentages is to use a histogram.

4.2.2 Matrix similarity measure

The elementwise similarity between two matrices of the same dimensions can be studied using the L_2 norm according to the formula given below:

$$\frac{\|\mathbf{A}-\mathbf{B}\|}{\frac{1}{2}\|\mathbf{A}+\mathbf{B}\|}$$

where **A** and **B** are the two matrices being studied. For two matrices that are the exact same, this formula would evaluate to zero and any dissimilarities between the elements of the two matrices result in a non-zero output from the formula, where the magnitude of the non-zero output is directly proportional to the dissimilarity.

This similarity measure will be used to compare the following weight matrices for mapping $\mathcal{G} \mapsto \mathcal{P}$:

1. $\widetilde{\mathbf{W}}$ obtained by training the linear regression model $\mathbf{G}\widetilde{\mathbf{W}} = \mathbf{P}$.
2. $\widetilde{\mathbf{W}}^*$obtained by explicitly computing the inverse of $\mathbf{W}$. Note that this computation is only possible for the linear model and the affine model.

The computation of $\widetilde{\mathbf{W}}^*$ given a $\mathbf{W}$ is as follows:
If $\mathbf{W}$ comes from a linear model, then obtaining $\widetilde{\mathbf{W}}^*$ is very straightforward as below:

$$\widetilde{\mathbf{W}}^* = \mathbf{W}^{-1}$$

This is because $\mathbf{W}$ from a linear model in this thesis is a square matrix and is invertible.
This $\widetilde{\mathbf{W}}^*$ can be used to predict the parameter matrix as follows:

$$\widehat{\mathbf{P}} = \mathbf{G}\widetilde{\mathbf{W}}^*$$

If $\mathbf{W}$ comes from an affine model, $\mathbf{W}$ needs to be decomposed into $\mathbf{b}$ and $\mathbf{W}^*$**,** where $\mathbf{b}$ is a row vector with the elements from the first row of $\mathbf{W}$ and $\mathbf{W}^*$ is a matrix with elements from the rest of the rows of $\mathbf{W}$. Now, $\widetilde{\mathbf{W}}^*$can be calculated as:

$$\widetilde{\mathbf{W}}^* = \mathbf{W}^{*-1}$$

However, making predicting the parameter matrix using $\widetilde{\mathbf{W}}^* = \mathbf{W}^{*-1}$ is slightly different and it is as follows:

$$\widehat{\mathbf{P}} = (\mathbf{G}-\mathbb{1}\mathbf{b})\widetilde{\mathbf{W}}^*$$

Where $\mathbb{1}$ is column vector of ones with the same number of rows as $\mathbf{G}$.

Both $\widetilde{\mathbf{W}}^*$ and $\widetilde{\mathbf{W}}$ will be evaluated in terms of $\mathbf{R_p}$.

4.2.3 Norms of the weight matrix

As outlined previously, ridge regression penalises the L_2 norm of the weights and LASSO regression penalises the L_1 norm of the weights. Therefore, it is relevant to study the effects of varying the regularisation parameter λ on the L_2 norm of ridge regression weights and on the L_1norm of the LASSO regression weights.

The number of non-zero weights is another aspect of the weight matrix that's worth considering for the regularised models. This is because one of the advantages of LASSO is that it sets some weights exactly to zero. A weight matrix with a lot of zeros is beneficial as it results in a model that is computationally efficient when making predictions.

The number of non-zero elements in a matrix can be obtained by its L_0 norm. Therefore, a model which has a weights matrix with a low L_0 norm is desirable.

4.3 Linear Model

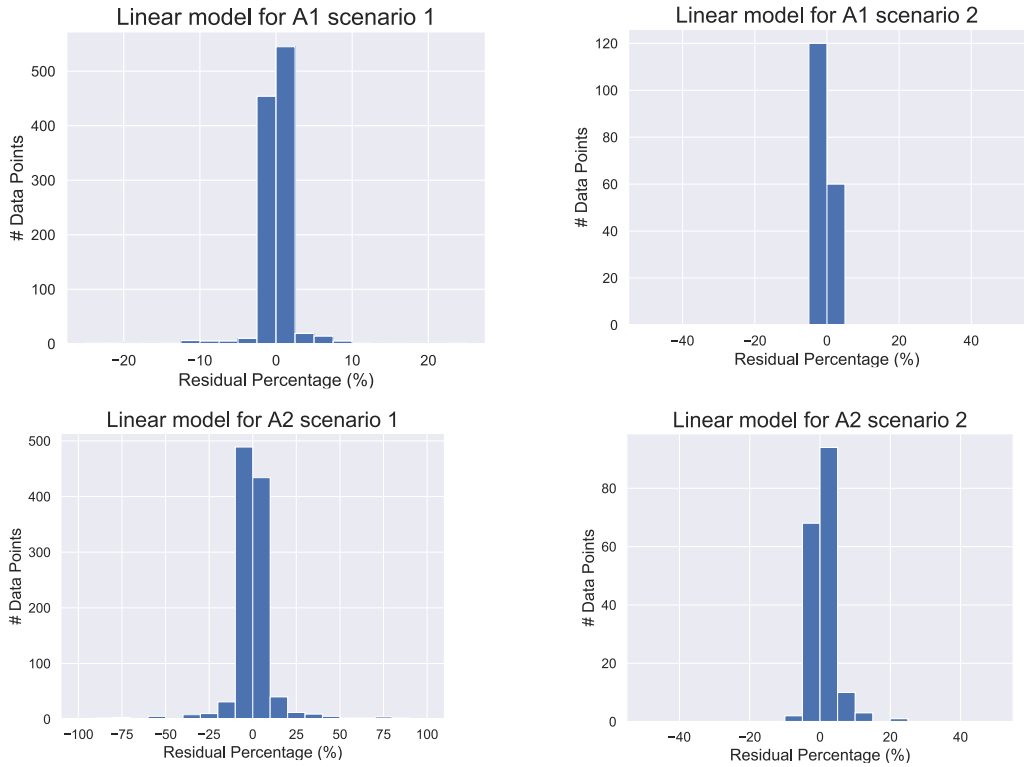
This section will present the results obtained by the linear model as described in section 3.1.1. The results will be presented in terms of the MSR and also in terms of the distribution of the residual percentage.

4.3.1 Mapping $\mathcal{P} \mapsto \mathcal{G}$

The MSR and histogram of $\mathbf{R}_g$ residuals for scenario 1 and 2 for all available device/parameter region combinations are shown below in Table 5 and Figure 9 respectively.

	A1	A2	A3	B2
Scenario 1	0.0052	0.1090	6.2583	4.3150
Scenario 2	0.0401	0.2460	3.8806	1.1645

Table 5: Linear model MSR for different device/parameter region combinations for scenario 1 and scenario 2. These MSR values correspond to the mapping $\mathcal{P} \mapsto \mathcal{G}$.



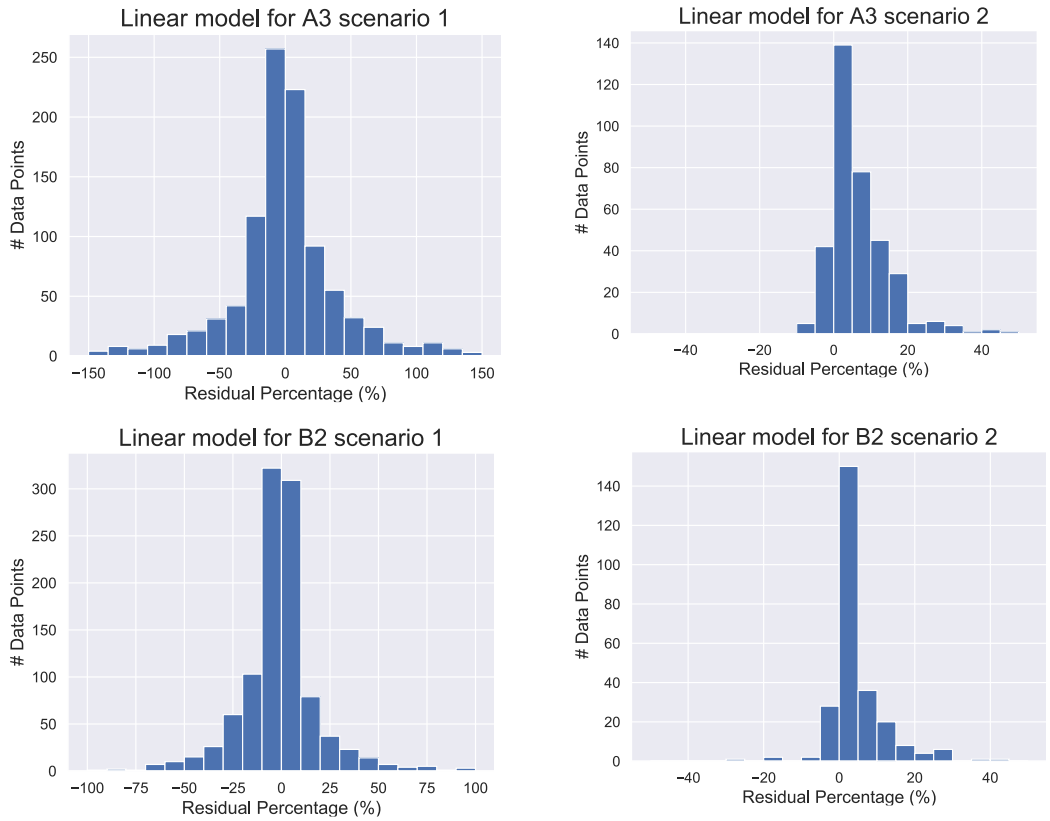


Figure 9: Histograms of residual percentage of all device/parameter region combinations for scenario 1 and scenario using the linear model. These models are producing the mapping $\mathcal{P} \mapsto \mathcal{G}$.

It can be seen that the MSR and the residual percentage increases as the parameter region number increases. This is to be expected as the difficulty of mapping increases with the parameter region as outlined in Table 4. It can also be observed that the prediction error is much higher for B2, when compared to A2.

In general, predictions for scenario 2 produces less error than scenario 1. One possible reason for this is because there are only 9 elements in the **p** vector for scenario 2, whereas, there are 18 elements in the in the **p** vector for scenario 1. Thus, training a model for scenario 2 requires less data (for system of linear equations to be not underdetermined) and less data is required for the system of linear equations to be overdetermined. Having an overdetermined system is preferred as some of the data points could be outliers (measurement noise, etc.) and having more data points may reduce random disturbances by averaging. Thus, the weights obtained from solving this system will produce more accurate predictions.

4.3.2 Mapping $\mathcal{G} \mapsto \mathcal{P}$

The mapping $\mathcal{G} \mapsto \mathcal{P}$ was also studied and the results are presented below.

4.3.2.1 Weight matrix similarity

As previously mentioned, the similarity between $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{W}}^*$ was calculated and it can be seen below in Table 6.

	A1	A2	A3	B2
Scenario 1	0.3059	0.1319	0.3053	1.2602
Scenario 2	1.8947	1.6981	1.8827	1.5022

Table 6: Similarity between $\widetilde{\boldsymbol{W}}$ (trained from scratch) and $\widetilde{\boldsymbol{W}}^*$ (obtained from inverting $\boldsymbol{W}$) where values close to zero indicate that the matrices are very similar.

A general trend that can be observed is that the dissimilarity between $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{W}}^*$ is higher for scenario 2 than scenario 1. In addition, the dissimilarity is substantially higher for B2 than A2 for scenario 1. This indicates that for these cases, the mapping $\mathcal{P} \mapsto \mathcal{G}$ and $\mathcal{G} \mapsto \mathcal{P}$ requires different linear models in a least squares sense.

4.3.2.2 $\mathbf{R}_\mathrm{p}$

The $\mathbf{R}_\mathrm{p}$ MSR produced by $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{W}}^*$ for scenario 1 and scenario 2 by the linear model are represented in a bar chart in Figure 10.

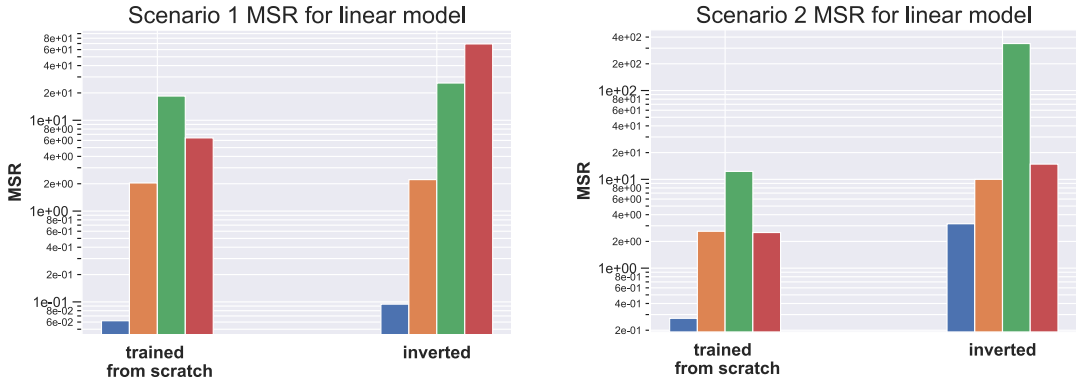


Figure 10: MSR of $\boldsymbol{R_p}$ for linear models for scenarios 1 and 2, where the weights $\widetilde{\boldsymbol{W}}$ (obtained by training from scratch) and $\widetilde{\boldsymbol{W}}^*$ (obtained by inverting $\boldsymbol{W}$) were used. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively.

It is seen that the weights that were trained from scratch perform better and this is more pronounced in the case of scenario 2. It should also be noted that the scenario 2 MSR is higher than scenario 1 and this is a contrast to the case of $\mathbf{R}_\mathrm{g}$. This might indicate that it might be more difficult to map $\mathcal{G} \mapsto \mathcal{P}$ than $\mathcal{P} \mapsto \mathcal{G}$.

It should be noted that on this plot, the horizontal gridlines are spaced logarithmically, so differences in height between the bars are very substantial in regard to the magnitude differences of the MSR. The same is also true regarding the variable presented on the ordinate axis for other log plots that will be presented in this report.

The MSR trend seen from Figure 10 is further substantiated by the residual percentage histograms presented in Figure 11, where the residual percent histograms produced by weights trained from scratch has lesser instances of high error percentages in comparison to weights obtained from inverting $\mathbf{W}$.

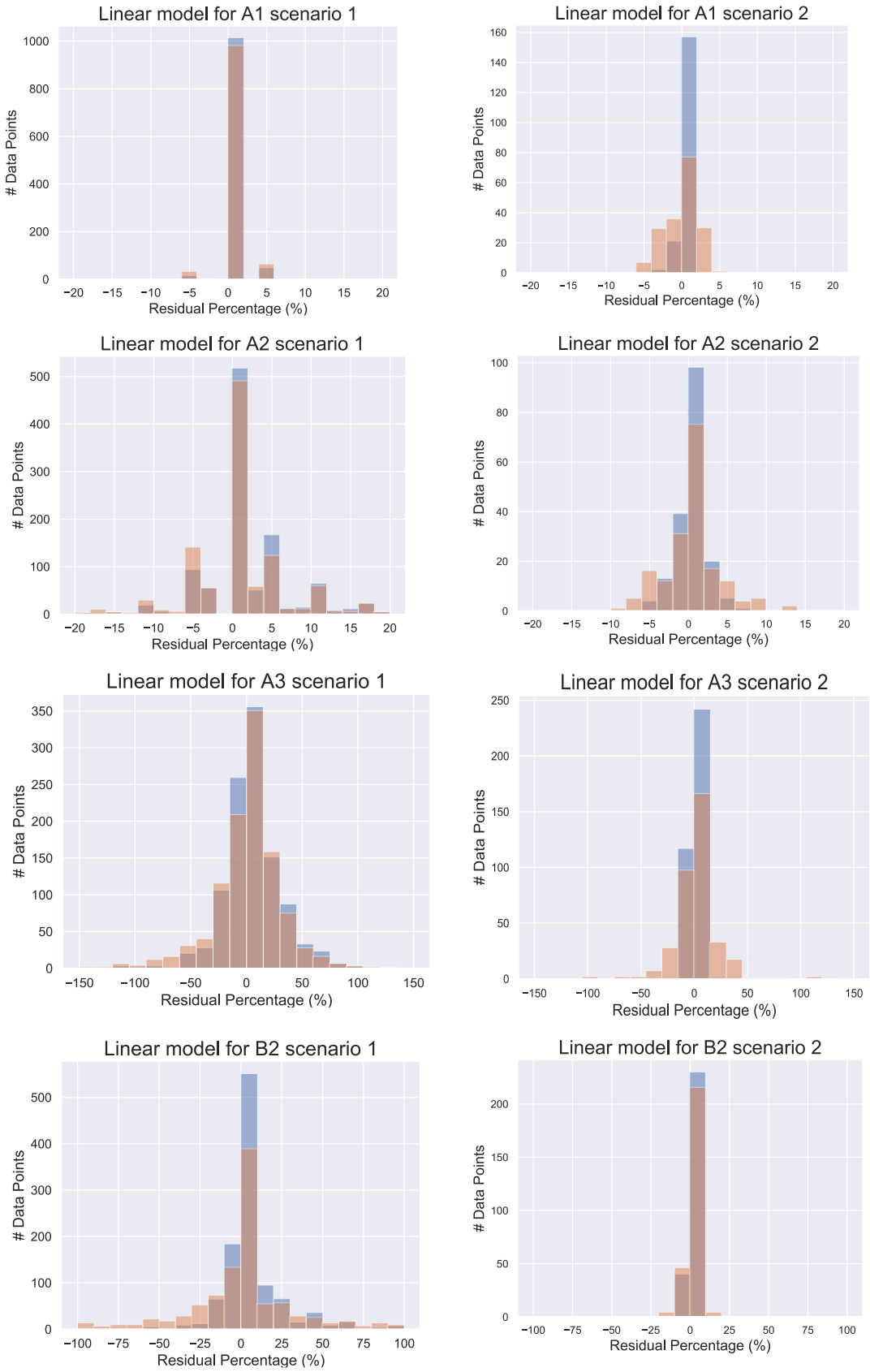


Figure 11: Residual percentages expressed as histograms for linear models that map $\mathcal{G} \mapsto \mathcal{P}$. The results produced by $\widetilde{\boldsymbol{W}}$ (obtained by training from scratch) and $\widetilde{\boldsymbol{W}}^*$ (obtained by inverting $\boldsymbol{W}$) are shown in blue and orange respectively.

4.3.2.3 $\mathbf{R}_g^*$

The MSR for $\mathbf{R}_g^*$ for the weights trained from scratch for both scenarios and for all available device/parameter region combinations are presented below in Table 7.

	A1	A2	A3	B2
Scenario 1	0.0026	0.1906	3.5116	2.2074
Scenario 2	0.0200	0.1805	2.0264	0.7094

Table 7: The $\boldsymbol{R}_g^*$ MSR values produced by a linear model with weights trained from scratch ($\widetilde{\boldsymbol{W}}$) for all available device/parameter region combinations for scenarios 1 and 2.

4.4 Affine Model

The results obtained for the affine model will be presented in this section. As previously discussed, an affine model is a linear model with an added bias term which is not dependant on the independent variables in the model.

For the sake of brevity, residual percent histograms are omitted from this section in the body of the report, as the trends observed were the same as in the case of the linear model. These histograms are included in the appendices of this report for the interested reader.

4.4.1 Mapping $\mathcal{P} \mapsto \mathcal{G}$

The $\mathbf{R}_g$ MSR for the scenarios 1 and 2 for all the available device/parameter region combinations is presented in Table 8.

	A1	A2	A3	B2
Scenario 1	0.0048	0.0792	5.8837	4.0677
Scenario 2	0.0196	0.1751	1.4065	0.3314

Table 8: Affine model MSR for different device/parameter region combinations for scenario 1 and scenario 2. These MSR values correspond to the mapping $\mathcal{P} \mapsto \mathcal{G}$.

It is seen that the MSR for the affine model is lower than that of the linear model for both scenarios and all device/parameter region combinations. This shows that the additional descriptive power of the affine model is beneficial in comparison to the simpler linear model.

4.4.2 Mapping $\mathcal{G} \mapsto \mathcal{P}$

Similar to the linear model, weight matrix similarity, MSR for $\mathbf{R}_P$ and $\mathbf{R}_g^*$ were studied and the results are presented in this section.

4.4.2.1 Matrix Similarity

The weight matrix similarity between $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{W}}^*$ were calculated and is shown in Table 9.

	A1	A2	A3	B2
Scenario 1	0.3050	0.0865	0.3141	1.3437
Scenario 2	0.9073	1.6771	1.8987	1.4599

Table 9: Similarity between $\widetilde{\boldsymbol{W}}$ (trained from scratch) and $\widetilde{\boldsymbol{W}}^*$ (obtained from inverting $\boldsymbol{W}$) where values close to zero indicate that the matrices are very similar.

Similar, to the linear model the dissimilarity between the weight matrices $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{W}}^*$is higher for scenario 2 than scenario 1 and there is also a significantly higher dissimilarity for B2 than A2 for scenario 1.

4.4.2.2 $\mathbf{R_p}$

The $\mathbf{R}_\mathrm{p}$ MSR produced by $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{W}}^*$ for scenario 1 and scenario 2 by the affine model are represented in a bar chart in Figure 12.

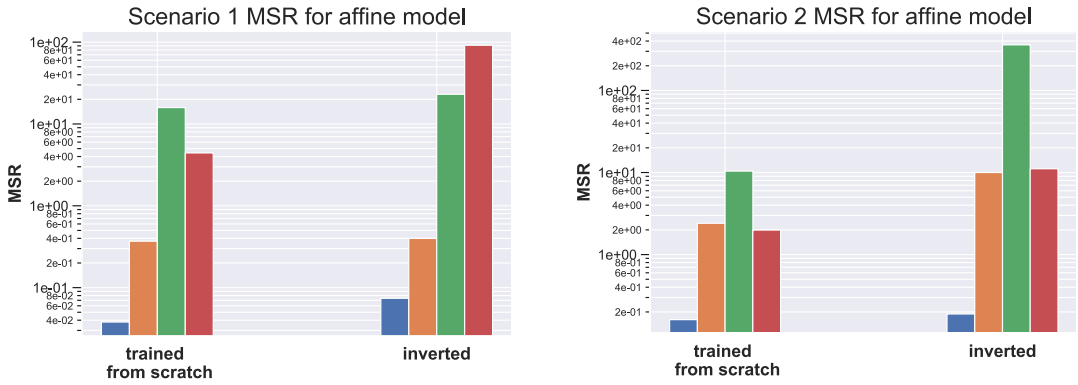


Figure 12: : MSR of $\boldsymbol{R_p}$ for affine models for scenarios 1 and 2, where the weights $\widetilde{\boldsymbol{W}}$ (obtained by training from scratch) and $\widetilde{\boldsymbol{W}}^*$ (obtained by inverting $\boldsymbol{W}$) were used. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively.

Similar to the linear model, the affine model where the weights were trained from scratch produced better MSR for $\mathbf{R}_\mathrm{p}$. The MSR for scenario 2 was also higher than scenario 1 as it was for the linear model. Slight improvements in the MSR of $\mathbf{R}_\mathrm{p}$ were seen for the affine model in comparison to the linear model.

4.4.2.3 $\mathbf{R}_\mathbf{g}^*$

The $\mathbf{R}_\mathrm{g}^*$ MSR produced by $\widetilde{\mathbf{W}}$ for scenario 1 and scenario 2 by the affine model are presented in Table 10.

	A1	A2	A3	B2
Scenario 1	0.0012	0.0025	1.1894	3.1721
Scenario 2	0.0014	0.0237	0.2069	0.0477

Table 10: The $\boldsymbol{R_g^*}$ MSR values produced by an affine model with weights trained from scratch $(\widetilde{\boldsymbol{W}})$ for all available device/parameter region combinations for scenarios 1 and 2.

The MSR of $\mathbf{R}_\mathrm{g}^*$ is vastly improved for the affine model in comparison the linear model. The only exception to this B2 scenario 1 where the MSR $\mathbf{R}_\mathrm{g}^*$ is worse for the affine model.

4.5 Higher Order Models

In this section the results obtained from models of orders 2, 3 and 4 will be discussed. The model types that will be explored are ordinary least squares, ridge regression and LASSO regression.

4.5.1 Ordinary Least Squares

This section will discuss results obtained by ordinary least squares models of orders 2, 3 and 4.

4.5.1.1 Mapping $\mathcal{P} \mapsto \mathcal{G}$

The MSR and residual percent histograms for $\mathbf{R}_g$ are presented in Figure 13 and Figure 14.

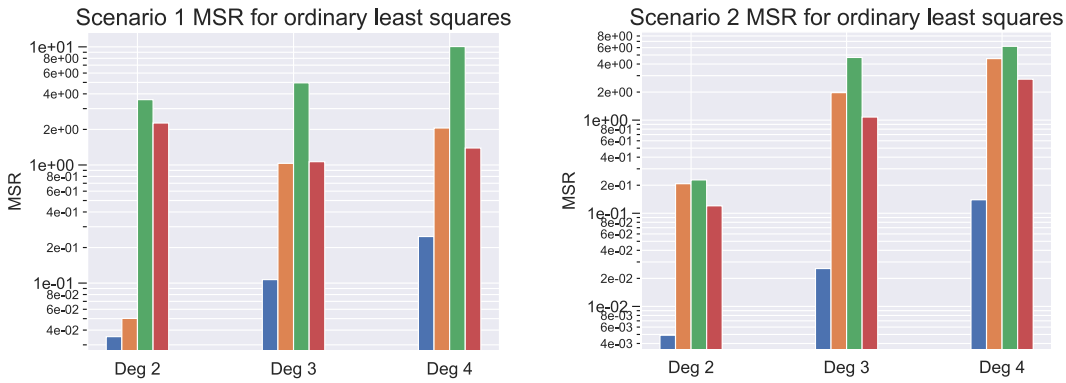
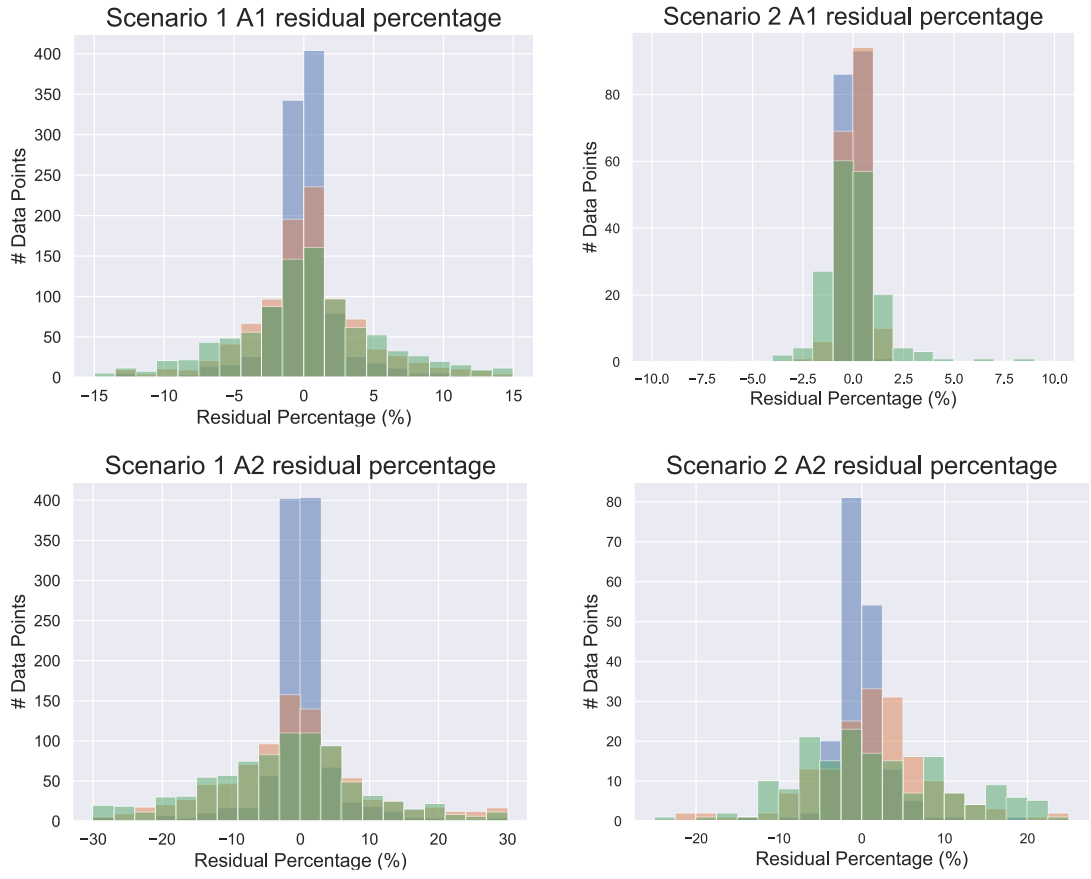


Figure 13: MSR of $\boldsymbol{R}_g$ for OLS models of varying orders for scenarios 1 and 2. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively.



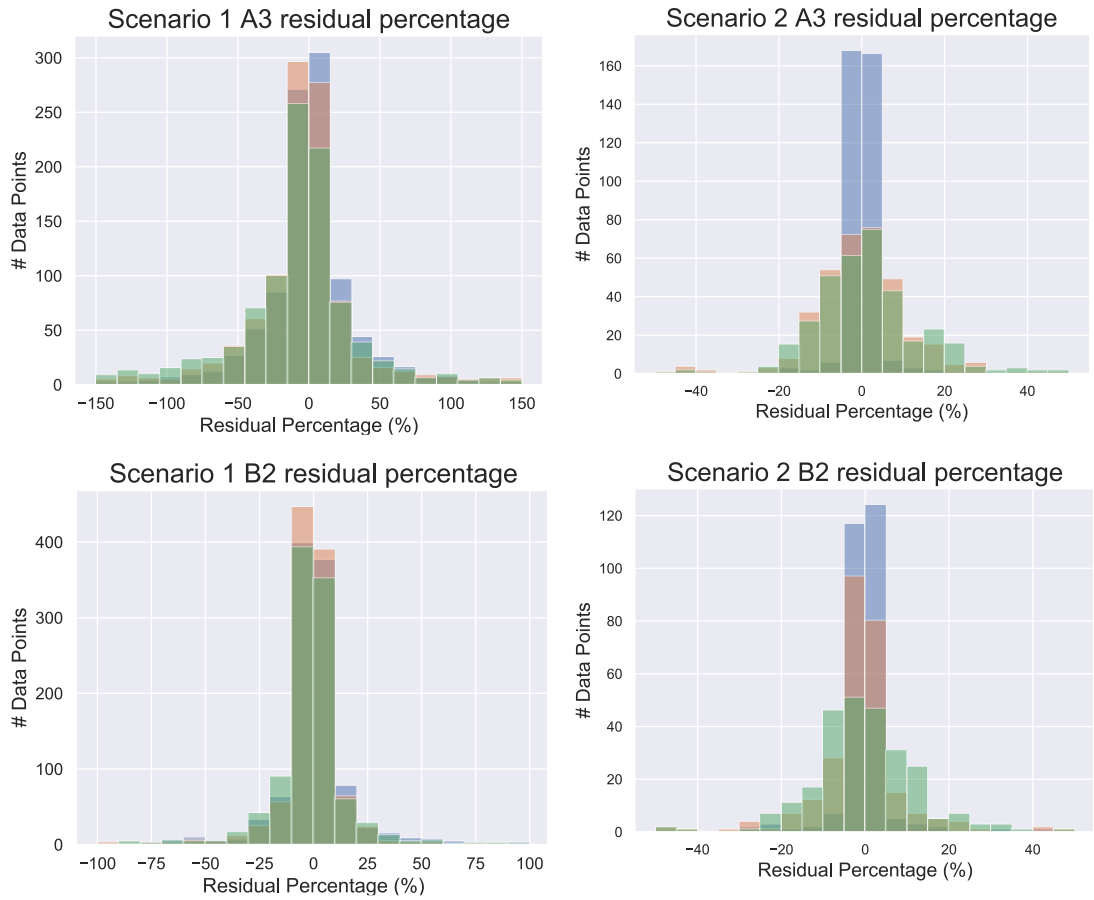


Figure 14: $\boldsymbol{R}_g$ residual percentage histogram for OLS models of order 2 (blue), 3 (orange) and 4 (green) for scenarios 1 and 2 for all device / parameter range combinations.

As previously discussed in section 4.1.1, models of degree 3 and 4 were obtained from underdetermined systems and these models are prone to overfitting. This can be seen by the increase in the MSR for models of degree 3 and 4 in comparison to degree 2 models. This is also evident in the residual percent histograms as the order 2 models have lower error percentages. The MSR and residual percentage for scenario 1 B2 is an exception to this trend as the residuals for orders 3 and 4 are better than they are order 2.

As expected, the order 2 model performed better than the order 1 (affine) model for all device/parameter region combinations for scenario 1, except for A1 which produced better residuals with the affine model. This indicates that a model of order 2 might be too complex for making predictions for A1 and this is logical as A1 was expected to exhibit linear behaviour according to Table 4 and a model of order 1 is linear.

For scenario 2, models of order 2 performed better than order 1 for all device/parameter region combinations except for A2, where the MSR was slightly worse for order 2.

4.5.1.2 Mapping $\mathcal{G} \mapsto \mathcal{P}$

The MSR and residual percentage histograms for $\mathbf{R}_\mathrm{p}$ are shown below in Figure 15 and Figure 16.

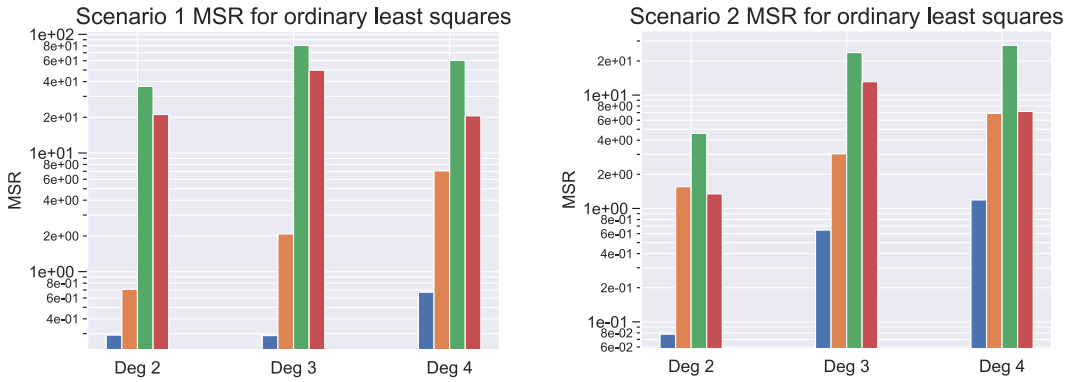
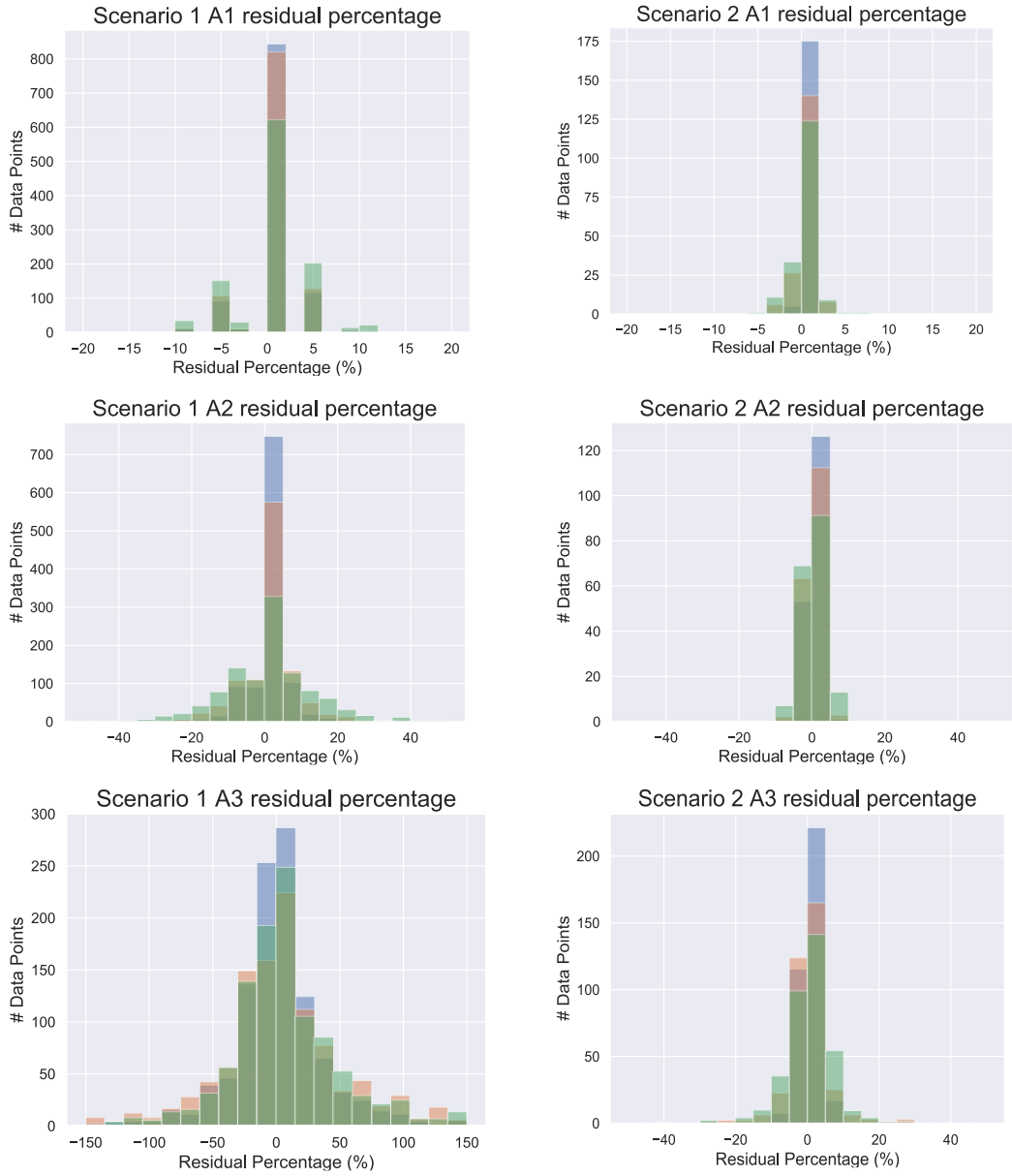


Figure 15: MSR of $\boldsymbol{R}_p$ for OLS models of varying orders for scenarios 1 and 2. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively.



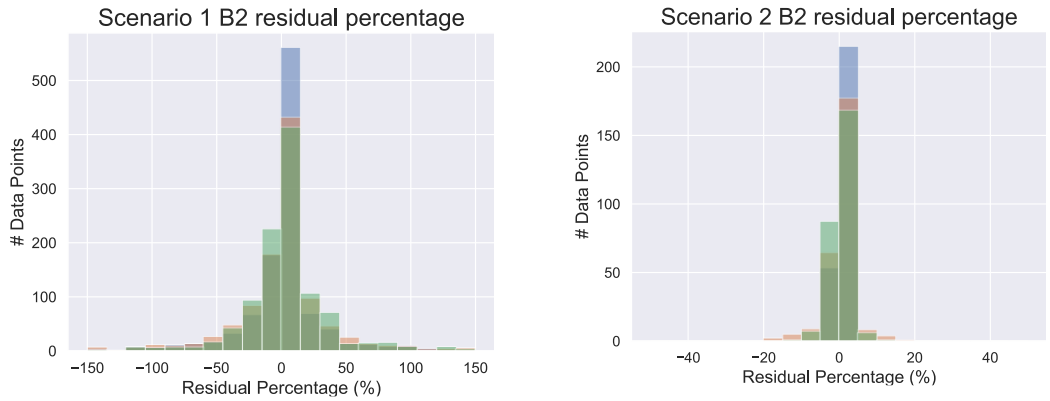


Figure 16: $\boldsymbol{R}_P$ residual percentage histogram for OLS models of order 2 (blue), 3 (orange) and 4 (green) for scenarios 1 and 2 for all available device / parameter region combinations.

In the case of scenario 1, as expected, models that were derived from underdetermined systems produced worse residuals, which can be seen from the MSR and residual percentage histograms. For scenario 2, similar to the $\mathcal{P} \mapsto \mathcal{G}$ MSR values, the order 2 models performed better than order 1 for all cases except A2.
The affine model outperformed the higher order OLS models for all device parameter region combinations for scenario 1. This may indicate that the mapping $\mathcal{G} \mapsto \mathcal{P}$ for scenario 1 might be best produced by the simpler affine model instead of a higher degree model.

The $\mathbf{R}_{\mathrm{g}}^{*}$ residuals were also studied for the higher order OLS models and the MSR for this can be seen in Figure 17.

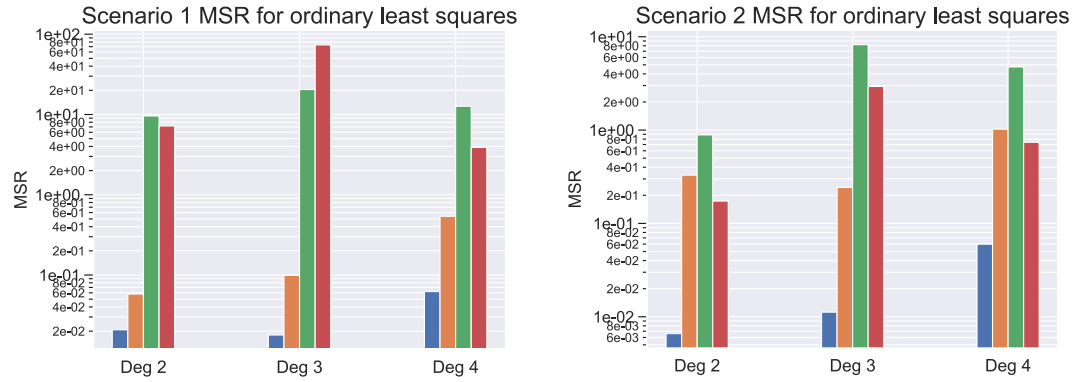


Figure 17: MSR of $\boldsymbol{R}_g^*$ for OLS models of varying orders for scenarios 1 and 2. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively.

From the comparison of the above figure with Table 10, it is evident that the affine model produced better $\mathbf{R}_{\mathrm{g}}^{*}$ MSR for scenarios 1 and 2, than the higher order OLS models.
As seen previously, the 2nd order model produced better $\mathbf{R}_{\mathrm{p}}$ residuals for scenario 2 in comparison to the affine model. This suggests, as it was initially hypothesised, that the $\mathbf{R}_{\mathrm{p}}$ residual may not have much validity when assessing the predicted parameter vector $\hat{\mathbf{p}}$ as multiple $\mathbf{p}$ vectors could produce the same $\mathbf{g}$ vector. Therefore, just because a given predicted parameter vector $\hat{\mathbf{p}}_{\mathbf{1}}$is close to $\mathbf{p}_{ts}$, the "true" parameter vector from the test set, doesn't mean that it will produce the required $\mathbf{g}_{ts}$. Another predicted parameter vector $\hat{\mathbf{p}}_{\mathbf{2}}$,which is further from $\mathbf{p}_{ts}$, could produce the required $\mathbf{g}_{ts}$. Thus, $\mathbf{R}_{\mathrm{g}}^{*}$ might be a better criterion to assess $\hat{\mathbf{p}}$.

4.5.2 Ridge regression

The previously discussed higher order models were also studied with ridge or L_2 regularization. The obtained results are presented in this section. The residual histograms for ridge regression were omitted from the main body of this report and can be found in the appendices. This was because they didn't add much more insight than the MSR values alone.

4.5.2.1 Effect of varying the regularization parameter (λ) on the MSR

The regularization parameter λ was varied and the effect this had on the MSR for the validation data and the MSR for the training data was studied. The results are presented in the plots in Figure 18 and Figure 19. It should be noted that for training data MSR vs. λ plots only A2 is presented in the main body of this report in the interest of brevity and the trends seen in Figure 19 was the same for all device/parameter region combinations.

From here onwards, all the plots that have λ on the abscissa axis are plotted with log scale on both axes. As a consequence, it is impossible to include $\lambda = 0$ as the logarithm of 0 is undefined. In order to remedy this, a λ of very small magnitude was found for each plot which yields the same result as $\lambda = 0$ and this non-zero λ was used as the equivalent of $\lambda = 0$. Thus, when discussing results for $\lambda = 0$, it should be noted that in reality the result produced by this small λ is what is being referred to.

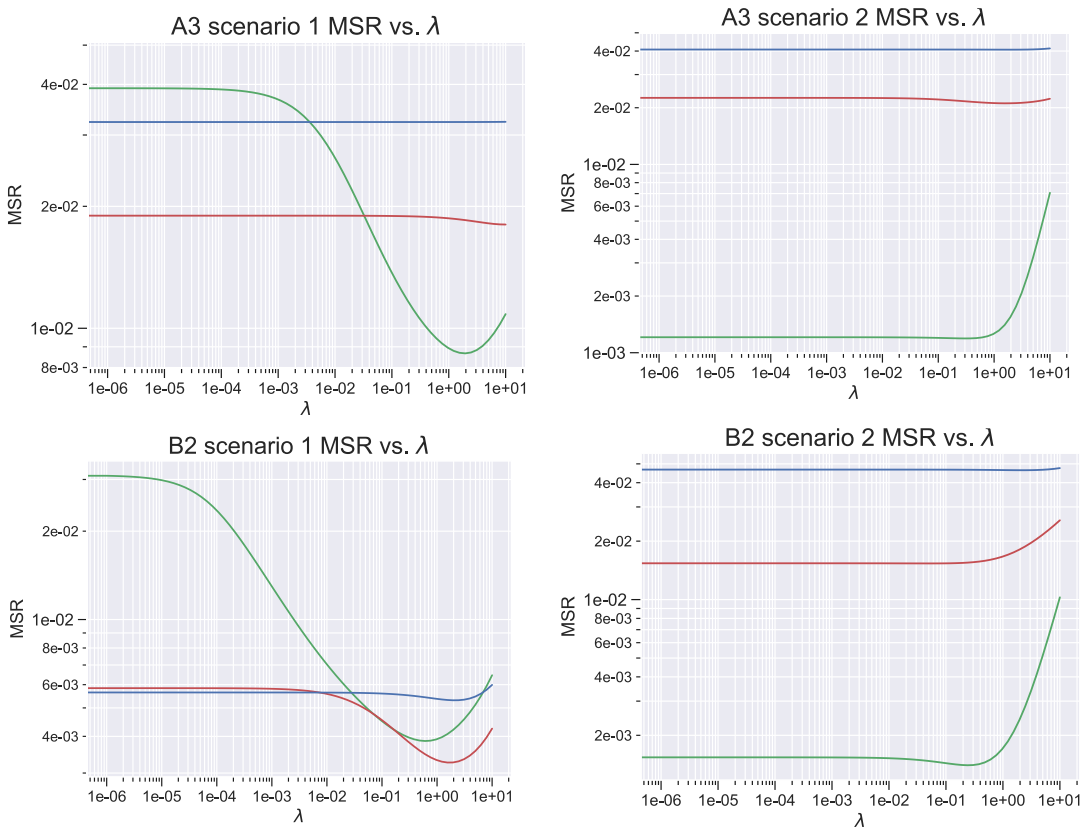


Figure 18: MSR vs. λ for ridge regression with polynomials of degree 2 (green), degree 3 (red) and degree 4 (blue) for validation data.

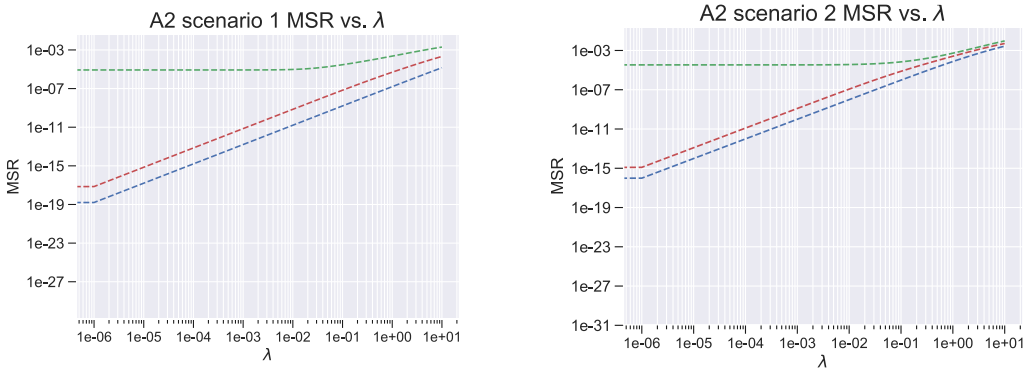


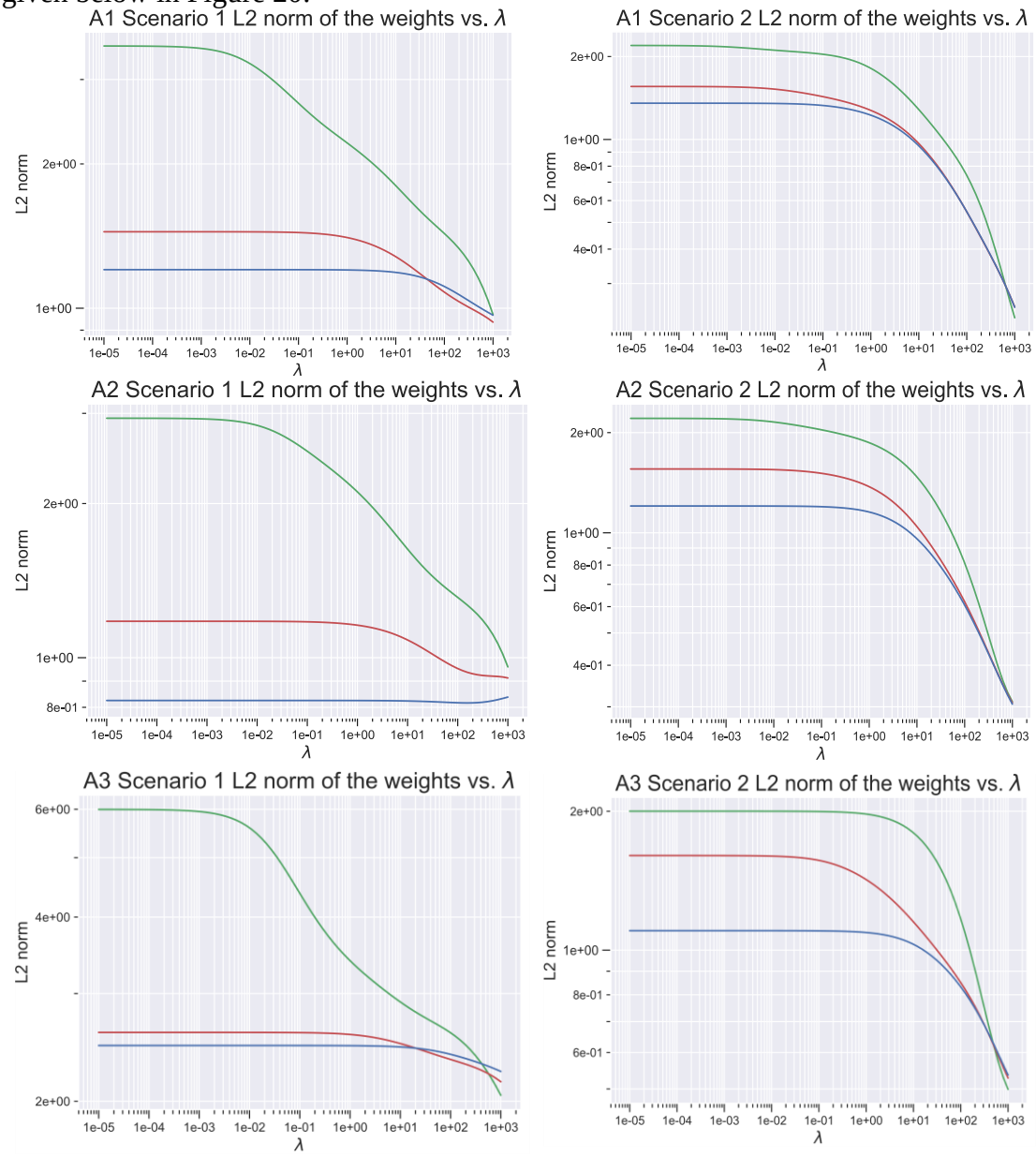
Figure 19: MSR vs. λ for ridge regression with polynomials of degree 2 (green), degree 3 (red) and degree 4 (blue) for training data for scenarios 1 and 2 for A2.

A general trend that can be observed from Figure 18 is that there seems to be an optimal value of λ where the MSR is the lowest. Unlike all the other cases, for the 3rd order and 4th order models for A1 and A2 for scenario 2, the optimal value for λ was found to be zero. It is seen from the plots that the MSR for these cases is almost independent of λ as it stays relatively constant as λ is varied and there is no pronounced minimum. If $\lambda = 0$ is used, there is no regularization and the model is an OLS model. This means that L_2 regularization will not improve the OLS models of these particular degrees. This could be a consequence of the optimisation algorithm not finding the most optimal and non-zero λ. However, it is seen that most of the models produce better residuals when L_2 regularization is used and is better than the OLS models.

The effect of varying λ on the training data MSR is shown in Figure 19 and it is seen, as expected, that the training data MSR becomes worse as λ is increased from 0, where the validation data MSR becomes better. This is a classic sign of overfitting and it is more evident in the higher order models which are more prone to overfitting.

4.5.2.2 The effect of λ on L_2 norm of the weights

The effect that λ has on the r L_2 norm of the weights was studied, and the results are given below in Figure 20.



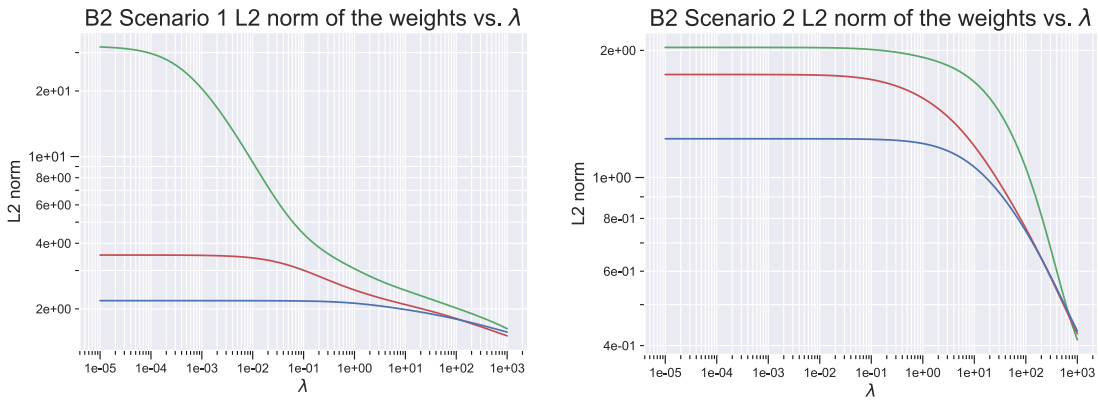


Figure 20: L_2 Norm of the weights vs. λ for ridge regression with polynomials of degree 2 (green), degree 3 (red) and degree 4 (blue).

A general trend that can be seen in all the plots of the L_2 norm of the weights vs. λ is that the L_2 norm decreases as λ is increased.

4.5.2.3 The effect of λ on the L_0 norm of the weights

The effects of varying λ on the number of non-zero weights were studied for L_2 regularized models of order 2, 3 and 4. For all these models, the number of non-zero weights remained constant for all ranges of λ that were trialled.

4.5.2.4 Mapping $\mathcal{P} \mapsto \mathcal{G}$

The $\mathbf{R}_g$ residual for L_2 regularized models were studied and the MSR is given below in Figure 21.

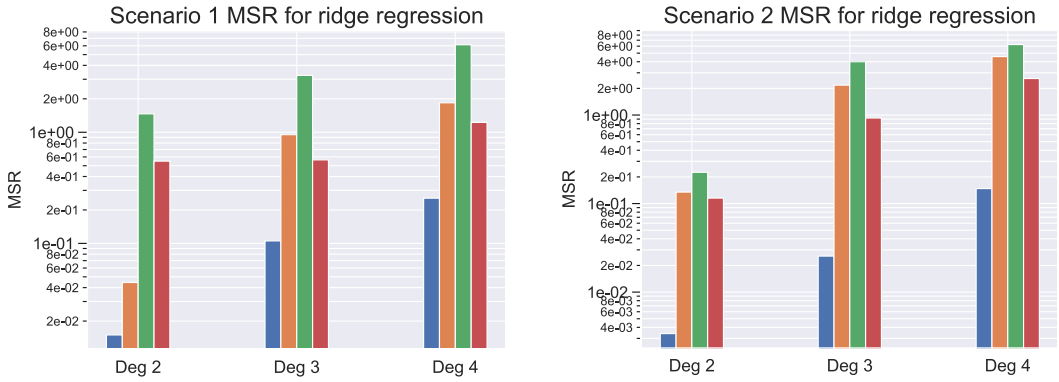


Figure 21: MSR of $\boldsymbol{R}_g$ for ridge regularized models of varying orders for scenarios 1 and 2. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively.

It is observed that the MSR values are better for the L_2 regularized model than the linear and the affine models in all cases and better than the higher degree OLS models in most cases. The cases where the OLS models out-perform the L_2 regularized models are where the optimal value for λ was found to be 0, i.e. for models of order 3 and 4 for A1 and A2 for scenario 2.

4.5.2.5 Mapping $\mathcal{G} \mapsto \mathcal{P}$

The MSR values for the $\mathbf{R}_p$ and $\mathbf{R}_g^*$ were studied and are presented below in Figure 22 and Figure 23.

$\mathbf{R}_p$

From the MSR of the $\mathbf{R}_p$ residuals given below, for scenario 1 it is seen that the MSR is consistently better for the ridge regularized models. However, this trend did not always appear for scenario 2 and there were exceptions where the OLS models performed better.

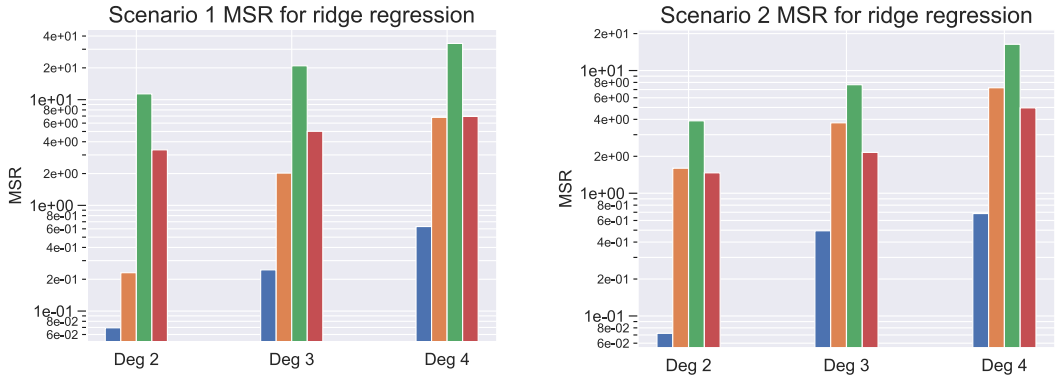


Figure 22: MSR of $\boldsymbol{R}_p$ for ridge regularized models of varying orders for scenarios 1 and 2. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively.

$\mathbf{R}_g^*$

From the $\mathbf{R}_g^*$ MSR values presented below, it is seen that for majority of the cases, the ridge regularized model performs better.

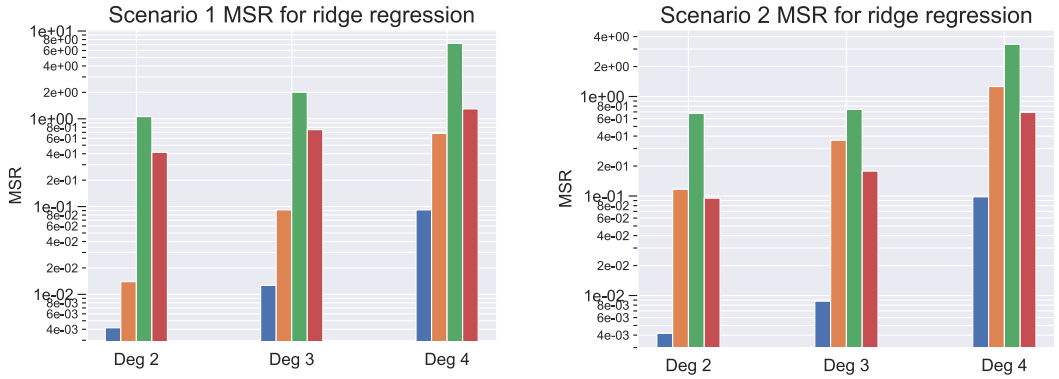


Figure 23: MSR of $\boldsymbol{R}_g^*$ for ridge regularized models of varying degrees for scenarios 1 and 2. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively. LASSO regression

4.5.3 LASSO regression

LASSO regularization was also performed on the higher order models and the effect of varying λ on the MSR, L_1 norm and L_0 norm was studied, and the results are presented in the following sections.

As in the case of ridge regression, the residual histograms were omitted from the main body of this report. However, these figures are included in the appendices of this report. In addition, plots indicating the change in MSR for the training data was also omitted as the trends are the exact same as that observed in the case of ridge regression.

4.5.3.1 Effect of varying the regularisation parameter on the MSR

Plots indicating the change in $\mathbf{R}_g$ MSR with respect to λ for different degree models is seen below in Figure 24.

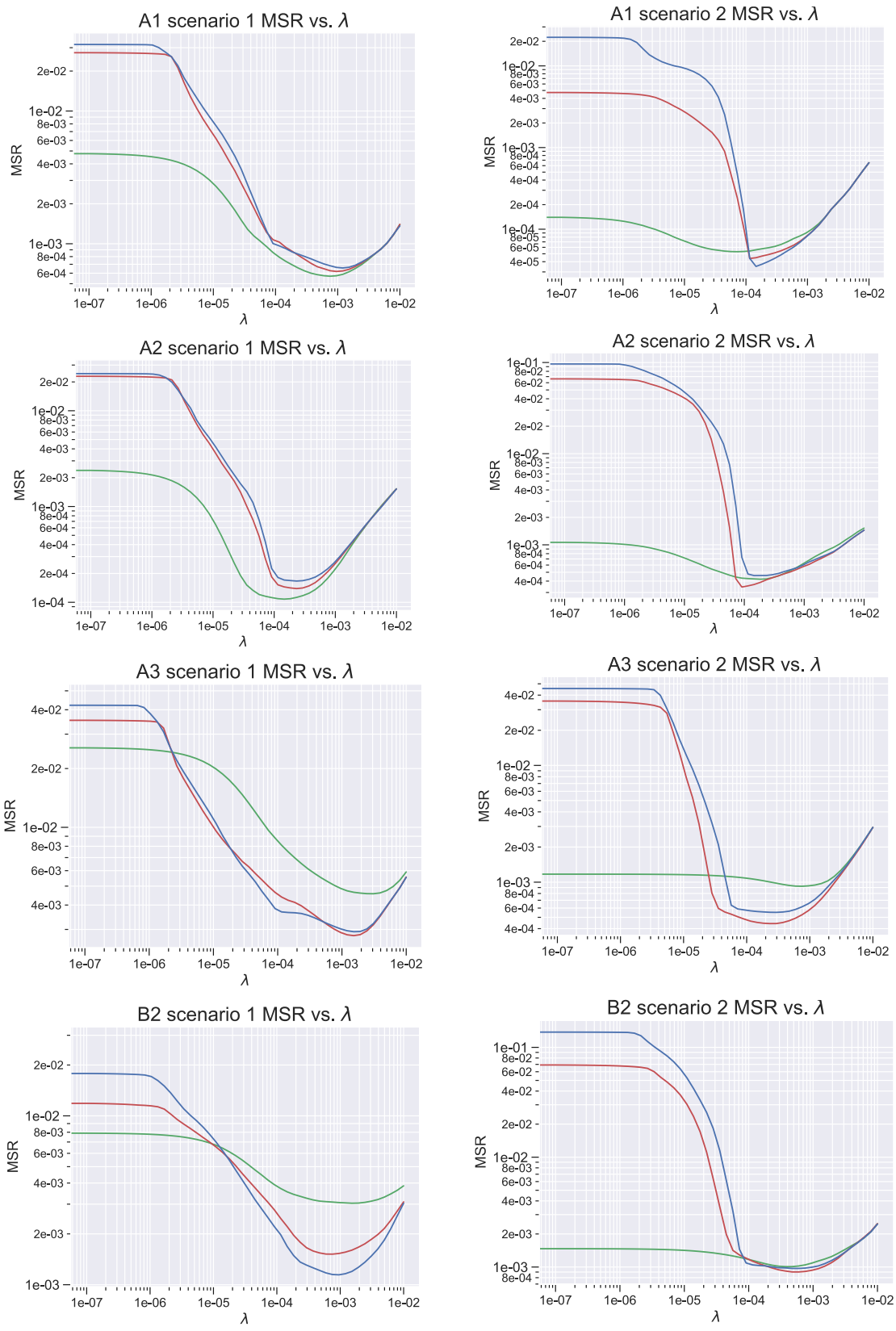


Figure 24: MSR vs. λ for lasso regression with polynomials of degree 2 (green), degree 3 (red) and degree 4 (blue) for the validation data.

Similar to the case for ridge regression, it is seen that MSR decreases when λ is increased from 0, until it reaches a minimum. A few differences to the corresponding

ridge regression plots shown in Figure 18 is that the decrease in MSR is much higher and the MSR is much more sensitive to change in λ.

4.5.3.2 Effect of varying λ on the L_1 norm of the weights

The effect of varying λ on the regularisation term (L_1 norm of the weights) was studied and the results can be seen below in Figure 25.

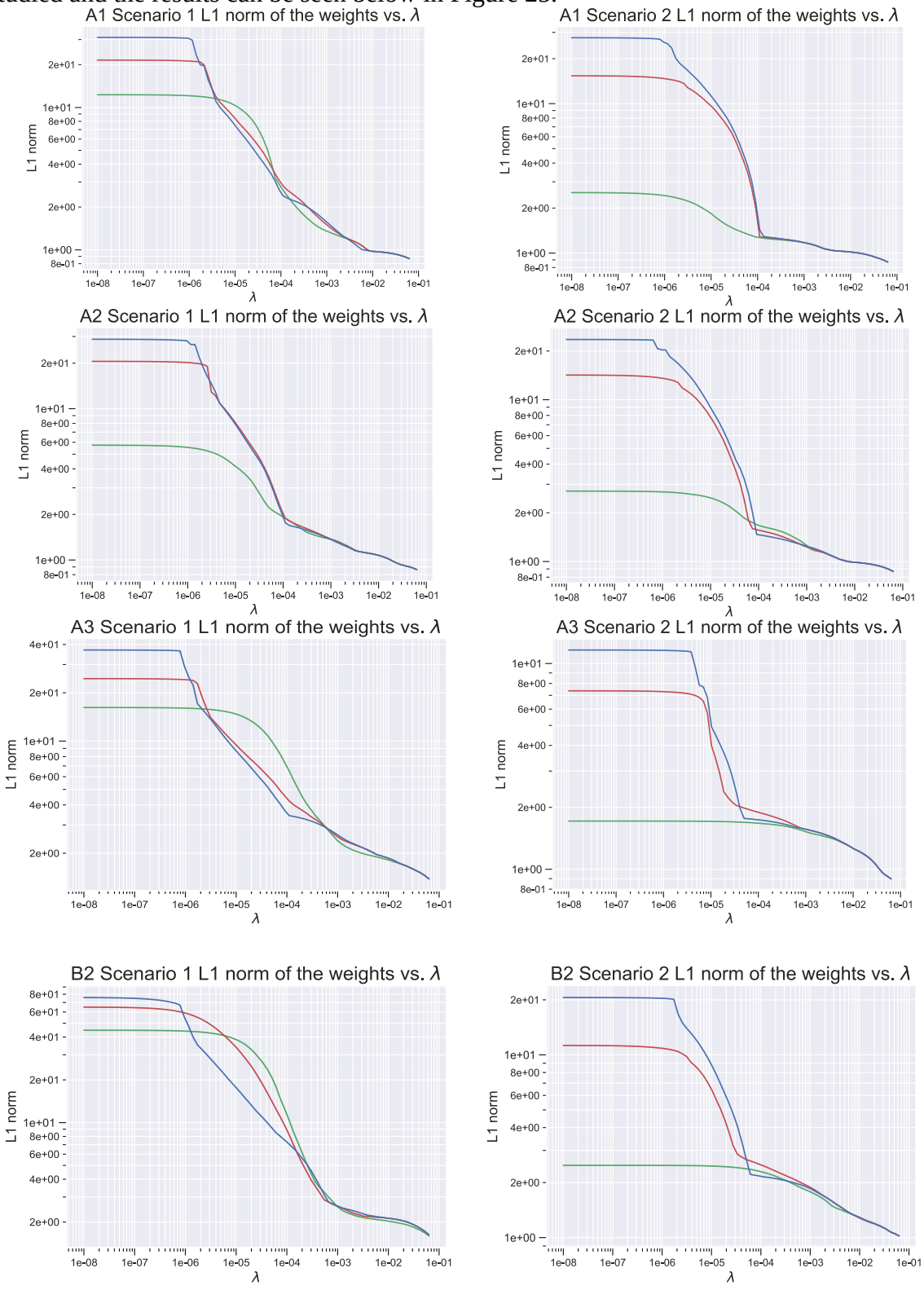


Figure 25: L_1 Norm of the weights vs. λ for LASSO regression with polynomials of degree 2 (green), degree 3 (red) and degree 4 (blue).

It is seen, similar to the ridge regression that the L_1norm which represents the summed magnitude of the weights decreases as λ is increased.

4.5.3.3 Effect of varying the λ on the L_0 norm of the weights

The L_0 norm of the weights which represents the number of non-zero weights were studied as a function of λ and the results can be seen in Figure 26.

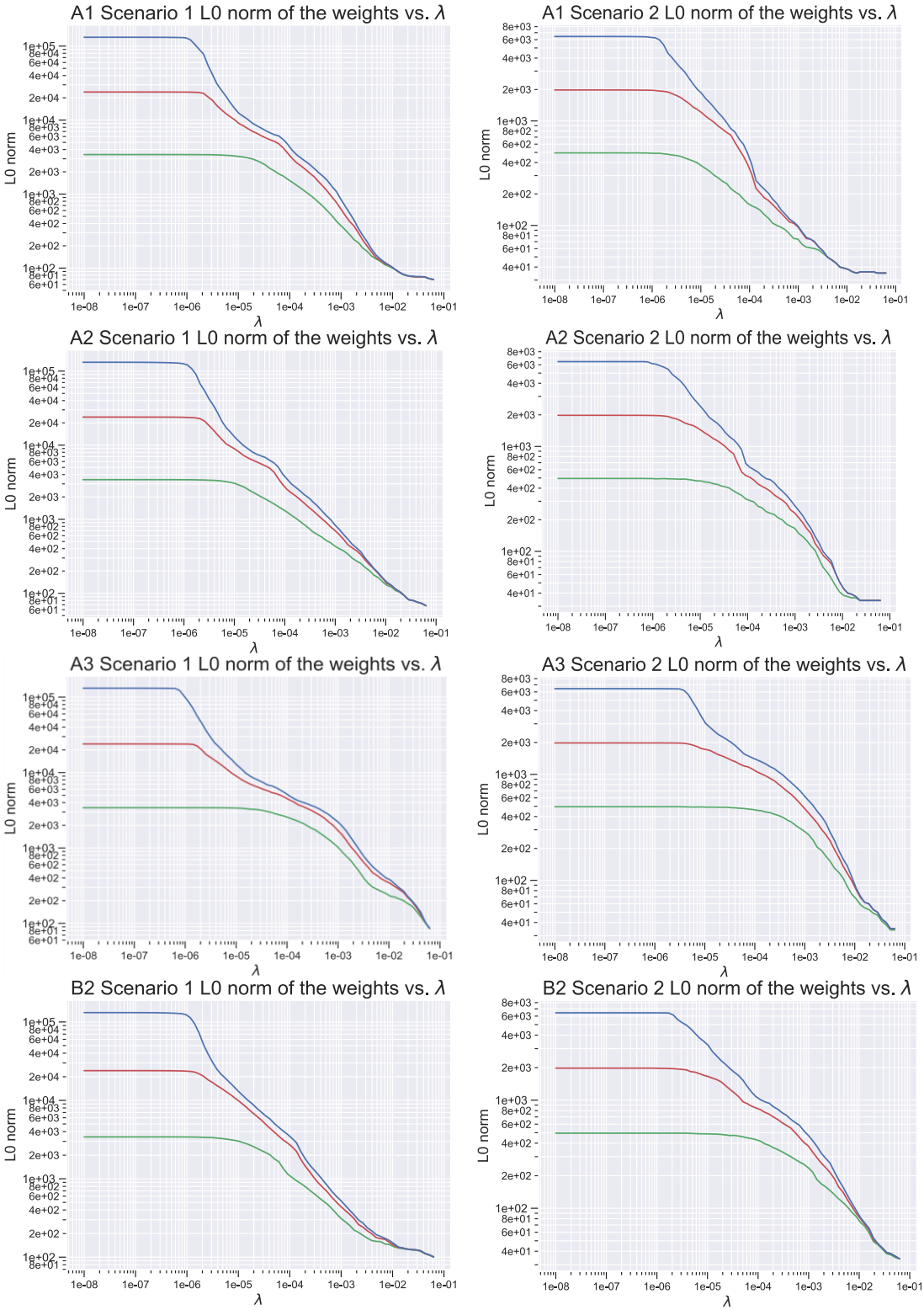


Figure 26: L_0 Norm of the weights vs. λ for LASSO regression with polynomials of degree 2 (green), degree 3 (red) and degree 4 (blue).

It is seen that the number of non-zero weights decreases as λ is increased. This is a contrast to ridge regression where the L_0 norm of the weights did not change with λ. This means that for ridge regression, none of the weights were set to zero as a result of regularization.

4.5.3.4 Mapping $\mathcal{P} \mapsto \mathcal{G}$

The $\mathbf{R}_g$ residual was studied for the higher order models with LASSO regression and the MSR values are presented below in Figure 27.

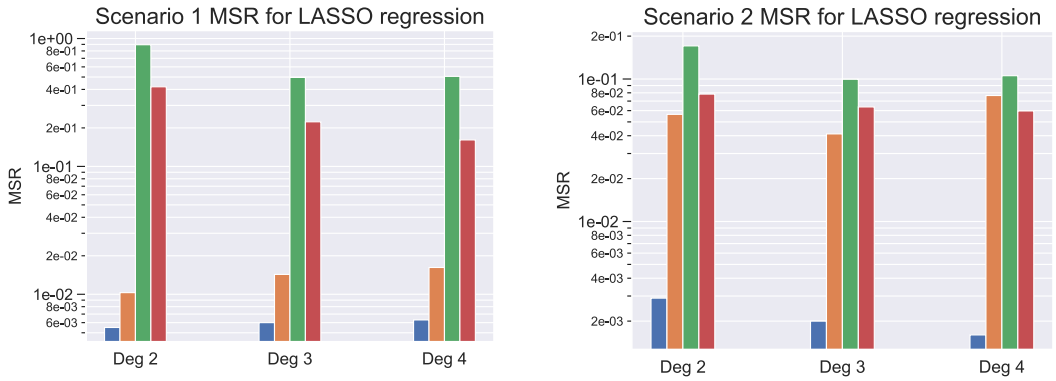


Figure 27: MSR of $\boldsymbol{R}_g$ for LASSO regularized models of varying orders for scenarios 1 and 2. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively.

It is seen that $\mathbf{R}_g$ MSR values using LASSO regularization is substantially better than OLS and moderately improved over ridge regression. This could be attributed to some of the weights being set to zero and therefore only the parameters that effect the gain most is used for prediction by the model.

4.5.3.5 Mapping $\mathcal{G} \mapsto \mathcal{P}$

Higher order models were studied with LASSO regularization for the mapping $\mathcal{G} \mapsto \mathcal{P}$ and the results are presented in terms of the MSR values of $\mathbf{R}_p$ and $\mathbf{R}_g^*$ in Figure 28 and Figure 29.

$\mathbf{R}_p$

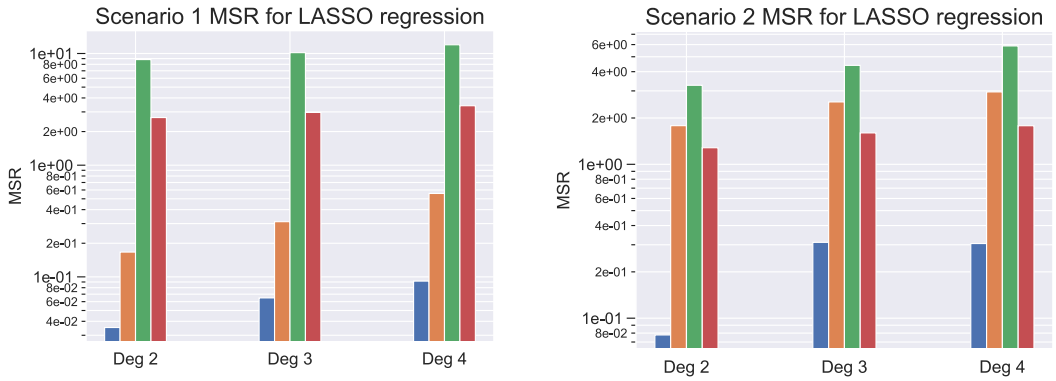


Figure 28: MSR of $\boldsymbol{R}_p$ for LASSO regularized models of varying orders for scenarios 1 and 2. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively.

It is seen that the MSR values of $\mathbf{R}_p$ are consistently better than OLS and ridge regression for scenario 1. However, ridge regression produced better MSR for the 2nd degree polynomial for A1 and A2 for scenario 2.

$\mathbf{R}_g^*$

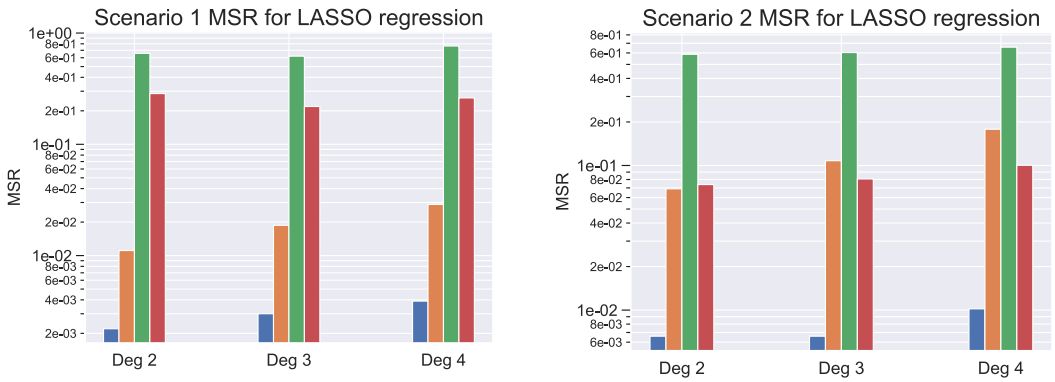


Figure 29: MSR of $\boldsymbol{R}_g^*$ for LASSO regularized models of varying orders for scenarios 1 and 2. The MSR for A1, A2, A3 and B2 are represented in blue, orange, green and red respectively.

It is seen that LASSO regression model produced better $\mathbf{R}_g^*$ MSR than OLS for all cases and better than ridge regression for all cases except scenario 2 A1 for the polynomial of order 2.

4.6 Benchmarking

The performance of the models that produced the lowest MSR for both scenarios and each device/parameter region combination, was benchmarked against the current models used by CBAS for the same purpose. It should be noted the benchmarking was carried out using a script and the official CBAS fitting software was not used. $\mathbf{R}_g$ and $\mathbf{R}_g^*$were used as the benchmarking criteria. In order to make the comparison fair, the exact same data points were used for training and testing by both models. The models chosen for $\mathbf{W}$ and $\widetilde{\mathbf{W}}$ for all device/parameter region combinations were based on the validation scores produced. The MSR values obtained from benchmarking is shown below in Figure 30.

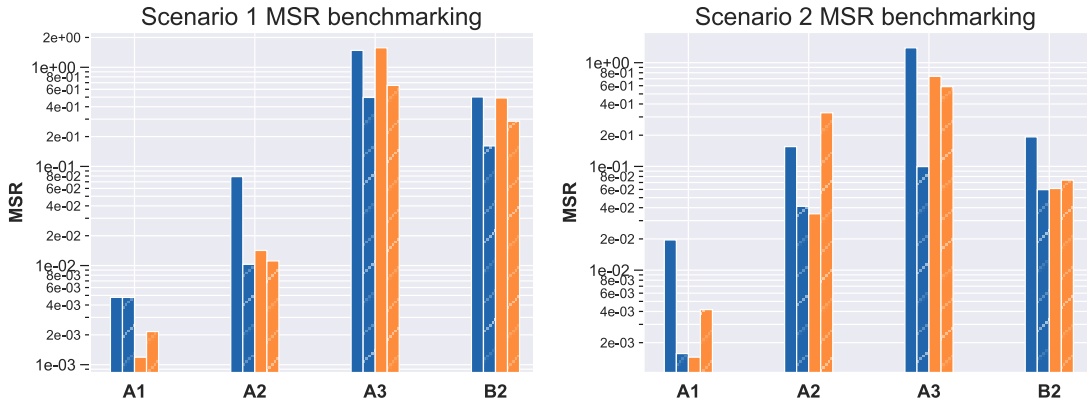


Figure 30: Benchmarking the models obtained from this thesis against CBAS's current methods. The MSR values of $\boldsymbol{R}_g$ (blue)and $\boldsymbol{R}_g^*$ (orange) are given for both scenarios for all device/parameter combinations. The hatched and the unhatched bars indicate MSR values produced by the "new models" and "current models" respectively. "New models" refers to models proposed in this thesis and "current models" refers to models currently used by CBAS.

It can be observed that the $\mathbf{R}_g$ residual is consistently better for the new models for both scenarios. For scenario 1, the new model also performs better in terms of $\mathbf{R}_g^*$ residual in all cases except A1. However, for scenario 2, the $\mathbf{R}_g^*$ residual is only better for the new model in 1 case out of 4.

5. Conclusion

It can be concluded that the project was executed successfully as machine learning based on regression was applied to the fitting of a BAHA. Linear, affine and higher order models were fitted to the measured data. Ridge and LASSO regularization were applied in an effort to mitigate problems associated with overfitting. The results for the regularized models were quantified in terms of the MSR of the validation data and the norm of weights. The norms applied to the weights were L_2, L_1 and L_0. In comparison to OLS models and ridge regression models, LASSO regression was found to be particularly effective in terms of reducing the validation data MSR and the number of non-zero weights. The inverse mapping was studied with linear and affine models and comparisons were made between models obtained using analytical expressions and "training from scratch". Higher order OLS and regularized models were applied to the inverse mapping and LASSO regression was found to have the best results for the inverse mapping. From benchmarking, the methods proposed in this thesis was found to perform better in most cases than the current methods used by CBAS.

6. References

[1] World Health Organisation, “Deafness and hearing loss,” 2019. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss. [Accessed: 10-Sep-2019].

[2] International Insititute of Medicine and National Research Council, *Hearing Loss and Healthy Aging : Workshop Summary*. Washngton, D.C: National Academies Press, 2014.

[3] B. C. J. Moore, *Cochlear Hearing Loss*, Second. West Sussex, England: John Wiley & Sons, Ltd, 2007.

[4] G. R. Popelka, B. C. J. Moore, R. R. Fay, and A. N. Popper, Eds., *Hearing Aids*, vol. 56. Cham: Springer International Publishing, 2016.

[5] L. Vestergaard Knudsen, M. Öberg, C. Nielsen, G. Naylor, and S. E. Kramer, “Factors Influencing Help Seeking, Hearing Aid Uptake, Hearing Aid Use and Satisfaction With Hearing Aids: A Review of the Literature,” *Trends Amplif.*, vol. 14, no. 3, pp. 127–154, Sep. 2010.

[6] D. G. Blazer, S. Domnitz, and C. T. Liverman, Eds., *Hearing Health Care for Adults*. Washington, D.C.: National Academies Press, 2016.

[7] N. A. Lesica, “Hearing Aids : Limitations and Opportunities,” *Hear. J.*, vol. 71, no. 5, p. 43, May 2018.

[8] N. A. Lesica, “Why Do Hearing Aids Fail to Restore Normal Auditory Perception?,” *Trends Neurosci.*, vol. 41, no. 4, pp. 174–185, Apr. 2018.

[9] R. Dauman, “Bone conduction: An explanation for this phenomenon comprising complex mechanisms,” *Eur. Ann. Otorhinolaryngol. Head Neck Dis.*, vol. 130, no. 4, pp. 209–213, Sep. 2013.

[10] J. Colquitt *et al.*, “Bone-anchored hearing aids (BAHAs) for people who are bilaterally deaf: a systematic review and economic evaluation,” *Health Technol. Assess. (Rockv).*, vol. 15, no. 26, Jul. 2011.

[11] A. F. Snik, E. A. Mylanus, and C. W. Cremers, “The bone-anchored hearing aid compared with conventional hearing aids. Audiologic results and the patients’ opinions.,” *Otolaryngol. Clin. North Am.*, vol. 28, no. 1, pp. 73–83, Feb. 1995.

[12] Medical Advisory Secretariat, “Bone anchored hearing aid: an evidence-based analysis.,” *Ont. Health Technol. Assess. Ser.*, vol. 2, no. 3, pp. 1–47, 2002.

[13] S. A. Gelfand, *Essentials of Audiology*, 4th ed. New York: Thieme Medical Publishers, Incorporated, 2016.

[14] American National Standards Institute, *ANSI S3.22-2003: Specification of Hearing Aid Characteristics*. New York: Acoustical Society of America, 2003.

[15] A. B. Williams, *Analog Filter and Circuit Design Handbook*. New York: McGraw-Hill Education, 2014.

[16] S. A. Tretter, *Communication System Design Using DSP Algorithms with Laboratory Experiments for the TMS320C6713™ DSK*. Boston, MA: Springer US, 2008.

[17] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM J. Res. Dev.*, vol. 44, no. 1/2, pp. 206–226, 2000.

[18] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine Learning*. Boca Raton : CRC Press, 2017.: CRC Press, 2016.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY: Springer New York, 2009.

[20] A. M. Mathai and H. J. Hauboid, *Linear Algebra: A course for physicists and engineers*. Berlin: Walter de Gruyter GmbH, 2017.
[21] Z. Byliskii, "Generalized Linear Regression with Regularization." 2015.
[22] R. C. Penny, *Linear Algebra : Ideas and Applications*, 4th ed. New Jersey: John Wiley & Sons, 2015.
[23] F. E. Szabo, *The Linear Algebra Survival Guide*. Elsevier, 2015.
[24] S. Theodoridis, *Machine Learning : A Bayesian and Optimization Perspective*. London: Elsevier, 2015.
[25] R. R. Rhinehart, *Engineering Optimization : Applications, Methods, and Analysis*. New Jersey: John Wiley & Sons Ltd., 2018.
[26] R. Hamming, *Numerical Methods for Scientists and Engineers*, 2nd ed. New York: Dover Publications, 1973.
[27] S. Aksoy and R. M. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," *Pattern Recognit. Lett.*, vol. 22, no. 5, pp. 563–582, Apr. 2001.
[28] S. Kaufman, S. Rosset, and C. Perlich, "Leakage in data mining : Formulation, Detection, and Avoidance," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*, 2011, p. 556.
[29] F. Öztürk and F. Akdeniz, "Ill-conditioning and multicollinearity," *Linear Algebra Appl.*, vol. 321, no. 1–3, pp. 295–305, Dec. 2000.

Appendix A: Affine model R_g histogram

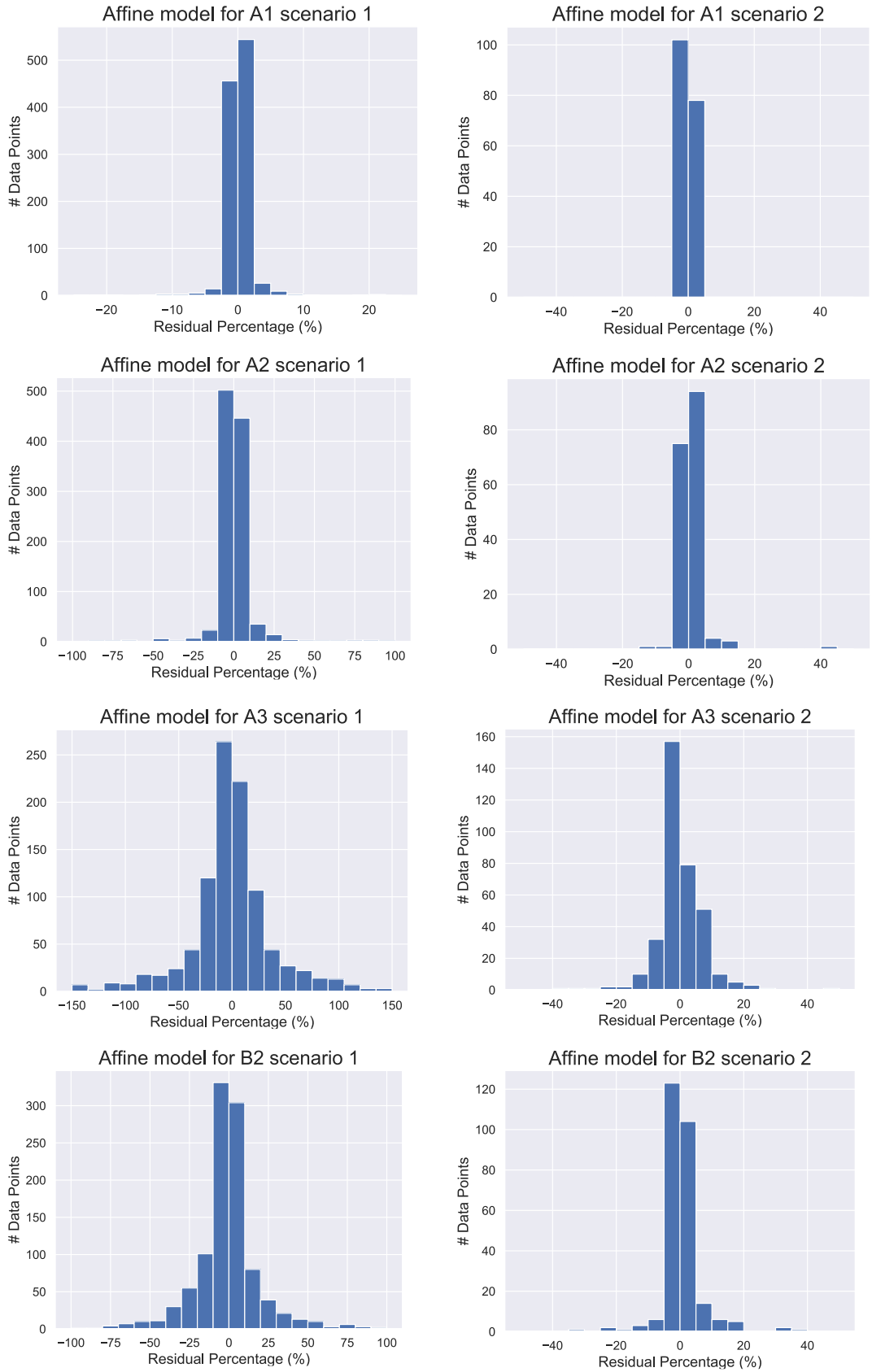


Figure 31: Histograms of residual percentage of all device/parameter region combinations for scenario 1 and scenario using the affine model. These models are producing the mapping $\mathcal{P} \mapsto \mathcal{G}$.

Appendix B: Affine model R_p histogram

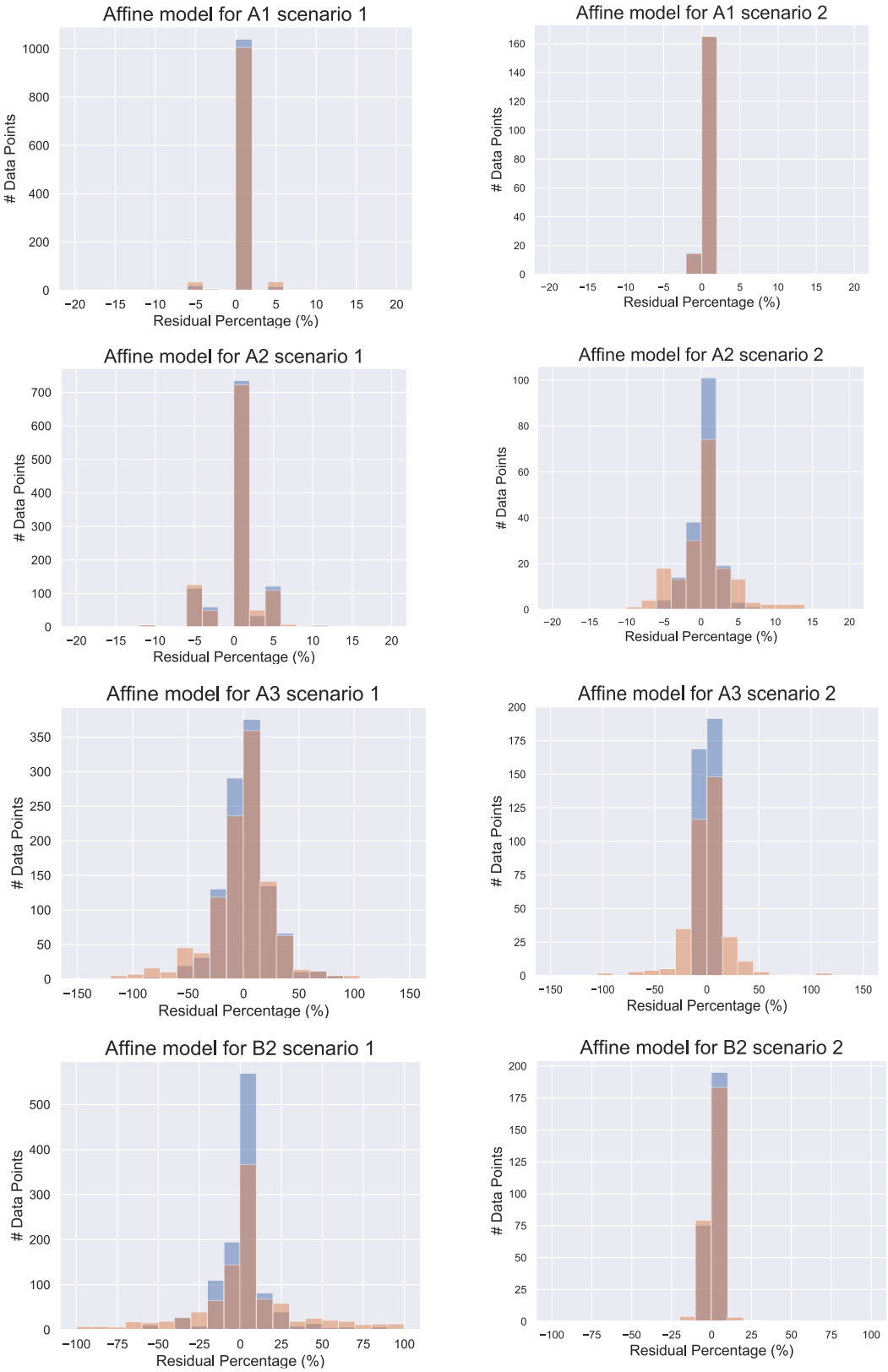


Figure 32: Residual percentages expressed as histograms for affine models that map $\mathcal{G} \mapsto \mathcal{P}$. The results produced by $\widetilde{\boldsymbol{W}}$ (obtained by training from scratch) and $\widetilde{\boldsymbol{W}}^*$(obtained by inverting $\boldsymbol{W}$) are shown in blue and orange respectively.

Appendix C: Ridge - training data MSR vs λ

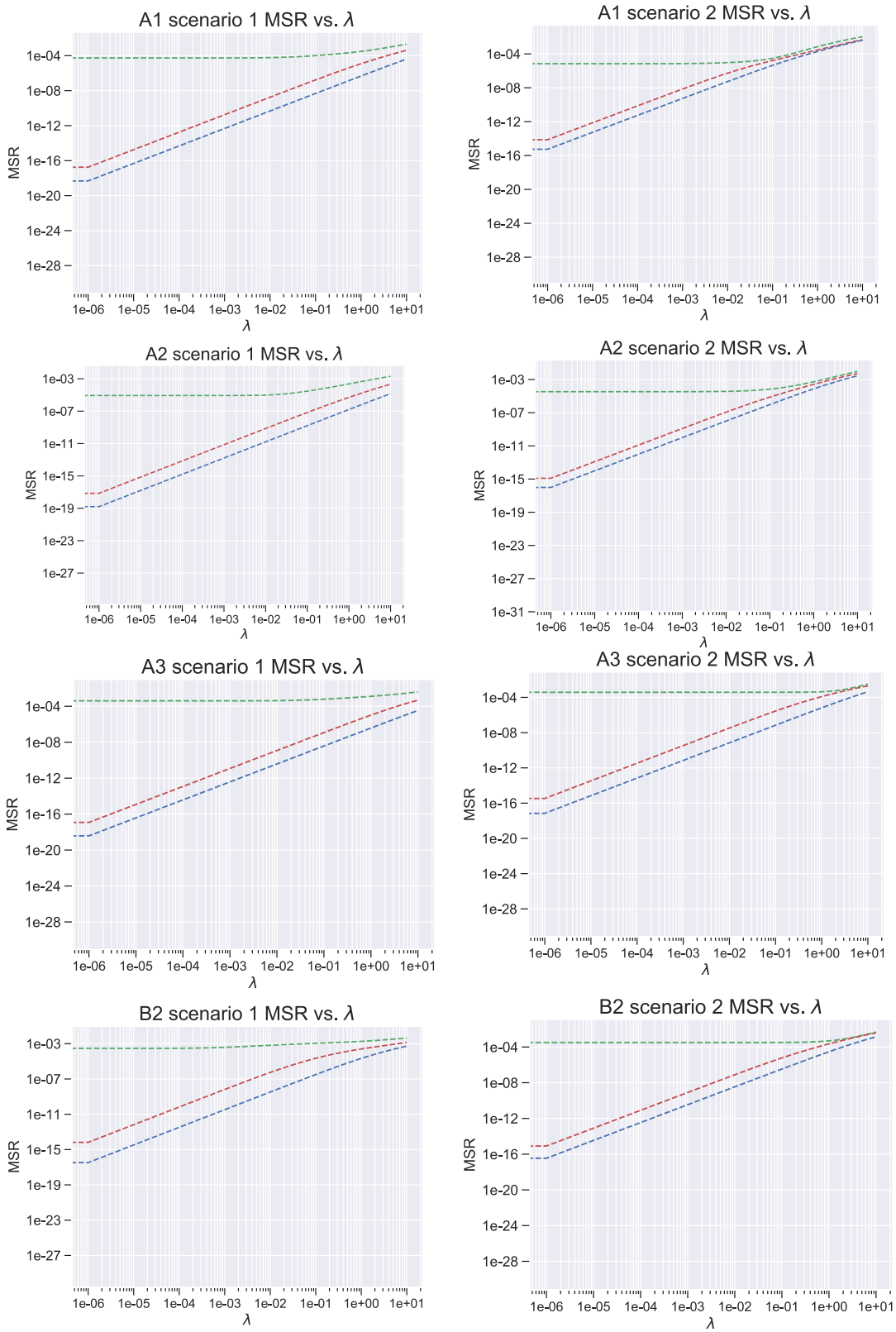


Figure 33: MSR vs. λ for ridge regression with polynomials of degree 2 (green), degree 3 (red) and degree 4 (blue) for training data.

Appendix D: Ridge - R_g histogram

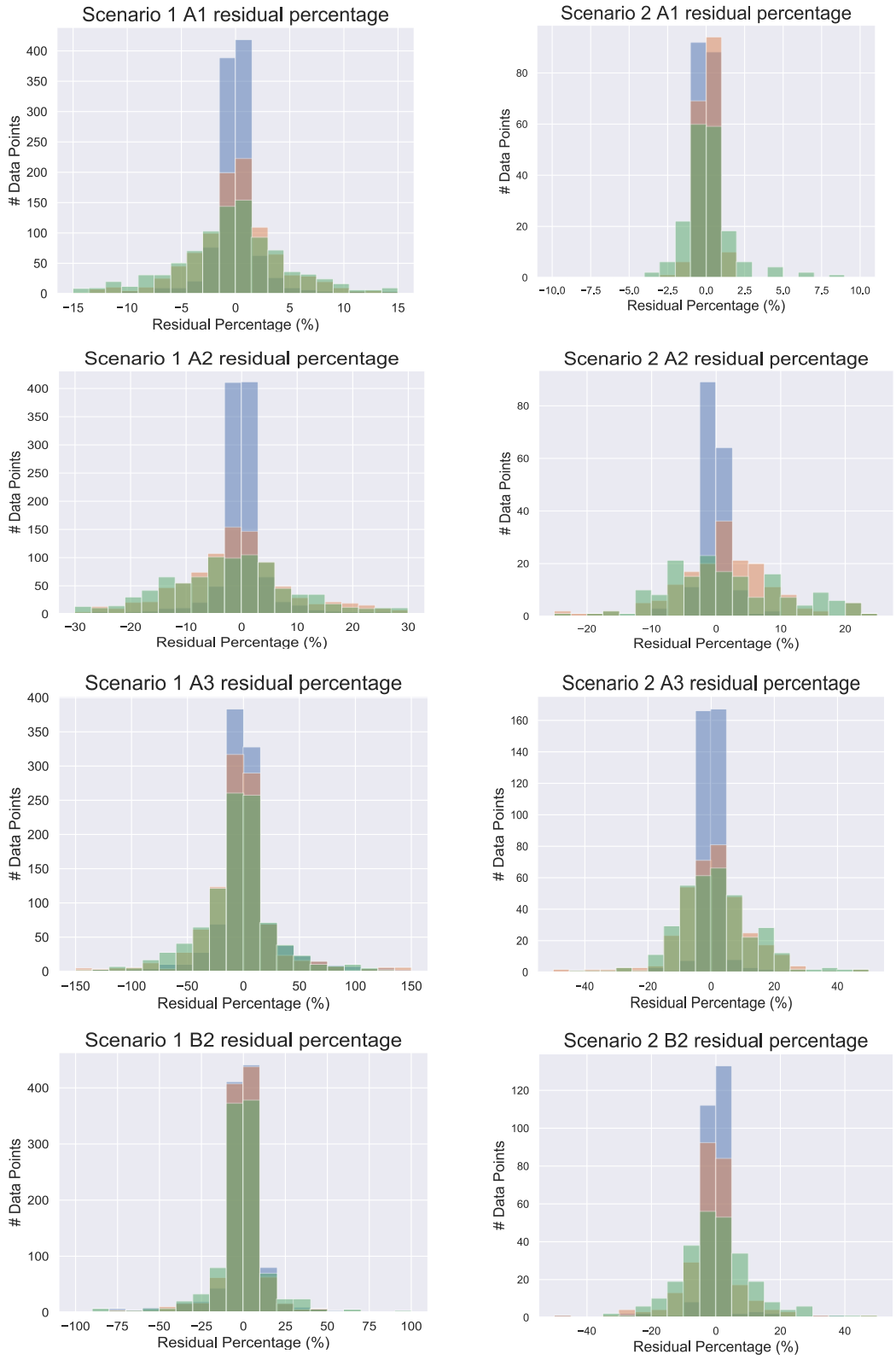


*Figure 34: **R_g** residual percentage histogram for ridge regularized models of order 2 (blue), 3 (orange) and 4 (green) for scenarios 1 and 2 for all device / parameter range combinations.*

Appendix E: Ridge - R_p histogram

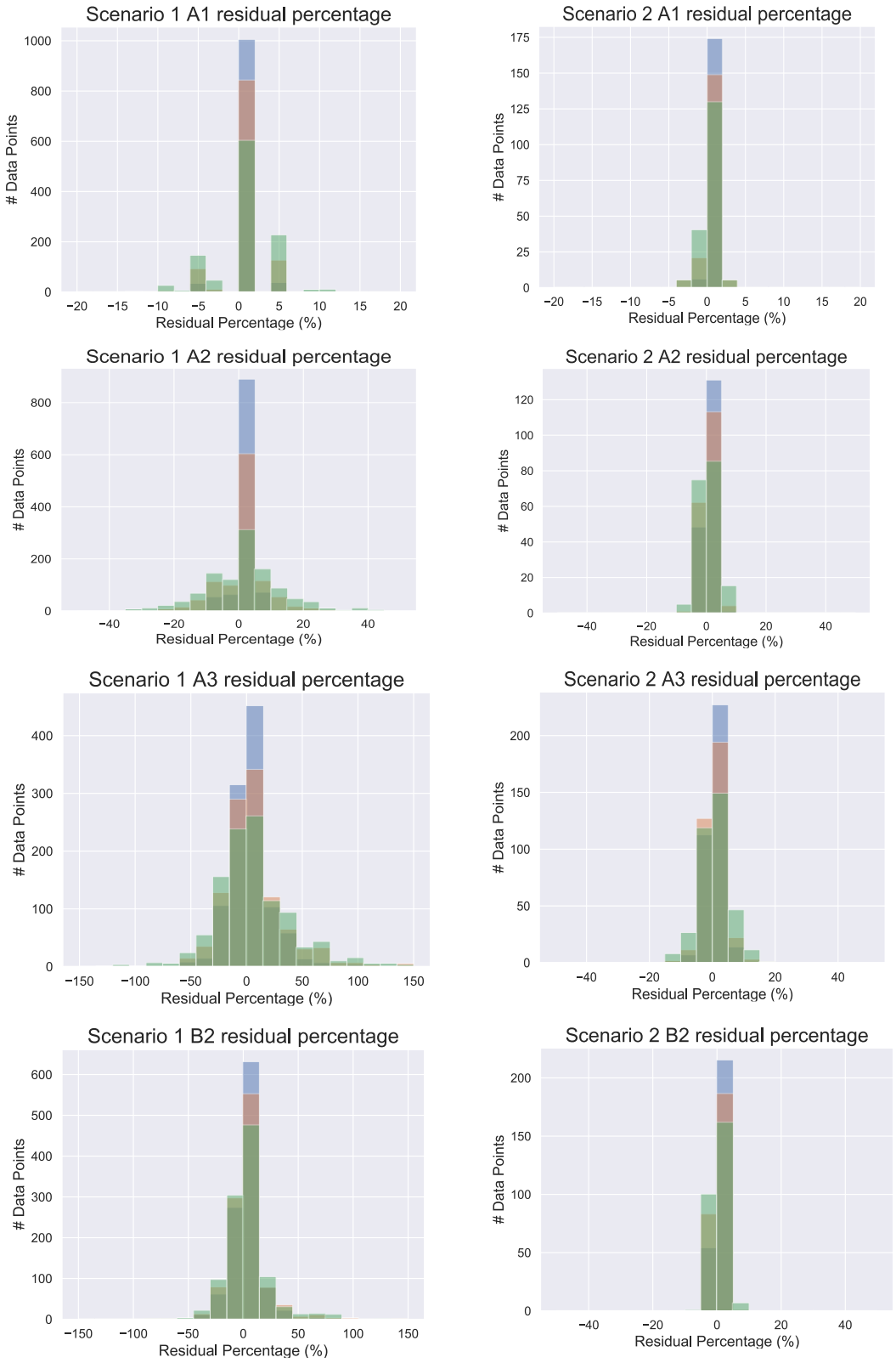


Figure 35: R_p residual percentage histogram for ridge regularized models of order 2 (blue), 3 (orange) and 4 (green) for scenarios 1 and 2 for all device / parameter range combinations.

Appendix F: LASSO - R_g histogram

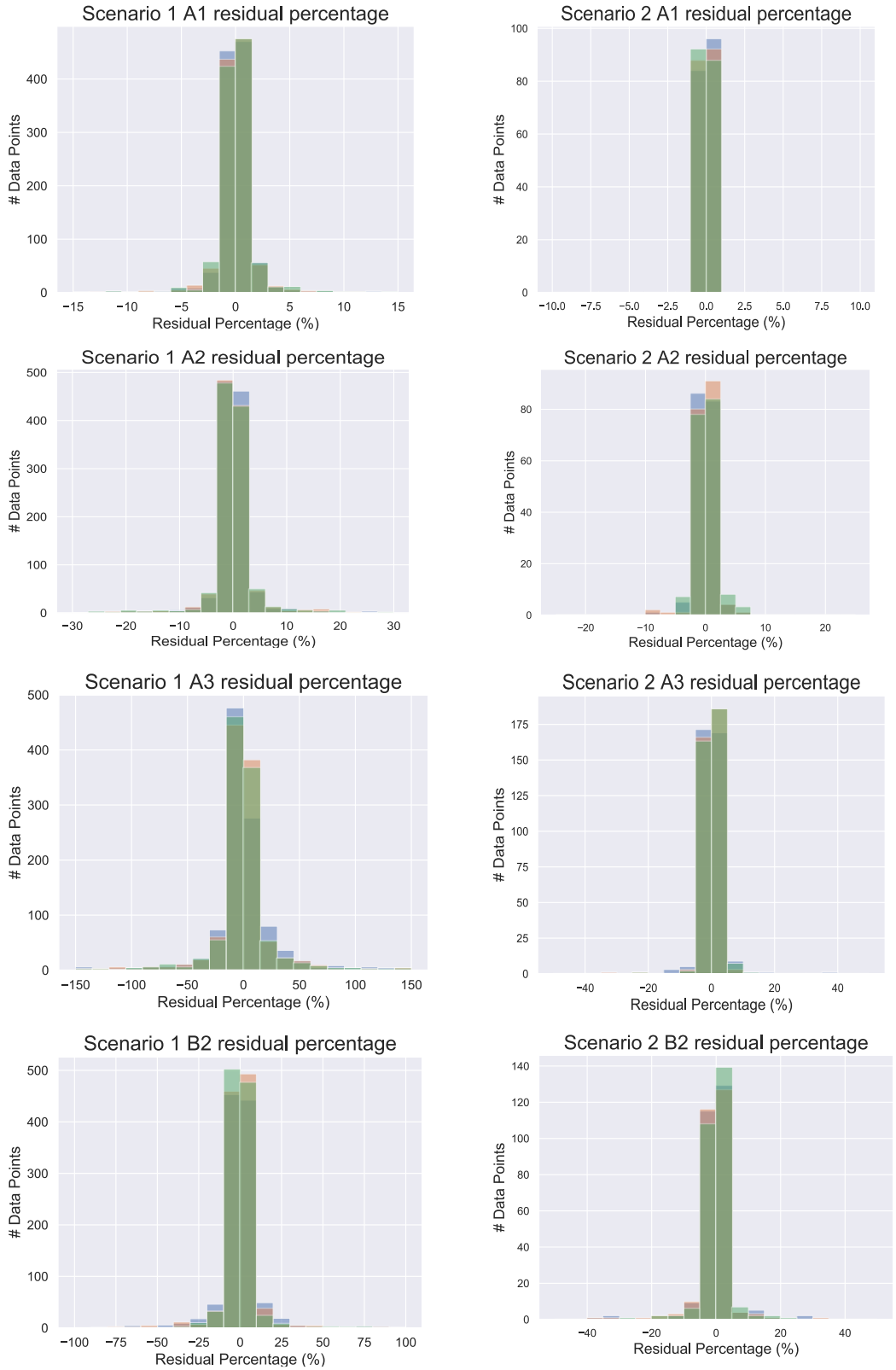


*Figure 36: **R_g** residual percentage histogram for LASSO regularized models of order 2 (blue), 3 (orange) and 4 (green) for scenarios 1 and 2 for all device / parameter range combinations.*

Appendix G: LASSO - R_p histogram

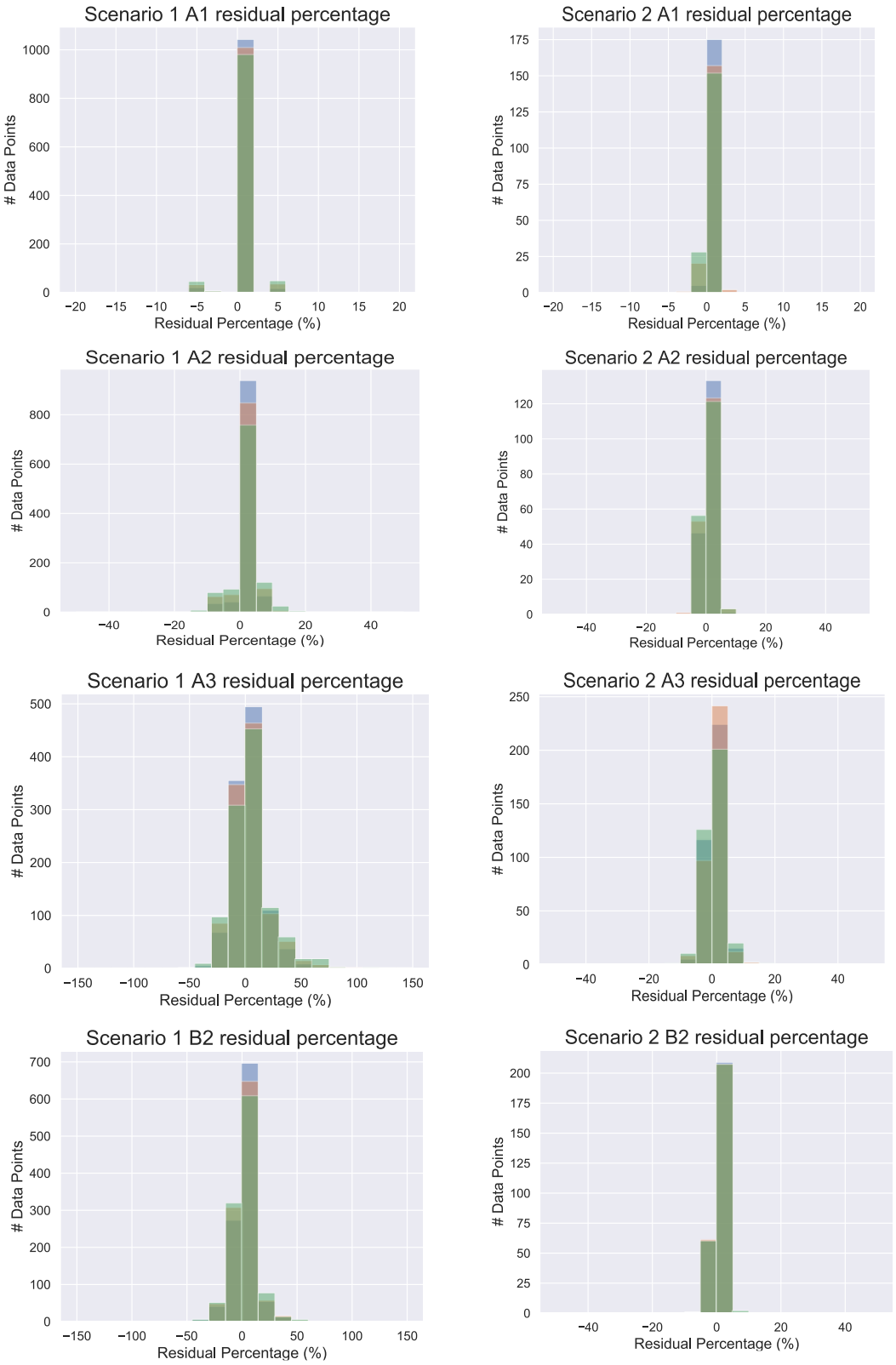


Figure 37: R_p residual percentage histogram for LASSO regularized models of order 2 (blue), 3 (orange) and 4 (green) for scenarios 1 and 2 for all device / parameter range combinations.