Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2021

Data-driven Strain Sensor Modelling in Mining Applications

Artificial strain sensors for material fatigue estimation

Alex Rydén and Mattias Langsér

LINKÖPING UNIVERSITY

Master of Science Thesis in Electrical Engineering

Data-driven Strain Sensor Modelling in Mining Applications: Artificial strain sensors for material fatigue estimation

Alex Rydén and Mattias Langsér

LiTH-ISY-EX--21/5404--SE

Supervisor: **Max Johansson**
ISY, Linköpings universitet
Robert Pettersson
Epiroc Rock Drills AB

Examiner: **Erik Frisk**
ISY, Linköpings universitet

Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Copyright © 2021 Alex Rydén and Mattias Langsér

Acknowledgments

We would like to thank our supervisor at the university, Max Johansson, for keeping this thesis on track and for answering our many questions. We would also like to thank Robert Petterson and Erik Jacobsson at Epiroc, for the many fruitful and sometimes challenging discussions about the topic of material fatigue estimation. Lastly, we would like to thank our examiner, Erik Frisk, for sharing his expertise and giving us valuable insights on how to approach large problems.

Linköping, June 2021
Alex Rydén *Mattias Langsér*

Abstract

When boring machines are used, large loads are exerted on their structure. The load cycles cause material fatigue on the boring machine structure. If the material fatigue can be estimated in real-time, maintenance can be planned more efficiently and the effect of different types of usage can be evaluated. Because of the many advantages of knowing the material fatigue, the goal of this thesis is to develop a model to predict the strain of a boring machine structure and then derive an estimate of the material fatigue caused by the strain. To do this several approaches using machine learning techniques are evaluated. The input signals were selected using both coherence analysis and mutual information. It was found that linear models outperform the tested non-linear model structures, and that non-linear mechanical connections cause difficulties. The signals to be modelled contained high frequency components that were not present in the available input signals. The results show that given favorable sensor positions, an estimate of the material fatigue can be made with sufficient accuracy when using a noise model and noise realization to cover the non-existent high frequency components.

Contents

Abbreviations and Nomenclature

Abbreviations

CV	Cross validation
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
HS	High frequency System
LOOCV	Leave-One-Out Cross Validation
MIMO	Multi-input multi-output system
MSE	Mean Squared Error
RCS	Rig Control System
ReLU	Rectified Linear Unit
RUL	Remaining useful life
S-N	Stress to number of cycles
SGD	Stochastic Gradient Descent
SSE	Sum of squared errors

Nomenclature

α_k	The weight of node k in a neural network layer
α_{reg}	Parameter tuning the regularization in TC-kernels
β_k	The bias of nodes in a neural network layer

γ_k	The output weight of a neural network layer
$\hat{\lambda}_e$	Estimated noise variance
$\hat{y}(t)$	Estimated output signal
κ	The activation function in a neural network layer
λ_{lev}	Scaling of the Levenberg-Marquardt regularization
λ_{reg}	Scaling of the ridge regularization
μ	Step-length used in parameter estimation of linear models
Φ	The spectrum of a signal
Π	Covariance matrix of the parameter vector
θ	Vector of model parameters
ε	Prediction error
φ	Predictor holding input signals
A^N	Matrix holding network weights
A^{par}	Matrix holding model parameters
B^N	Vector holding network biases
B^{par}	Matrix holding model parameters
C^N	Vector holding network output weights
C^{par}	Matrix holding model parameters
D^{par}	Matrix holding model parameters
d_N	The amount of nodes in a neural network layer
F^N	Non-linear function representing a layer in a neural network
F^{par}	Matrix holding model parameters
F_T^{cdf}	Cumulative distribution function of T-distribution
L	Number of layers in a neural network
n^p	Number of samples used in the correlation test
N_b	Number of bins used in distribution estimation
N_d	Number of data points in each synchronization segment
N_i	The number of cycles to failure at a specified stress index

n_i	The amount of cycles at the specified stress amplitude
N_s	Number of data points
n_x	Distribution bin of x
n_y	Distribution bin of y
N_{LOOCV}	Number of sections used in LOOCV
P	Probability of a certain bin
P_x	Probability of a certain bin
P_y	Probability of a certain bin
Q_{norm}	Set of normalization constants
r^p	Pearson's correlation coefficient
T^p	T distribution test score
$y^*(t)$	True output signal
D_m	Cubic mean of load cycles
DMG	Accumulated Damage
e	Noise term
F_{el}	Equivalent fatigue load
N_{des}	Number of cycles to be designed for when calculating F_{el}
Q	Regularization term
q	Time shift operator
R	Regularization weight matrix
S	Search direction modifier.
t	Time
T*	Measurement time
T_{des}	Life time to be designed for when calculating F_{el}
u	Input signal
V	Cost function
x	Dummy variable used in presentation of equations
y	Dummy variable used in presentation of equations

1 Introduction

1.1 Background

Epiroc is a global producer of rock excavation products. One of these products is the Mobile Miner 40V, which is a tunnel boring machine. When the Mobile miner excavates rock, large forces and structural stress is exerted on the machine. To increase cost efficiency and prolong the life of the machine, it is important to understand how the usage affects the structure. In order to help operators use the machine in the best possible way with respect to the fatigue of the machine, to plan maintenance and more, there is an interest to investigate possibilities of simulating the strain imposed on the structure of the Mobile miner and estimate the material fatigue, using already existing on-board control signals.

1.1.1 Mobile Miner 40V

The Mobile Miner is a tunnel boring machine that is used in mining applications that require a tunnel size of approximately 4x4 meters. The machine has a large body with a length of 21 meters. It carries a vertical cutter wheel on a large boom, see Figure 1.1. The boom has an adjustable angular setting, specifically the tilt and swing angle of the boom, as shown in Figure 1.2 and 1.3. The tilt and swing angles are adjusted using the lift cylinder and the swing cylinders, which are placed above the boom and on the sides of the boom, respectively. The boom is fastened on a smaller frame, the thrust-frame, which in turn is mechanically connected to the main frame with guide rails. The boom and thrust-frame can be extended and retracted using a thrust cylinder. The thrust, tilt and swing cylinders are shown in Figure 1.2. A conveyor belt that collects and carries away excavated rock is mounted beneath the cutter wheel.

Figure 1.1: *The Mobile Miner 40V*

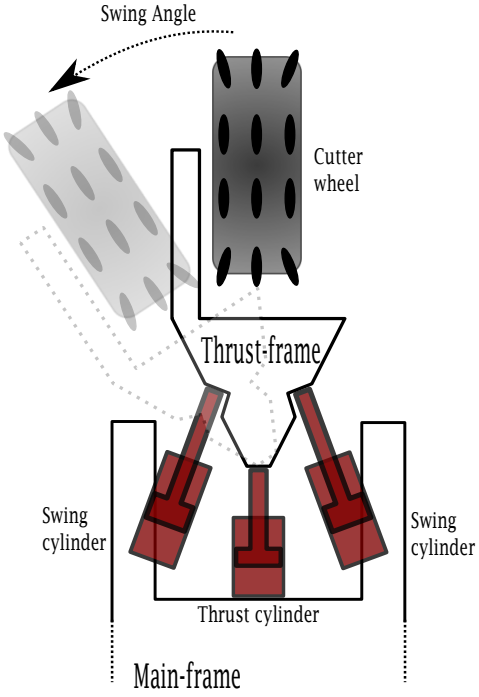


Figure 1.2: *A schematic topview of the mobile miner, with relevant parts named.*

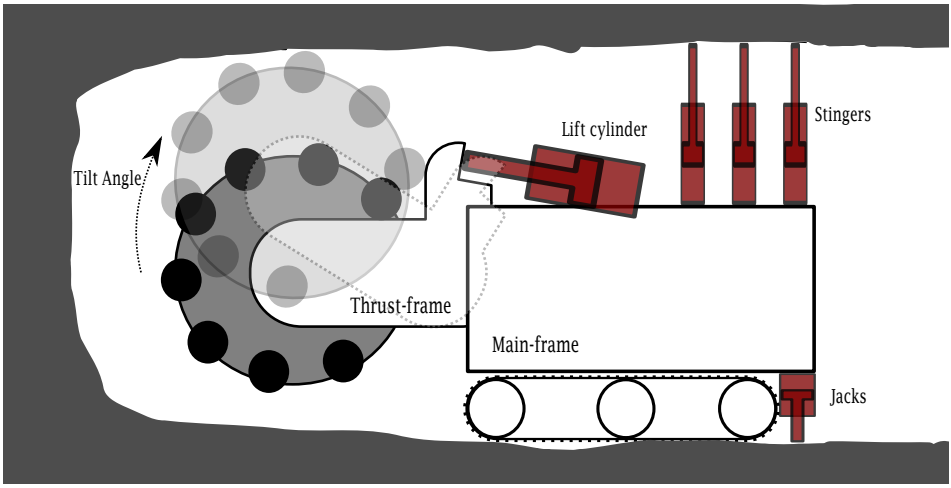


Figure 1.3: *A schematic sideview of the mobile miner, with relevant parts named.*

1.1.2 Excavation cycle

An excavation cycle consists of several steps. Before the actual excavation starts, the machine has to secure itself from moving during the excavation. It does so by extending hydraulic cylinders, stingers and jacks, against the rock above and below itself as show in Figure 1.3. When the machine is secured the cutter wheel is adjusted to the desired position. Following this the cutter wheel starts to rotate and the machine thrusts it into the tunnel face. When the thrust cylinder is fully extended it retracts and a stroke has been completed. The cutter wheel's setting is then adjusted and another stroke is carried out. This is repeated until an even tunnel face is achieved. Then the stingers and jacks are retracted and the complete carriage advances in the tunnel using its tracks. When the carriage has advanced a length equivalent to the stroke length the whole procedure is repeated. A typical excavation procedure to reach an even tunnel face is illustrated in Figure 1.4.

1.1.3 Sensors and signals

In order to learn about how the usage affects the structure, a Mobile Miner has been operating in a test mine equipped with several sensors of different kinds. The Mobile Miner is originally equipped with sensors associated with the control functions of the machine. These sensors measure hydraulic pressures, extension lengths of cylinders and body and boom angular settings. In addition to these, more sensors have been installed to provide valuable information about strain, shear and acceleration imposed on various structural parts of the machine. The additional sensors enable developers to draw conclusions about material wear and structural health. However, the sensors are expensive and to avoid additional

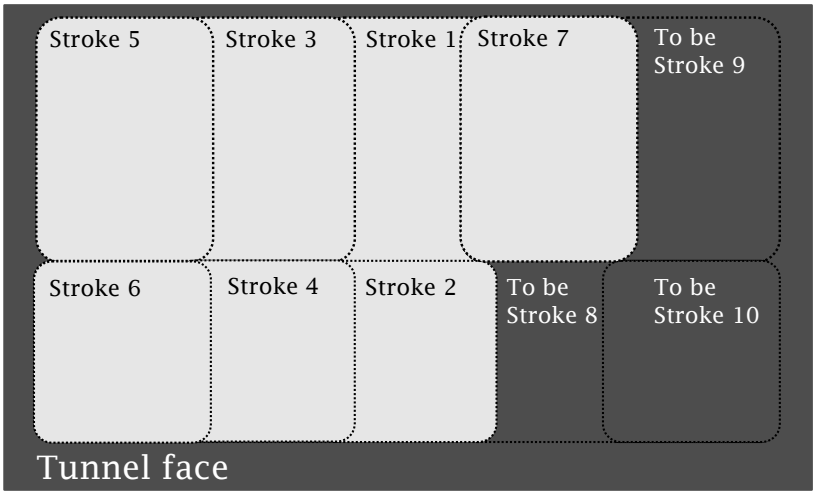


Figure 1.4: *A figure of a tunnel face excavation procedure from the machine's point of view. It showcases that the cutter wheel has a different contact with the rock each stroke.*

costs of installation and maintenance of the strain gauges Epiroc wants to model them using the originally existing sensors.

1.1.4 Available Data set

A data set has been collected during consecutive excavation sessions over one year. The excavation sessions have a large variation in terms of excavation settings such as angular settings and rotational speed of the cutter wheel. The data set can generally be divided into two parts. One part containing high frequency signals, sampled with a frequency of 5000 Hz. These signals will be referred to as HS-signals. The HS-signals are signals from the sensors added with the purpose of aiding future development, and are thus not originally a part of the machine. The second part of the data set contains the low frequency signals, sampled with a frequency of 12.5 Hz. These signals will be referred to as RCS-signals. The RCS-signals are a part of the control system and are thus sampled from sensors originally mounted on the machine.

There are 54 and 44 existing HS-signals and RCS-signals respectively. This value varies as some data sets are incomplete. Because of processing time in the measurement systems, clock drifting is present between the HS- and RCS-sets. As previously mentioned, the HS-set of signals contains strain and shear sensors, accelerometers and also some pressure sensors. The pressure sensor measurements also exist in the RCS-set, which enables synchronization and provides other possibilities which are discussed later on. Out of the 54 existing HS-signals, 6 strain signals are particularly informative regarding future development and will thus be the focus during the rest of the work. Another signal that is interesting to

model is F_lift, which is created using available pressure signals in the HS-set. F_lift are derived for X, Y and Z directions, and the total force. Table 1.1 and Table 1.2 present the HS- and RCS-signals respectively. The pressure signals PRE_3_1 - PRE_3_8 are available in both the HS- and RCS-data sets.

Table 1.1: *Table of relevant signals in the HS-data set.*

HS-signals		
Signal name	Placement	Function
EPS_2_1	In proximity of the guide rail connecting the frames	Measures strain in proximity of the guide rail
EPS_2_3	In proximity of the guide rail connecting the frames	Measures strain in proximity of the guide rail
EPS_2_7	In the rear of the main frame	Measures strain on the main frame
EPS_2_9	In proximity of the right swing cylinder attachment to the main-frame	Measures strain in proximity of the attachment
EPS_2_28	In proximity of the left swing cylinder attachment to the main-frame	Measures strain in proximity of the attachment
F_lift_tot	-	X, Y and Z directional forces in the lift cylinder. Derived from pressure signals

Table 1.2: *Table of relevant signals in the RCS-data set.*

RCS-signals		
Signal name	Placement	Function
PRE_3_1, PRE_3_2	On the cutter wheel thrust cylinder	Measures the pressures on the rod and piston side respectively when the cutter wheel is thrusted forward
PRE_3_3, PRE_3_4	On the cutter wheel lift cylinder	Measures the pressures on the rod and piston side respectively when the cutter wheel is lifted
PRE_3_5, PRE_3_6	On the right cutter wheel swing cylinder	Measures the pressures on the rod and piston side respectively when the cutter wheel performs a swing movement
PRE_3_7, PRE_3_8	On the left cutter wheel swing cylinder	Measures the pressures on the rod and piston side respectively when the cutter wheel performs a swing movement
PRE_3_9, PRE_3_10, PRE_3_11, PRE_3_12	On the right and left jack cylinders	Measures the pressures on the rod and piston side respectively when the machine secures itself into the rock
PRE_3_13, PRE_3_14, ⋮ PRE_3_26	On the stingers	Measures pressures on the rod and piston side when the machine secures itself up into the rock
PRE_3_27AB, PRE_3_28AB	On the pumps supplying the hydraulic motors with pressure	Measures pressures on the high and low side of the two pumps
ANG_4_1	Boom	Measures the lift angle of the boom
ANG_4_2	Boom	Measures the swing angle of the boom
ANG_4_3	Cutter wheel	Measures the rotational speed of the cutter wheel
ANG_4_4	Main body	Measure the pitch of the main body
ANG_4_5	Main body	Measures the roll of the main body
DIS_5_1	Boom	Measures the stroke length
DIS_5_2	Stinger	Measure the extension of the stinger
DIS_5_3	Stinger	Measure the extension of the stinger

1.2 Problem formulation

The problem this thesis is faced with is to develop a data driven model for the strain sensors fastened on different structural parts on the Mobile Miner and the derived force signal, all listed in Table 1.1, by using the control system data from the test mine. The strain estimations should also be used to calculate the material fatigue on the various structural parts.

To accomplish this the following questions will be answered:

Do, and if so which, signals carry relevant information about the strain and force imposed on the structure of the Mobile Miner?

Is it possible to recreate the strain and force signals with a data-driven model using the signals existing in the control system of the Mobile Miner as input?

Is it possible to accurately estimate the accumulated damage and equivalent load on the Mobile Miner's structural parts with the created strain and force signals models?

1.3 Related Work

This work touches on several topics. A major topic is remaining useful life (RUL) estimations, along with the topic of system identification. Related work covering direct RUL estimations are presented and variants where a strain signal first is estimated are also presented.

Ramon Fuentes et al. used Gaussian process regression to map several flight parameters, such as indicated air speed and lateral acceleration to a corresponding strain signal. The resulting prediction of the strain signal was then used to derive the accumulated damage, and the performance was evaluated using mean squared error and the ratio between estimated accumulated damage and the true accumulated damage. The results showed that the methodology worked well while the aircraft operated in ranges covered by the training data [6].

Erik Jakobsson et al. faced a similar problem of estimating the accumulated damage of a mining vehicle. It was found that a segmentation of usage scenarios was effective to simplify the modelling condition, and the accumulated damage was successfully estimated when using a linear model structure on segmented scenarios. While the models were successful at their main task, slower dynamics, or trends, were harder to capture with the models. The problem of selecting adequate input signals was solved by using the coherence measure [12].

Sangram Patil et al. states that the procedure of creating a mathematical model of a bearing degradation process is too complex, and approaches the problem using ensemble regression techniques. The problem of selecting relevant input signals was solved by an iterative search using cross validation, that is a model was trained with specific inputs and the performance was then evaluated and the input's impact was ranked thereafter. It was found that the Gradient Boosting

technique was superior to the Random Forest, and the future potential of the technique for estimating Remaining Useful Life (RUL) was pointed out [25].

Vimala Mathew et al. conducted a large survey on the applicability of several different machine learning techniques to estimate the RUL of a turbofan engine. Ten different techniques were evaluated and while many of them performed well, it was found that the Random Forest performed the best [20].

1.4 Delimitations and limitations

1.4.1 Limitations

There is limited data on the mining conditions in the test mine, thus the model does not account for effects associated with it. This may reduce the model accuracy outside similar mining conditions.

1.4.2 Delimitations

This thesis will not investigate alternative methods for damage accumulation estimation other than the method described in Section 2.1, which has been used in similar work [12].

This thesis will not investigate alternative methods to data-driven models.

This thesis will only focus on the signals mentioned in Table 1.1 and Table 1.2.

1.4.3 Assumptions

It is assumed that all signals within the same excavation session, the HS- or RCS-set, are correctly synchronized.

2
Theory

In this chapter the theory necessary to follow the methodology later presented is described. The chapter is divided into three sections describing how the accumulated damage is derived from a strain signal, the theory behind the signal selection and lastly a description of the modelling techniques.

2.1 Damage accumulation and equivalent fatigue load

Damage accumulation is very useful for estimating the remaining useful life of a machine. There are numerous ways to estimate damage, or equivalently material fatigue, on structural parts. Fatigue is caused by cyclic loading at different stress levels below the materials ultimate tensile material strength. Repeated cycles will cause microscopic cracks in the material which will eventually cause failure. A cycle is defined as two half-cycles where a half-cycle is the consecutive pair of two extremas in a load signal, which can be compared to one period of a sinus signal.

One commonly used method to determine material fatigue is cycle-counting [4]. The basis of cycle counting lies in the fact that a typical material has a finite number of load cycles it can withstand at any given stress range. This is often represented in the form of a stress to number of cycles curve, a S-N curve, as shown in Figure 2.1. In real life applications, the stress cycles are stochastic and not at a constant amplitude. Stress cycles can be counted with the help of the rainflow-counting method [22]. In rainflow-counting, load cycles are defined at various amplitudes and then the amount of cycles for each load cycle are counted for the stress-time signal. The cycles are then sorted in to bins to form a spectrum.

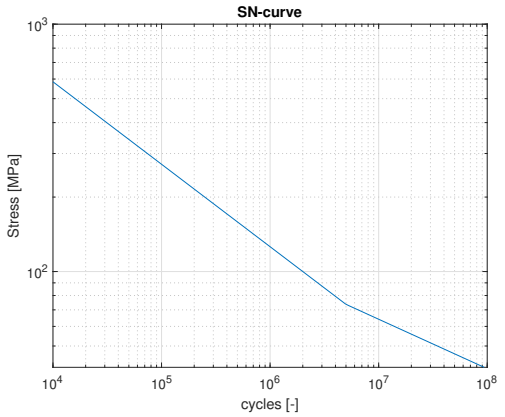


Figure 2.1: *The S-N curve used for this work. It represents the amount of cycles to failure at a given stress range.*

Given the information from the rainflow-counting method combined with the knowledge of how many cycles to reach failure obtained from the S-N graph, the damage accumulation can be estimated per Palmgren-Miner's rule as follows:

$$DMG = \sum_{i=1}^{k} \frac{n_i}{N_i}, \tag{2.1}$$

where n_i is the amount of cycles at the specified stress amplitude index (i), N_i the respective number of cycles to failure at the specified index (i) and k is the amount of cycles in each bin. DMG is the accumulated damage ranging from 0 to 1, where one corresponds to material failure [4].

F_lift_tot does not measure strain, but instead measures the directional forces on a specific part. On this signal the equivalent fatigue load, F_{el}, is instead calculated which then can be used with other methods to analyse fatigue. The equivalent fatigue load is calculated by taking mean load cycle where the load cycles for the data is calculated using rainflow counting, this is then normalized over a specified preset usage time and number of cycles. F_{el} is defined as

$$F_{el} = D_m \left(\frac{T_{el} n^*}{N_{el} T^*} \right)^{1/3}, \tag{2.2}$$

where n^* is the amount of cycles and T^* is the time length of the F_lift signal. T_{el} is the specified life time for the design and N_{el} is the amount of cycles to completed within T_{el}. D_m is the cubic mean of F_lift after rainflow counting has been applied.

2.2 Signal Selection

2.2.1 Cross and auto covariance

In order to quantify the strength of the linear relationship between two signals, x and y, or one signal, x, when one of them is shifted τ time steps, the cross covariance and auto-covariance are used. They are estimated as in (2.3a) and (2.3b), respectively:

$$\hat{R}_{xy}^{N_s}(\tau) = \frac{1}{N_s} \sum_{t=1}^{N_s} y(t)x(t-\tau), \tag{2.3a}$$

$$\hat{R}_{x}^{N_s}(\tau) = \frac{1}{N_s} \sum_{t=1}^{N_s} x(t)x(t-\tau), \tag{2.3b}$$

where N_s is the sample size [16].

2.2.2 Correlation

To determine if a linear relationship exists and if it is strong enough to take into account, a significance test is performed. The significance test is done based on Pearson's correlation coefficient r^p which measures the linear correlation between two signals, a hypothesis test is then performed to decide if the linear relationship has statistical significance. The hypothesis test is based on the T^p-statistic defined as

$$T^p = \frac{r^p \sqrt{n^p - 2}}{\sqrt{1 - r^{p2}}}. \tag{2.4}$$

With knowledge of T^p the p-value can be calculated by using the cumulative distribution function of the T-distribution, F_T^{cdf}, as given by (2.5):

$$p = 2(1 - F_T^{cdf}(T^p, n^p - 2)). \tag{2.5}$$

The p-value describes the probability of attaining the correlation by random chance. If the p-value is less than a threshold $\alpha = 0.05$ then the correlation is deemed significant [11].

2.2.3 Coherence analysis

Coherence analysis is a tool to determine if a linear relationship between two signals exists, as well on what frequency span it exists. The coherence function is a normalized version of the cross spectrum and has a range between 0 and 1, where 1 corresponds to a full description between signals. It is derived using the cross spectrum, Φ_{xy}, and the spectrum of the signals, Φ_{xx} and Φ_{yy}, as given by (2.6):

$$C(\omega) = \frac{|\Phi_{xy}(\omega)|^2}{\Phi_{xx}(\omega)\Phi_{yy}(\omega)} \in [0, 1]. \tag{2.6}$$

The spectrum and cross-spectrum are the discrete fourier transforms (DFT) of the auto and cross-covariance functions. The cross spectrum and spectrums are smoothed using Welch's method which averages the periodograms over several sections of itself [8].

2.2.4 Mutual information

Mutual information is a method to find general, both linear and non-linear, relationships between signals [1]. It is defined in as

$$MI(x,y) = \int\int p(x,y) log_2 \frac{p(x,y)}{p_x(x)p_y(y)} dxdy, \tag{2.7}$$

where $p(x,y)$ is the joint distribution of the signals x and y and $p_x(x)$ and $p_y(y)$ the marginal distribution of x and y, respectively. Since we do not initially know the distribution of the signals, the mutual information is estimated using bins as in (2.8) [14],

$$MI(x,y) \approx MI_{bin}(x,y) = \sum_{i=1}^{N_b}\sum_{j=1}^{N_b} P(i,j) log_2 \frac{P(i,j)}{P_x(i)P_y(j)}. \tag{2.8}$$

The data points that x and y holds are portioned into a N_b amount of uniformly distributed bins. These bins are then used to approximate $P_x(i)$ and $P_y(j)$ as follows:

$$P_x(i) \approx \frac{n_x(i)}{N_s}, \tag{2.9a}$$

$$P_y(j) \approx \frac{n_y(j)}{N_s}, \tag{2.9b}$$

where N_s is the amount of samples and $n_x(i)$ and $n_y(j)$ are the i:th and j:th bin for x and y, respectively. $P(i,j)$ is derived as the intersection between these bins, that is for each x data point falling into $n_x(i)$ ($i = 1, ..., N_b$), determine the distribution of the corresponding y data points collected in the same time instants. The N_b distributions derived will collectively represent $P(i,j)$.

2.3 Model structures

2.3.1 Linear Models

The linear models presented in this section have various structures, and are chosen because of their frequent use [18]. An overview of the models are given in Figure 2.2.

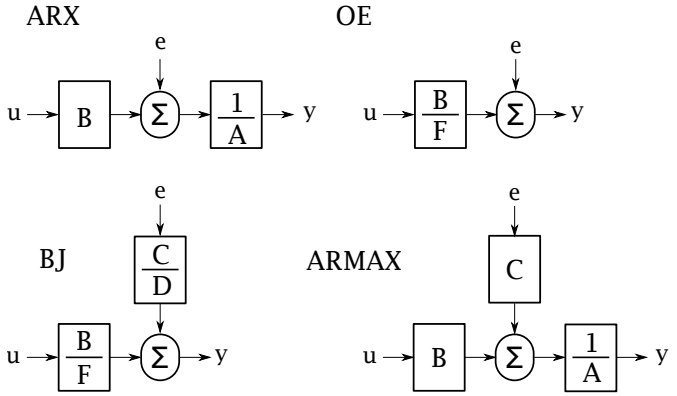


Figure 2.2: *Overview of the linear model types depicted in a block scheme. u denotes the input signals, e the noise signal and y the output.*

ARX

The ARX-model is the simplest of the linear models evaluated. In a multi-input multi-output system (MIMO), with $y(t) \in \mathbb{R}^{n_y}$ and $u(t) \in \mathbb{R}^{n_u}$, it has the form formulated in (2.10), (2.11a) and (2.11b). Its foremost advantage is its simplicity and the ease to estimate the parameters. It has the disadvantage that the systems poles are shared with the noise dynamics. This means that if the signal to noise ratio is low, the estimate may be incorrect [18]. The model is given by

$$\mathbf{A}(q)y(t) = \mathbf{B}(q)u(t) + e(t), \tag{2.10}$$

where q is the time shift operator. The $\mathbf{A}(q)$ and $\mathbf{B}(q)$ polynomials have the following forms:

$$\mathbf{A}(q) = I + A_1^{par} q^{-1} + ... + A_{na}^{par} q^{-na}, \tag{2.11a}$$

$$\mathbf{B(q)} = B_1^{par} q^{-nk-1} + ... + B_{nb}^{par} q^{-nk-nb}, \tag{2.11b}$$

where nk specifies input delay and na and nb specifies the polynomial orders, and consequently the model order. I is the identity matrix and $A_i^{par} \in \mathbb{R}^{n_y \times n_y}$ and $B_i^{par} \in \mathbb{R}^{n_y \times n_u}$ holds the model parameters, θ, and are estimated using the available data and techniques described later in Section 2.3.1.

ARMAX

The second model is the ARMAX-model. It is depicted in (2.12). It is an extension of the ARX model which provides more flexibility since it includes an additional polynomial when estimating the noise dynamics. Since the ARMAX- and ARX-model structures lumps the process and the noise dynamics together, they are preferable when noise enters early in the process, that is when noise are present in the input.

$$\mathbf{A}(q)y(t) = \mathbf{B}(q)u(t) + \mathbf{C}(q)e(t), \tag{2.12}$$

where $\mathbf{C}(q)$ is formed as in (2.13), with nc specifying noise polynomial order and $C_i^{par} \in \mathbb{R}^{n_y \times n_y}$.

$$\mathbf{C}(q) = I + C_1^{par} q^{-1} + ... + C_{nc}^{par} q^{-nc}. \tag{2.13}$$

OE

The OE-model is defined in (2.14) and (2.15), with $F_i^{par} \in \mathbb{R}^{n_y \times n_u}$. The model structure has the advantage that the system dynamics are described separately from the noise. It results in that the system between the inputs and output can be obtained. This means that frequency content present in the HS-set that is not present in the RCS-set will be disregarded along with possible noise when the model is estimated.

$$y(t) = \frac{\mathbf{B}(q)}{\mathbf{F}(q)} u(t) + e(t), \tag{2.14}$$

$$\mathbf{F}(q) = F_1^{par} q^{-1} + ... + F_{nf}^{par} q^{-nf}. \tag{2.15}$$

BJ

The BJ-model is the most general model, and is capable of estimating the noise model and system model separately. This means that dynamics outside the frequency range of the RCS-set will be modelled as filtered white noise. It is defined as follows:

$$y(t) = \frac{\mathbf{B}(q)}{\mathbf{F}(q)} u(t) + \frac{\mathbf{C}(q)}{\mathbf{D}(q)} e(t), \tag{2.16}$$

$$\mathbf{D}(q) = I + D_1^{par} q^{-1} + ... + D_{nd}^{par} q^{-nd}, \tag{2.17}$$

where $D_i^{par} \in \mathbb{R}^{n_y \times n_y}$.

Forming a prediction

All previously mentioned models fit in (2.18), depending on how one forms $\mathbf{G}(q)$ and $\mathbf{H}(q)$.

$$y(t) = \mathbf{G}(q)u(t) + \mathbf{H}(q)e(t). \tag{2.18}$$

By pre-multiplication of both sides with $\mathbf{H}^{-1}(q)$ and removing the noise component, the following prediction of y is formed as a function of inputs and previous outputs:

$$\hat{y}(t) = (I - \mathbf{H}^{-1}(q))y(t) + \mathbf{H}^{-1}(q)\mathbf{G}(q)u(t). \tag{2.19}$$

Which can be rewritten to highlight that the prediction is a simulation with an addition of previous simulation errors scaled with the noise model [18]:

$$\hat{y}(t) = \mathbf{G}(q)u(t) + \Big(I - \mathbf{H}^{-1}(q)\Big)\Big(y(t) - \mathbf{G}(q)u(t)\Big). \tag{2.20}$$

Parameter estimation

When estimating the parameters of the models, the goal is to minimize a cost function formulated using the prediction error. The prediction error is formulated as follows:

$$\varepsilon(t,\theta) = y^*(t) - \hat{y}(t|\theta), \tag{2.21}$$

where $y^*(t)$ is the measured output, $\hat{y}(t|\theta)$ the predicted output and θ the model parameters. By using the prediction error, a quadratic cost function can be formed as

$$V_{N_s}(\theta) = tr\Big(\frac{1}{2N_s}\sum_{t=1}^{N_s}\varepsilon(t,\theta)\varepsilon^T(t,\theta)\Big), \tag{2.22}$$

where N_s is the amount of samples. If the case is that the ARX model structure is used, the value of θ that minimizes (2.22) can be found analytically and has the following solution:

$$\hat{\theta}^{ARX}_{N_s} = \Big(\frac{1}{N_s}\sum_{t=1}^{N_s}\varphi(t)\varphi^T(t)\Big)^{-1}\frac{1}{N_s}\sum_{t=1}^{N_s}\varphi(t)y^T(t), \tag{2.23}$$

where $\varphi(t)$ is called the predictor and holds all input signals in time t [16]. If one of the more complex model structures are used, ARMAX, OE or BJ, finding the model parameters is an iterative search for the minimum. The minimum is found by solving (2.24).

$$\frac{d}{d\theta}V_{N_s}(\theta) = 0. \tag{2.24}$$

A general family of solvers find this by updating the parameters as follows:

$$\hat{\theta}^{(i+1)}_{N_s} = \hat{\theta}^{(i)}_{N_s} - \mu^{(i)}_{N_s}[S^{(i)}_{N_s}]^{-1}V'_{N_s}(\hat{\theta}^{(i)}_{N_s}) \tag{2.25}$$

Where i denotes the i:th iteration, μ the step-length and S_{N_s} a search direction modifier which is formulated differently depending on the used solver. V' is the first order-derivative of the cost function, and is formulated as given by (2.26), using (2.21) and (2.22):

$$\begin{aligned} V'_{N_s}(\theta) = {} & \frac{-1}{N_s}\sum_{i=1}^{N_s}\frac{d}{d\theta}\hat{y}_1(t_i|\theta)(y_1(t_i) - \hat{y}_1(t_i|\theta)) + ... \\ & ... + \frac{d}{d\theta}\hat{y}_{n_y}(t_i|\theta)(y_{n_y}(t_i) - \hat{y}_{n_y}(t_i|\theta)) = \\ & \frac{-1}{N_s}\sum_{i=1}^{N_s}\sum_{k=1}^{n_y}\frac{d}{d\theta}\hat{y}_k(t_i|\theta)(y_k(t_i) - \hat{y}_k(t_i|\theta)). \end{aligned} \tag{2.26}$$

The simplest choice of S_{N_s} is the identity matrix I, which makes (2.25) the steepest-descent method, which works well but is inefficient close to the minimum. Another choice is the Levenberg-Marquardt procedure. It selects S_{N_s} as the approximated and regularized hessian of the cost function as in (2.27). λ_{lev} controls the

step-length and how much the step-direction turns towards the gradient. If λ_{lev} is set to zero the search procedure is called Gauss-Newton. Gauss-Newton works sufficiently well in most cases, but in some cases numerical problems are avoided using the Levenberg-Marquardt procedure [16].

$$S_{N_s}^{(i)}(\lambda_{lev}) = \frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{k=1}^{n_y}\left(\frac{d}{d\theta}\hat{y}_k(t_i|\theta)\right)\left(\frac{d}{d\theta}\hat{y}_k(t_i|\theta)\right)^T + \lambda_{lev}I. \tag{2.27}$$

Regularization

When estimating the parameters θ of a model one can take the model flexibility into account. That is how much freedom exists when estimating the parameters. A flexible model reduces the bias but increases the variance of the estimate. This means that a flexible model has an easier time estimating complex behaviour but loses generality and exposes itself to the risk of adapting too much to present measurement noise [3]. A way to control the flexibility of the model is by introducing a regularization term to the cost function in (2.22) as in (2.28). The regularization term adds a cost to the size of the parameters in the model. A commonly used form of the term is ridge regularization, which is expressed as in (2.29). It applies the product of the l_2-norm of the parameters, a scalar λ and a weighting matrix R, where a large λ reduces the variance but increases the bias and vice versa [13].

$$V_{N_s}^{reg}(\theta) = V_{N_s}(\theta) + Q(\lambda, \theta, R), \tag{2.28}$$

$$Q(\lambda, \theta, R) = \frac{1}{N_s}\lambda\theta R\theta^T. \tag{2.29}$$

The regularization term changes the solution for the parameters of the ARX-model and the first order derivate of the cost function as shown in (2.30) and (2.31), respectively.

$$\hat{\theta}_{N_s}^{ARX} = \left(\frac{1}{N_s}\sum_{t=1}^{N_s}\varphi(t)\varphi^T(t) + Q\right)^{-1}\frac{1}{N_s}\sum_{t=1}^{N_s}\varphi(t)y^T(t), \tag{2.30}$$

$$V_{N_s}^{'reg}(\theta) = V_{N_s}'(\theta) + Q' = V_{N_s}'(\theta) + \frac{1}{N_s}\sum_{k=1}^{N_\theta}2\lambda\theta, \tag{2.31}$$

where (2.31) holds when R is chosen as the identity matrix, which is called ridge regression [13].

Under the assumption that the system has a smooth and exponentially decaying finite impulse response, the regularization of an ARX model may be systematically selected as the covariance matrix of the parameter vector. That is, for a parameter vector θ, the elements k, j in Q in (2.29) may be selected as

$$\Pi_{k,j}(\alpha) = C_{reg}\min(\lambda^k, \lambda^j);\ \alpha_{reg} = |C_{reg}, \lambda|, \tag{2.32}$$

where α_{reg} is selected as the minimizing argument to the maximum likelihood estimate of Π [3].

Stopping criterium

A stopping criterium is what stops the otherwise endless iterative update of the parameters as in (2.25). The stopping criterium can be a max amount of iterations, i, or a minimum allowed improvement each iteration.

Simulation of Models

When a linear model has been created, it is used to produce estimates of an output signal by using a set of input signals and an estimated noise signal as input. In a scenario where no output signals are available as measurements, previous estimates are fed into the model with the simple scheme in Figure 2.3.

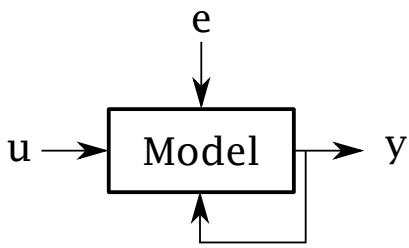


Figure 2.3: *Figure of how the simulation uses previous estimates and input.*

The noise signal e is assumed to have a gaussian distribution with zero mean and a covariance estimated as follows:

$$\hat{\lambda}_e = \frac{1}{N_{s,tr}} \sum_{k=1}^{N_{s,tr}} \varepsilon \varepsilon^T = \frac{1}{N_{s,tr}} \sum_{k=1}^{N_{s,tr}} \begin{pmatrix} \left(y_1 - \hat{y}_1\right)^2 & \cdots & \left(y_1 - \hat{y}_1\right)\left(y_{n_y} - \hat{y}_{n_y}\right) \\ \vdots & \ddots & \\ \left(y_1 - \hat{y}_1\right)\left(y_{n_y} - \hat{y}_{n_y}\right) & & \left(y_{n_y} - \hat{y}_{n_y}\right)^2 \end{pmatrix}, \tag{2.33}$$

where $N_{s,tr}$ is the amount of samples in the training set and y and $\hat{y}$ exists in each time-step t_k.

Residual analysis

When a linear model has been made and an estimate produced, the residual formed between the estimate and the true value in (2.21) contains information. If the model perfectly describes the system the residual is completely independent of the input to the system. This can be investigated by forming the cross-covariance between the residual and each input signal and see if it is close enough to zero over inspected lags. If the cross-covariance largely deviates from zero it indicates that there are system dynamics that the model did not capture properly

and that the model can be improved. The auto-covariance of the residual can also be investigated. If the model and noise dynamics are perfectly modelled the residual is white noise, and the auto-covariance will thus produce a dirac signal. The cross-covariance and auto-covariance are determined in Section 2.2.1. $\hat{R}_{\varepsilon u}^{N_s}$ and $\hat{R}_{\varepsilon}^{N_s}$ naturally varies and thus deviates from zero to some extent. To determine whether the deviation is because of an actual correlation or not, two confidence bounds are formed and used as a comparison in the residual plots:

$$\left|\hat{R}_{\varepsilon u}^{N_s}(\tau)\right| \leq \sqrt{\frac{P_1}{N_s}} N_\alpha, \tag{2.34a}$$

$$\hat{R}_{\varepsilon}^{N_s}(\tau) \leq \frac{N_\alpha}{\sqrt{N}}, \tag{2.34b}$$

where (2.34a) tests (2.3a) and (2.34b) tests (2.3b). N_α denotes the α-level of the $N(0,1)$ distribution. P_1 is defined as following:

$$P_1 = \sum_{k=-M}^{M} R_\varepsilon(k) R_u(k), \tag{2.35}$$

where $R_\varepsilon(k)$ and $R_u(k)$ are the covariance functions of the residual and input, respectively [16]. M is chosen as the plot window size.

2.3.2 Neural networks

A neural network can be a function mapping from a d_0-dimensional input vector x to a scalar y and is in the case of a single linear output layer formulated as in (2.36).

$$y = \sum_{k=1}^{d_N} \gamma_k \kappa(\alpha_k (x + \beta_k), \tag{2.36}$$

where d_N is the number of nodes in the single layer network, α_k the weight of the node, β_k the bias of the node and κ an activation function. A one layered network is depicted in Figure 2.4. The activation function can come in several shapes, and is what introduces the non-linearity in the network [7]. (2.36) can approximate any reasonable function with arbitrary accuracy if there is a large enough amount of nodes, d_N [19][18]. In this application the activation function used is the Rectified Linear Unit (ReLU) because it is widely used and considered successful in many neural network applications while it is also computationally efficient [23]. The ReLU is a max-value operation and is formulated as

$$y = \max(0, x) = x_+. \tag{2.37}$$

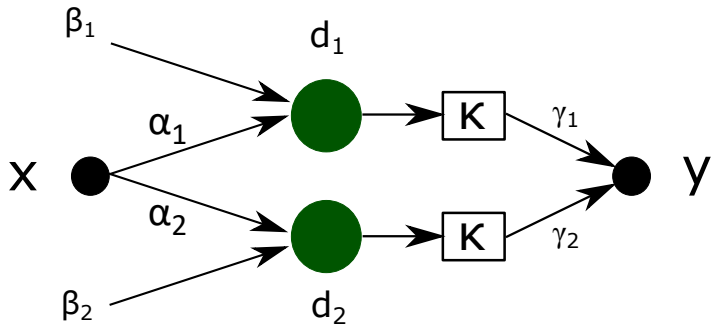


Figure 2.4: *A neural network in the form* (2.36) *with two nodes and an one dimensional input x.*

Feedforward neural network

Feedforward neural networks are one of the simpler forms of neural nets. The output from a previous layer of nodes are directly used as input to the next layer of nodes, and so on. The input signals moves through the hidden layers to the output layer without any feedback loops and finally form an estimate as depicted in Figure 2.5. Mathematically a ReLU feedforward network can be formulated as in (2.38) and (2.39), where F_1^N is a multi-dimensional non-linear function.

$$F_1^N(x) = (A_1^N x + b_1^N)_+, \tag{2.38}$$

where A_1^N is a matrix with a dimension of $(d_1 \times d_0)$ holding all weights α_k for each node and each of the input to each node in layer 1. b_1 is formed from α_k and β_k with a dimension of $(d_1 \times 1)$. If F_1^N is used as input to F_2^N, which in turn is used as input to F_3^N and so on up to layer L, F_L^N, one holds

$$y = C^N F_L^N(F_{L-1}^N(...F_2^N(F_1^N(x)))), \tag{2.39}$$

where C^N represents the output layer and holds all γ_k [7][19].

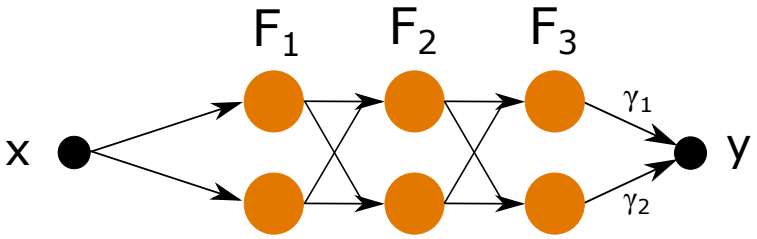


Figure 2.5: *A feedforward network of 3 layers of two nodes and a one dimensional input x. Each orange node holds the bias, activation function and weight illustrated in Figure 2.4.*

Cascadeforward neural network

All of the details of the feedforward network applies on the cascadeforward neural network, except that the output of each layer, including the input x is used an input to all following layers as depicted in Figure 2.6. This enables a preservation of purely linear relations between the input x and specific layers and the output y.

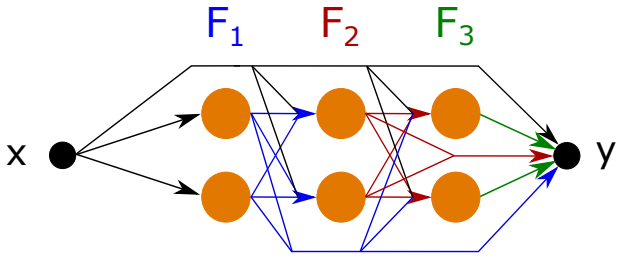


Figure 2.6: *A cascade forward network of 3 layers a one dimensional input x. The output from each layer is color coded, and illustrates how all outputs are cascaded forward.*

Parameter estimation

The neural networks parameters can be estimated using Stochastic Gradient Descent (SGD). The simplicity of implementation, computational efficiency, the wide use of it, and that it works well on big data sets makes SGD a good option for this work [15]. SGD performs parameter updates once per input sequence. The input sequence is randomly selected from the data set used for training to remove a bias to the samples appearing later in the set. The SGD updates the weight as

$$\theta = \theta - \eta \nabla_\theta V(\theta, x^{(i)}, y^{(i)}), \tag{2.40}$$

where θ is the model parameters, η the learning rate, ∇_θ the gradient of the cost function $V(...)$ with respect to the parameters, and lastly $x^{(i)}$ and $y^{(i)}$ the input and output sample i, respectively [26]. The cost function used is mean squared error (MSE) formulated in (2.41), and the gradients are calculated using backpropagation, which is very computationally efficient [16].

$$MSE = \frac{1}{N_s} \sum_{i=1}^{N_s} (\hat{y}_i - y_i^*)^2. \tag{2.41}$$

Regularization

Regularization, as described earlier, brings several advantages. An ad-hoc way of applying regularization when estimating neural network parameters is early

stopping. It is a simple procedure that relies on evaluating the cost function on a validation set during training, and cancelling the training if the cost function is not improved enough [26].

2.3.3 Ensemble learning

Ensemble learning is a method where several learners are combined to make a decision. The advantage of ensemble learners is that they can cover the weaknesses of specific learners. The ensemble learners discussed below are based on decision trees, where one tree represents one learner and the ensemble becomes the act of using several decision trees [27]. This work focuses on two methods, Random forest and Boosting, which have been used in previous works with good results [20][25].

Decision trees

A decision tree is a way to build a classification or regression model in the form of a tree structure. In this work a decision tree refers to the regression model version. A decision tree contains nodes and branches. A node contains a number of samples used for making a decision and a branch connects a node with its child node. Each node splits the data set into smaller subsets with at least two branches connected to it. The final nodes with no branches are called leaf nodes and these nodes represent the decision of the tree as illustrated in Figure 2.7. If the leaf node contains more than 1 sample the decision becomes the mean of all samples in the leaf node.

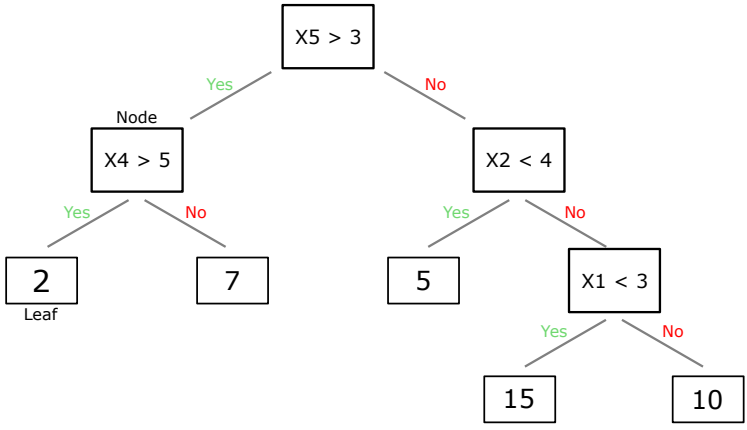


Figure 2.7: A simple decision tree with nodes connected using branches. Decisions are made in falling order from the top node, and results in a prediction in the leaf nodes.

When creating a decision tree, the splitting process used is the standard CART [2]. Each sample of the used inputs has a corresponding measured output. All input

signals, x_p ($p = 1, ..., n_u$), have their entire sequence of samples sifted through an inequality, a split. All possible splits are tested and for each split, the predicted output becomes the mean of the corresponding measured output of the input samples that passed the inequality. The goodness of the split is then determined by (2.42), which is the sum of the squared error (SSE) of the true output for each sample that passed the split and the derived mean. $1, ..., N_{p,L}$ ($N_{p,R}$) are the samples that passed (failed) the split. That is, the SSE is derived for both sides of the split, as they are equally important. The split with the lowest SSE is then selected, and the procedure is repeated.

$$e_t = \sum_{i=1}^{N_{p,L}} (y_i^* - \bar{y}_{t,L})^2 + \sum_{i=1}^{N_{p,R}} (y_i^* - \bar{y}_{t,R})^2. \tag{2.42}$$

The design of a decision tree is largely an iterative process. The amount of splits and number of samples in the leaf nodes and regular nodes are all parameters that needs to be tuned depending on the situation.

Boosting

Boosting is a method in which each tree is dependant on the tree before it. There are many variants of boosting algorithms, and one of them is the Least-Squared boosting method (LS-boost). First a decision tree is created from the input data. The predicted output is then compared to the actual output. The error then becomes input for the next tree and the sequence repeats. The prediction of the model then becomes the sum of all trees. The design of the boosting algorithm makes it so given enough iterations, it will create a very accurate model but it will also be prone to overfitting if dealing with high variance data.

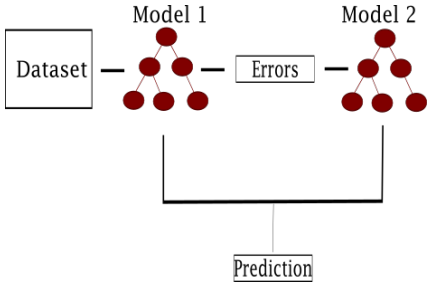


Figure 2.8: *The general approach of a boosting algorithm. Each following tree tries to fix the biggest errors from the previous model*

Random forest

Random forest is categorized as a non-parametric learning model. This means that it does not rely on any optimization methods to estimate parameters. Instead it could be considered an ensemble of several regression trees which are

trained through random signal selection [31]. A typical random forest flowchart is illustrated in Figure 2.9.

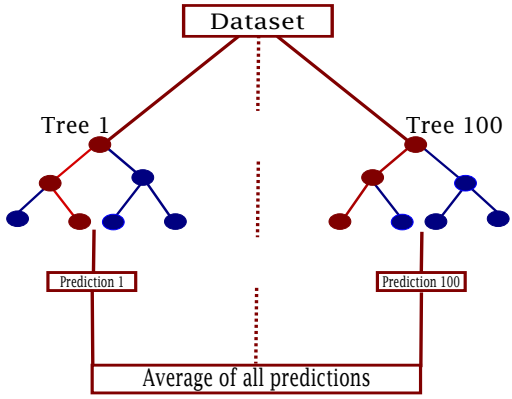


Figure 2.9: *A figure of the general approach of a random forest. Each tree is supplied with a random subset of the original dataset*

The random forest method trains a set amount of weak learners individually where each tree is trained as previously mentioned. The difference is how the samples are chosen. For each node a random subset of predictors is chosen and the best split is identified from that subset. The model output of the random forest then becomes the mean of the output of all regression trees in the random forest.

2.3.4 Model Validation

Leave-One-Out Cross Validation

Cross Validation (CV) is a procedure to validate a model. It involves training the model on a particular data set, and then validating the model on another previously unseen data set. There are several different ways to select the training and validation data sets. The intention is to determine how well the model performs on similar, but previously unseen data. One variant of CV is the Leave-One-Out CV, (LOOCV). LOOCV divides the available data set into N_{LOOCV} sections. One section is then used as validation data and the rest as training data and then a measure of performance is derived. When the measure of performance has been derived, another section is selected as validation data and the measure of performance is derived again. This is repeated until all sections has been used as validation data. The measure of performance of all validation sections are then compiled to an overall performance [13].

3
Methodology

In this chapter the methodology is described. At first, the steps taken to increase the understanding of the system are described. Then an explanation of the preprocessing is provided along with how the signals are sifted through to find the most adequate input signals for the models. Lastly, the modelling procedure and strategies are described. After the description of each strategy, the result of the strategy is concisely summarized, since intermediate results are used in the following tests. A thorough description and discussion of the results are found in the chapters 4 and 5.

3.1 Categorization of strokes

In order to ease the discussion and because the angular setting most likely introduces non-linearities in a possible relationship between HS-signals and RCS-signals, the strokes were categorized. The strokes were categorized into classes based on the angular setting of the boom during the strokes. The categorization was based on how wide or narrow the swing angle was and how high or low the tilt angle was. A stroke class could for example be LL1, meaning low-left-1, one depicting how wide it was. Table 3.1 shows the thresholds used and the corresponding joining when naming the class. None of the widest categories were used as training data during the modelling. This is because when a wide stroke is executed, several cylinders hits max length which cause faulty pressure signals. The lift angles categorized as M were from recordings during the excavation of a low tunnel, and were not used since there were so few of them and thus a comparably low sample size. The corresponding low strokes from the same low tunnel recordings were used up until a later stage, where the impact of the inclusion of these was determined. Only the recordings with a linear increase in stroke length

was used in order to further lower the complexity of the task. The definition used for a stroke recording, was that a recording started when the stroke length had reached a certain threshold and ended when the stroke length again passed the threshold.

Table 3.1: *Table of angular thresholds used when categorizing the strokes.*

Angular thresholds		
Angle	Threshold (degrees)	Category (XY)
Tilt	$x < -12$	X = U
Tilt	$-12 \leq x \leq -4$	X = M
Tilt	$-4 < x$	X = L
Swing	$y < -22$	Y = R2
Swing	$-22 \leq y < -6$	Y = R1
Swing	$-6 \leq y < 6$	Y = M
Swing	$6 \leq y < 22$	Y = L1
Swing	$22 \leq y$	Y = L2

3.2 Signal Contents

Frequency contents

To gather understanding of the contents available in the signals, it was necessary to view the frequency contents of the available signals. Especially in the strain signals, since that was the modelling target. It was done by applying the DFT using the Fast Fourier Transform (FFT) and plotting the spectrum. Since the spectrum is constant for white noise, the spectrum also revealed if contents at specific frequency ranges could be taken for white noise [8].

Relevant frequency ranges

Because of the large gap in sampling frequency between the RCS-signals and HS-signals and the strength in the relationship between accumulated damage and frequency, an important step was to determine on what frequencies the sought information exists. Hence, a good measure of what frequency range was relevant was to analyze the average drop-off of the accumulated damage when calculated from a strain signal passed through low-pass filters with varying cut-off frequencies. The low-pass filter used was the Butterworth-filter because it has the sharpest roll-off possible without inducing peaking in its pass-band in the Bode plot [5]. When the frequency range was determined, the HS-signals and RCS-signals could be downsampled and upsampled, respectively, to a frequency covering the relevant range.

The next step was to inspect how the different frequency components contribute to the signal in the time-domain, since that is where the DMG is determined. By

low-pass filtering the strain signals, one may find how the high frequency components contributes to the time signal. Thus a comparison between a low-pass filtered strain signal and a non-altered version of itself was made to find this relation. To do so the Butterworth filter was again used with a cut-off frequency varying between the Nyquist frequency of the RCS-set and the top of the frequency range, and then plotting the results.

3.3 Pre-processing of data

Before the signal relationships were determined, and before the signals were used to train models, they were passed through the following pre-processing steps.

Outlier detection and dead signal removal

When dealing with measurement data, outliers can appear. To detect outliers, the data was detrended and then each sample was compared to 4 standard deviations of the mean. The result of the removal of outliers turned out to be that they appeared so rarely or was hidden by other data points, that it was too hard to determine their validity. Thus no action was taken to remove outliers. Removing dead signals was important since it could completely disrupt any attempts of interpreting and modelling the data. If a signal is dead it will typically only display white noise as pictured in Figure 3.1 meaning that it is possible to find dead signals by checking for this phenomena. To find any potential dead signals in the data set each signal was checked to see if its behaviour was that of white noise by investigating trend and change in the signal.

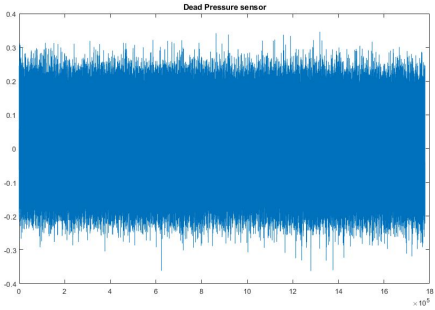


Figure 3.1: *Example of dead signal found in the data set*

Downsampling and Upsampling

Since the RCS- and HS-sets were sampled at different frequencies they had to be upsampled and downsampled, respectively. The signals were up- and downsampled to a frequency capturing most of the accumulated damage, as found

in Section 3.2. The upsampling was done by first zero-stuffing the data, meaning that new samples with a value of zero was inserted between already existing samples. The number of zeros inserted between the existing samples directly corresponds to how much the signal was upsampled. After the zeros had been inserted, the signal was low pass-filtered with a cut-off frequency of the original signals Nyquist frequency. The downsampling was done by first low pass-filtering the signal with a cut-off frequency that was half of the target frequency and then uniformly removing an amount of samples corresponding to how much the signal is downsampled [30].

The low pass-filter used in the two applications was a linear-phase finite impulse response filter with a poly-phase structure, resulting in a constant time-shift and fast computations [8][29].

Signal synchronization

As mentioned in Section 1.1.4 there was a time difference in the RCS- and HS-sets. The sets were synchronized by deriving the cross correlation of a signal present in both the RCS- and HS-set. Before the cross correlation was derived, the shared signals were high-pass filtered to better crystallize the fast dynamics. The signals were then shifted with the time lag corresponding to the highest cross correlation between the signals. This procedure was repeated over segments of N_d data points. N_d was determined to be 500 by evaluating how well the two signals fit as a function of N_d using MSE.

Off-set removal

Because the data set originally was not collected with the purpose that was pursued in this report, different signals were zeroed at different times with an inconsistency between the data-collection sessions. To remove these differences and get a more consistent data set, signals used during modelling were zeroed in the beginning of the stroke, meaning that only inter-stroke dynamics are captured. This was an iterative process to some extent, as a later investigation of relationships between larger trends and signals are examined in Section 3.5.1.

Normalization

When estimating the parameters in data-driven models it is favorable to have signals of roughly equal size. The input signals were thus normalized by finding the maximal value of the absolute value of all signals during a specific stroke and then dividing the signal with this value as loosely described in (3.1).

$$\begin{aligned} Q_{norm} &= max(|U|), \\ U &= \frac{U}{Q_{norm}}, \end{aligned} \tag{3.1}$$

where U represents all available input signals, and Q_{norm} the collection of normalization constants used. This resulted in the input signals roughly spanning

between -1 and 1. Since the EPS-signals in the HS-set approximately holds the same size, they were all divided by 100 to roughly match the size of the input. The F_lift signal had its constants derived in the same way as the inputs.

3.4 Signal Selection and Evaluation

When the signals available had been pre-processed, the search for relationships between input signals and output signals was initiated.

Correlation

To catch linear relationships between excavation settings that are fixed during strokes and the damage accumulation, a correlation test was performed. The value of each setting was plotted and fitted against the accumulated damage during the stroke, which indicates relationships that cannot be catched using coherence on inter-stroke changes. The correlation test was done on the average value of a handful of RCS-signals from each stroke recorded. This was then compared with the respective accumulated damage of that stroke. The RCS-signals used in the correlation test was the angular signals, since these were associated with the fixed settings of each stroke. To filter for meaningful correlations the significance test described in Section 2.2.2 was utilized and the correlations that failed the test was discarded.

Coherence Analysis and Mutual Information

Since the measure of coherence depicts how strong a relationship between two signals is, as well as on what frequencies it exists, it was used to sift through the available input signals. Since the RCS-set only contains information up to the Nyquist frequency of 6.25 Hz, the measure of coherence was estimated as the mean value of $C(\omega)$ with $\omega \in [0, 6.25]$. The coherence was determined between each EPS-signal in Table 1.1 and each RCS-signal in Table 1.2 on two different levels. The first level was more general and did not respect the stroke categorization, instead the average coherence of all RCS-signals and HS-signals were taken on all available strokes. The second level was more narrow, and determined the coherence of all RCS-signals and HS-signals in each stroke category respectively. The average coherence was then ranked according to its value and plotted, meaning that the first level gave a number of plots corresponding to the amount of HS-signals used, and the second level gave a number of plots corresponding to the amount of HS-signals used times the amount of strokes categories available.

The mutual information was applied in the same way as the coherence analysis, but on detrended data because the models that utilized the MI measure to select inputs was created for detrended signals. The trend appeared on very low frequencies, and its contribution to the average was thus very low and insignificant, removing the need to perform the coherence analysis for both the data with trends and detrended.

3.5 Modelling

There are a lot of ways to create and evaluate models using different software. The linear models were trained and implemented in Matlab with the help of the System identification toolbox, because of previous experience of it [17]. The ensemble learning part of the non-linear models was implemented using the Statistics and Machine Learning Toolbox in Matlab [21]. The neural networks were implemented in Python using the Pytorch toolbox because of the simplicity in its implementation and extensive documentation [24].

The modelling methodology covers a wide range of strategies, most involving several steps to narrow down the options in each respective strategy and thus, as previously mentioned, some intermediate results are briefly presented along the way. There were 3 main strategies which can be summarized in the order they were executed as following:

1. **Linear model for all dynamics**

   In this approach a search for the best possible linear model, catching both fast and slow dynamics, between selected input signals and output signals was conducted.

2. **Linear model for fast dynamics**

   This approach was similar to the first strategy, but only tried to capture the faster dynamics. That is, the signals were detrended before the modelling phase.

3. **Non-linear model for fast dynamics**

   This approach utilized non-linear models in an attempt to improve the models of sensors that proved to be particularly tough to model.

All linear models created were evaluated on a validation file using MSE, DMG and residual analysis. The MSE measure was used as a metric of how much to trust the estimated DMG. That is, if the DMG estimate was very close to the true DMG, the MSE measure helped determine if it was because of disproportional noise to signal ratio and luck, rather than a good model. When simulating the models, because the noise was added, the output was realized over 500 iterations. The noise variance used when creating the noise signal was estimated as described in Section 2.3.1. The MSE and DMG was derived and saved each iteration. It was then assumed that the MSE and DMG had a gaussian distribution. During evaluation they were plotted as their respective mean and 99%-confidence interval using error bars. During later stages of the modelling, when evaluating the robustness of the models, the LOOCV-procedure was used to iterate the model structures over each the available stroke recording in each stroke class to thoroughly evaluated their respective performance.

3.5.1 Linear model for all dynamics

Before walking through the methodology to find the best direct linear model, it is informative to point out that the methodology is intended to narrow down the options as much as possible. That is, the first steps have a very general approach, and only uses the ARX-model because of its property of being able to approximate linear system arbitrarily well given a sufficiently large model order [3]. The ARX has also proven successful in similar applications [12]. The ARX models are regularized using the TC-kernel. All results are generalized to cover all sensors. The workflow is summarized in the flowchart presented in Figure 3.2, where each box represents a test.

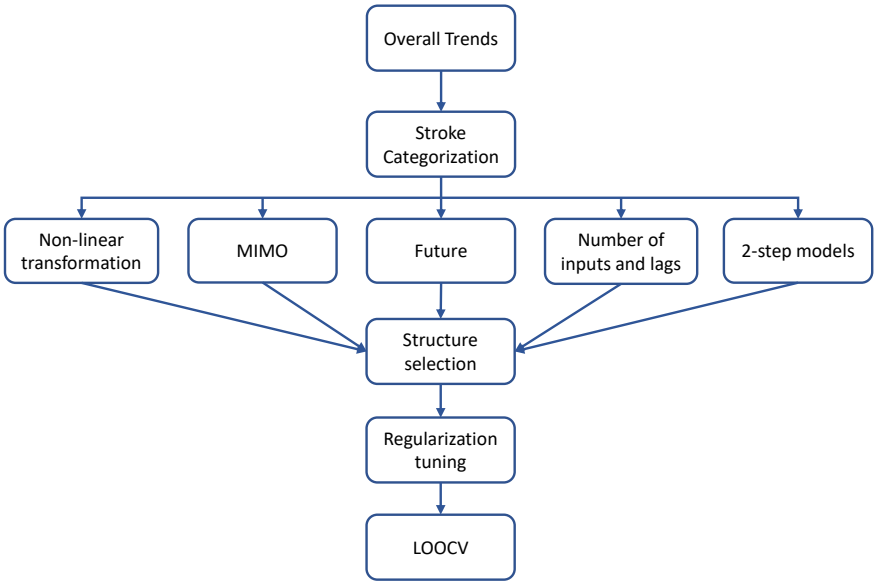


Figure 3.2: *Flowchart presenting the workflow with the intention to narrow down options.*

Investigating effect of overall trends

Section 3.4 describes the procedure to find general trends in the damage accumulation with respect to the signals that are somewhat fixed or has a static off-set during strokes. A natural thing to do when overall trends were found, was to determine whether the static levels improve the models if they were included, that is the specific signals with larger DMG trends were not zeroed. Thus, two multi-input single-output (MISO) ARX-models with 10 input signals and 1 output signal were created. One of the models zeroed the signals with, according to the test, meaningful static levels, while the other one did not. The output signal used was EPS_2_28 and the input signals were selected using the first level of the coherence ranking. The order of the models was determined to 40 by deriving the

cross correlation between the output and input signals, and then determining the amount of lags up to the first time the cross correlation crossed zero. The result is shown in Section 5.1.1, and yielded that including static levels did not improve the models. The result was then generalized to cover all strain sensors and all following linear methods.

Effect of stroke categorization when modelling

As mentioned in Section 3.1, it was suspected that non-linearities in the signal relationship was introduced when the cutter wheel's angular setting was changed between strokes. Training a model for each stroke category would help avoid possible non-linearities, and thus improve the results of the linear models.

MISO ARX-models with 8 input signals were created for two output signals, EPS_2_9 and EPS_2_7. The output signals were chosen in order to capture the possible differences in behaviour between the main-frame and the thrust-frame, as it was suspected that the guide rail connection between the thrust-frame and main-frame introduced non-linearities. Again, the input signals were chosen according to the coherence ranking, and the lags according to the correlation test. When selecting inputs, the fact that the categorization of strokes yields a more accurate coherence ranking, when compared to taking the average of all categories, was utilized. That is, the second level coherence ranking was used. One model was created on all data available, and then evaluated on a specific stroke. The data was then split into different stroke categories according to the categorization of strokes in Section 3.1, and a model was trained for each different stroke. The specialized models were then compared against the model created on all strokes. The result is presented in Section 5.1.2, and yielded that the categorized models were far superior, and because of that they were used for further investigations, along with the second level coherence ranking.

MIMO models for better noise isolation

The results of Section 3.2 indicated a need for a good noise model, because of the limited frequency range of the RCS-set and that the higher frequencies contribute to the accumulated damage. A possible way to improve the noise model is to use a MIMO-model instead of a MISO model. Since several outputs are estimated at the same time, the model is granted more information in terms of more reference output signals during training, which may lead to an improved noise model. While examining this possibility, the placement of the sensors was taken into consideration, again with the mechanical connection in mind. The approach was thus evaluated on the strain signals EPS_2_28 and EPS_2_3, which are placed on the sub-frame and main-frame respectively. One reference model was created for each of the evaluation signals. Then three models were trained with three different combinations of output signals. The first combined two sensors placed on the sub-frame, EPS_2_28 and EPS_2_9, the second combined one sensor placed on the thrust-frame and one on the main-frame, EPS_2_28 and EPS_2_3, and the third combined two sensors placed on the main-frame, EPS_2_3 and EPS_2_1.

The model stucture used for evaluation was ARX with 40 input lags and 10 output lags. 10 input signals were used in all models, and they were chosen using the coherence ranking. In the cases where the coherence ranking differed a lot, a trade-off with regards to the value of the coherence was made. The result, shown in Section 5.1.3, was not convincing enough to further evaluate the possibilities of MIMO models.

Evaluating non linear signal transforms as input signals

Because of the guide rail connection between the sub- and mainframe, the placement of the strain sensors has proved to be important for the success of the linear models. This phenomena was apparent when reviewing the results of the MIMO models. Because of this, a test with the simplest and most obvious non-linear transformation was conducted. Since the guide rail position is a function of the stroke length and the loads are changed with the change of length of the moment arm, the non-linear transformation tried was a multiplication of each of the RCS-signals with the highest coherence ranking and the measured stroke length, DIS_5_1. The model structure used was MISO ARX with 40 input lags and 10 output lags. The 10 inputs signals with the highest coherence ranking was chosen in the reference models, and in the models evaluated, the same 10 inputs signals was used in addition to these 10 inputs multiplied with the stroke length. The outputs modelled was EPS_2_28 and EPS_2_1, that is one sensor on the main frame and one sensor on the thrust frame. The result, presented in Section 5.1.4, showed no apparent improvement on the EPS_2_28 sensor, but a large improvement on the estimate's ability to catch the trend on the EPS_2_1 sensor.

Two-step linear model

As previously mentioned, a handful of pressure sensors existed in both the HS- and RCS-set. Since the RCS-set does not contain the higher frequencies, the higher frequencies were described using a noise model. In this approach, a test to see whether the noise model could be improved by using a two-step linear model was conducted. The two-step linear model was constructed by first creating a single input single output (SISO) ARX model between specific pressure sensors in the RCS-set and the HS-set. The output from these was then used as input to a MISO model of a strain sensor. The hypothesis was tested on the strain sensor EPS_2_9, since it had proven to be one of the easier to model. The MISO model used 9 inputs signals selected using the coherence ranking. The 4 with the highest coherence ranking was feed through SISO models before they were used an input to the MISO model. Both the SISO and MISO models used 40/10 input/output lags. The resulting performance was compared to a model that did not feed the top 4 input signals through SISO models. The results from the comparison is presented in Section 5.1.5, and it showed that no significant improvement was noticeable for the test.

Determining the effects of number of input signals and amount of lags

In theory, since the optimization problem solved in linear models seen in Section 2.3.1 is convex, including all RCS-signals as inputs should yield the best model. Inputs that don't contribute to the output will only increase computation time to a varying degree and reduce the reliability of the model, since it will be dependant on more sensors. The same goes for the amount of lags of the inputs signals used, if they do not better the model an amount proportional to the increase in computation time, they are deemed unnecessary.

By using the coherence ranking to choose input signals, a test was performed in which models built with 4, 10, 20 inputs were compared for the same output signal, EPS_2_9. The models were trained on the same data set and evaluated against the same data set, using a fixed amount of lags.

Following this, a test to determine whether the model estimate can be improved by using more lags of the inputs and output than 40 and 10, respectively, was done. It was conducted as a complement to the procedure used to find appropriate amount of lags when determining the significance of overall trends, but also to point out possible improvements. It was conducted by training an ARX-model for EPS_2_9 including different amounts of lags. The amount of lags tested was 15/3, 40/3, 40/10, 40/20 and 80/10 (Input lags/Output lags). The performance increase was evaluated using residual analysis. That is, the auto-correlation of the residual and the cross-correlation between between the residual and the input signals were compared to the thresholds described in (2.34a) and (2.34b).

The test results, presented in Section 5.1.6, revealed that more input signals generally increased the performance as expected. The same held regarding the amount of lags. Thus, going forward, in each investigation a trade-off is made between the time consumption and model performance increase.

Future input signals

Up until this point, it was assumed that the strain signals preceded the RCS-signals with respect to time. In this approach the opposite was tested, if future RCS-signals contained information about the strain signals. That is, future measurements, or negative lags, of the input signals was used and compared to a case where only past measurements of the RCS-signals was used to see if the models could be improved. This was possible since the entire signals were available. The test was performed by training three models, the first used 40 past measurements (regular half), the second used 40 past measurements and 40 future measurements (future), and the third used 80 past measurements (regular). The estimated signal was EPS_2_9. The evaluation was made on two stroke classes, LL1 and UR1. The results showed no improvements when future measurements was used. It is presented in Section 5.1.7.

Table 3.2: *Table of the used polynomial orders when evaluating model structures performance for each sensor*

Model structure	Polynomial orders
ARX	na = 10, nb = 40
ARMAX	na = 10, nb = 40, nc = 10
OE	nf = 10, nb = 40
BJ	nf = 10, nb = 40, nc = 10, nd = 10

Determining the model structure for each of the strain sensors

Since the strain sensors are placed in various locations, the circumstances and conditions for modelling also vary. In order to determine which of the ARX, ARMAX, OE and BJ model structures that fits the best for each of the sensors, each of the model structures was trained for each of the strain signals on one recording for each stroke class. All models structures, except ARX, used a minimum improvement of 0.01% and a max iteration of 20 as stopping criteria. The parameters of the model structures, except ARX, was updated using the Gauss-Newton, Levenberg-Marquardt and the steepest descent methods in sequence, where the first method leading to a loss function reduction was used. The iteration over the stroke classes was made to provide robustness to the result, and ensure that the model choice is the best possible. The model structures all have a varying amount of possible parameters, and to evaluate each model under approximately the same conditions, the orders of the different polynomials was selected so that they to some extent reflected that of the ARX structure. All models were MISO and their polynomial orders are presented in Table 3.2. When reviewing the result, the MSE and DMG, but also the consistency of the DMG estimation was taken into account. That is, if the DMG estimation contained a bias but had low variance. The thrust-frame sensors used the 10 highest ranked signals. The main-frame sensors also used the 10 highest ranked signals, with an addition of two extra inputs. Based on the results of Section 3.5.1, when evaluating the sensors on the main-frame, the two signals added were the non-linear transform of the highest ranked signals with the stroke distance, to aid the ability to follow the trend. The result showed the BJ structure was the best for all sensors except EPS_2_28 and EPS_2_9, for which the ARX structure was deemed the best. It is thoroughly presented in Section 5.1.8

Tuning the regularization of the chosen structures

Up until this point, the models had been arbitrarily regularized using the ridge regression with a $\lambda = 1$, except in the case of ARX models, which used the TC kernel. The regularization was a powerful tool to determine the bias to variance trade-off in the estimation error. Thus it was important to evaluate and tune the value of λ to increase the model performance. The tuning was made using the LL1 stroke class, the EPS_2_1 and the BJ model structure, as it was deemed the most fitting for all signals except EPS_2_9 and EPS_2_28. The performance was

evaluated for three λ values, 0.01, 1 and 100. Given the high MSE in the results from the previous section, more non-linear transformations were added to better help catch the trend. A total of 14 inputs was used, where 5 were transformations with the stroke distance. The result, as shown in Section 5.1.9, yielded that $\lambda = 1$ was a better choice with regards to catching the trend, and was then generalized to cover all sensors using the BJ structure.

LOOCV evaluation of selected structures

So far, the approaches yielded good results for the thrust-frame sensors but questionable results for the main-frame sensors.

The evaluation of the MSE for the different model structures, showed that among the main-frame sensors, EPS_2_1 had a comparably low MSE. Because of this the LOOCV evaluation was in first hand conducted on the thrust-frame sensors and EPS_2_1. EPS_2_1 was also a natural selection since the regularization was evaluated on it. Because of an accurate DMG estimate, and relatively low MSE, the LOOCV for EPS_2_1 was done on the UR1 stroke class. At this point, the amount of input signals was increased to further help the models. The 11 first inputs used was selected according to the coherence ranking, the last 5 used was the top 5 inputs non-linearly transformed with the stroke distance. The non-linear transform was used since Section 3.5.1 proved it to be necessary to capture the trend properly. Given the goal of recreating the trend while also providing a good DMG estimate, the $\lambda = 1$ was used. The LOOCV for EPS_2_9 was performed on the UL1 and LM stroke classes, as these showed the most promising and unpromising results with regards to the DMG estimate, respectively. The same argument was used when performing the LOOCV on EPS_2_28 and F_lift_tot, on which the stroke classes used was UL1 and LM, and LM and UR1, respectively. The top 15 inputs was used during the LOOCV of the thrust-frame sensors. The LOOCV on the selected gagues and stroke classes was performed twice. The first evaluation included the recordings of the excavation of a low tunnel, while the second did not. This was partly done to investigate why the categorization was successful. That is, was the categorization successful because of simplifying the problem in terms of only angular settings, or rather because of how the cutterwheel interacts differently with the mountain, depending on what kind of tunnel that was excavated. Of the tested stroke classes, the one containing recordings during a low tunnel excavation was LM. The results was promising on certain sensors, and motivated a similar investigation using detrended signals and an exclusion of the low tunnel recordings.

3.5.2 Linear model for fast dynamics

Given the results of the previous approach, where an attempt to model all the dynamics were made, the need of a simplification of the problem when using linear models was evident. This approach simplifies the problem by detrending the signals before the modelling phase. This approach inherited the intermediate results and conclusions drawn from the previous sections. That is, the models are

regularized and categorized MISO models with no future lags, which uses non-linear transformations with the stroke length when modelling the main-frame sensors. Further, the low tunnel recordings are not included in the training or validation data. The procedure is thus much faster and is summarized in Figure 3.3. All results from this section are thoroughly presented in Section 5.2.

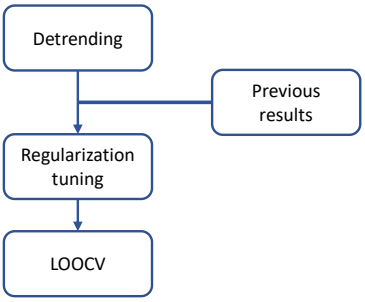


Figure 3.3: *Flowchart showing the faster procedure when finding adequate models for detrended signals.*

Detrending the signal

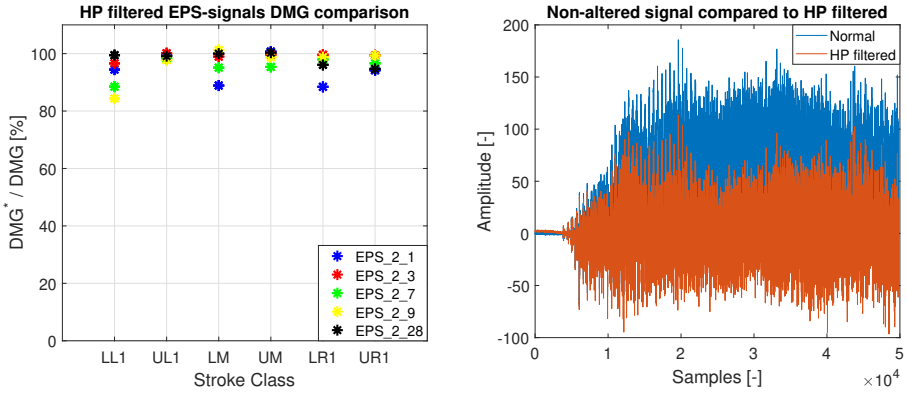
Before the detrending of the signals was implemented, two different methods were evaluated. The removal of the moving average (MVA) and high-pass filtering with a Butterworth filter. The evaluation metric was how well the DMG was preserved, whilst removing the slower dynamics. The MVA did not show potential up until a very large number of samples were included, which in the end basically resulted in removing the mean. As the ultimate purpose was to remove the trend, while keeping approximately the same DMG, MVA was deemed inferior to the high-pass filtering. The resulting signal managed to capture between 100% and 83 % of the true DMG when evaluated for all examined relevant signals over all stroke classes, as shown i Figure 3.4. The detrending was made on all signals except the stroke distance and was implemented after all the pre-processing steps.

Regularization of detrended models

After the data was detrended, the same regularization test as previously described was repeated on EPS_2_1 on stroke class LL1 to find the most fitting λ value. The inputs and number of lags used was the same as the previous regularization test. The results showed that $\lambda = 0.01$ was the most fitting.

LOOCV evaluation of the detrended models

When the λ value had been decided, a LOOCV evaluation of the detrended models was conducted. To better highlight possible improvements as a result of the



(a) DMG comparison *(b) Time plot*

Figure 3.4: *Plot of how well the DMG was preserved, expressed as a percentage of the true DMG*, and the resulting signal after high-pass filtering.*

detrending, the LOOCV evaluation was performed on the same sensors and the same stroke classes and used the same inputs as the previous LOOCV evaluation. The result showed that the detrended models generally performed better, and thus a compilation of the estimation results of all sensors for all stroke classes was performed for the detrended models.

Producing final results of the linear model approach

At this point, several model structures and approaches had been tested and excluded or included. Thus, as a last step a large LOOCV evaluation of all stroke classes on all sensors was performed. The LOOCV evaluation of the EPS_2_9 sensor was then compared to an equivalent LOOCV evaluation which used the top 4 input signals from the HS-set. This was done as a survey on how well the high frequency components are recreated using the noise realizations and models. The EPS_2_9 sensor was then used in a comparison where the output did not use a realization of noise but rather utilized a scaling of the results. Two types of scaling was used. The first scaled the output before deriving the DMG, and the second scaled the DMG after being derived from the output. The first scaling was done with the hypothesis that the contribution from the high frequency components may be proportional to the amplitude of the lower frequency components. The second was based on roughly the same hypothesis, with the difference that the proportional relationship existed after the DMG derivation.

3.5.3 Non-linear model for fast dynamics

As a complement to the linear models, which more or less struggled to accurately recreate the signals while also providing a good DMG estimate, two categories

of non-linear models were evaluated. The models were evaluated on detrended data as it showed an increase in performance and simplifies the problem. The non-linear models evaluated were ensemble learning, which has proven successful in remaining useful life (RUL) estimation, and feedforward and cascadeforward neural networks, which has been proved useful in identification of dynamic systems in addition with the system being open-loop [20][19][9][28]. Given the larger difficulties faced when modelling the main-frame sensors, the evaluation of the models was done on these particular sensors. The results of the different model structures are shown in Section 5.3.

Neural networks

Because there were two neural network structures that were regarded as good options, a test to determine the better structure was performed. The depth and width of the networks were highly important since a deeper network can introduce more non-linearities. Instead of evaluating the best possible network size with respect to performance in estimation versus the computational time of the training, a somewhat arbitrarily large network was used from the start. Both network structures, the feedforward and cascadeforward structures, thus used 2 hidden layers with an amount of nodes corresponding to the amount of inputs times the amount of lags of each input. The cascadeforward structure naturally grew slightly bigger than the feedforward structure because of its cascading property. Given that the networks may preserve linear relationships, and that the amount of inputs used when LOOCV evaluating the linear models proved somewhat successful, the same amount of inputs and input lags are used when training the networks. That is, 15 input signals with 40 lags each. The networks used the same stroke categorization to ease comparison and to simplify the task at hand. They were evaluated on the stroke class LR1 for the detrended EPS_2_1 sensor. The input signals were selected using the mutual information ranking, in the same manner as previous input signals selections with the coherence ranking. The top 4 ranked pressure signals were non-linearly transformed with the stroke distance. The loss function used to evaluate the performance was MSE, and the parameters were optimized using SGD. The training data batch size used was 5, which was selected randomly by shuffling the training data before the extraction of the batch. The learning rate was selected to ensure convergence while considering overfit. That is, the learning rate was selected so that the MSE was consistently improved. The results showed that the feedforward network structure performed better with regards to MSE and DMG, but because of the lack of higher frequency contents it was still unsuccessful. Two more tests were thus performed, the first used 4 hidden layers of 1800 nodes each with the feedforward structure. The second used the 2 hidden layers structure and gaussian distributed noise signal, with 0 mean and 0.1 variance, as input. Neither tests revealed any significant increase in performance.

Ensemble learners

The random forest model was trained on the same inputs as the neural networks. The models were also evaluated on the same stroke class, LR1, and EPS_2_1 sensor to get a clear comparison between the models. The random forest model was optimized using the built in optimizer in Matlab. The optimizer evaluates different models using Bayesian optimization. The parameters varied were the minimum leaf size, number of trees and maximum number of splits. For the sake of keeping the computational time and model size reasonable a cap of 700 was set for the maximum number of trees and maximum number of splits. The Boosting model was optimized using the same method and inputs as the random forest model with the exception of optimizing the learning rate instead of the minimum leaf size.

4

Signal Selection Results and Discussion

In this chapter results regarding the signal contents and signal selection methodology are presented and discussed.

4.1 Signal contents

Frequency contents

Figure 4.1 shows the result of taking the DFT of one the relevant HS-signal. All DFT's for the HS-signals exhibited approximately the same behaviour. The result shows that while the largest signal content is present in the frequency range covered by the RCS-set, there is still a non-negligible portion above the range. The rightmost plot also shows that the content above the RCS-range is not solely white noise.

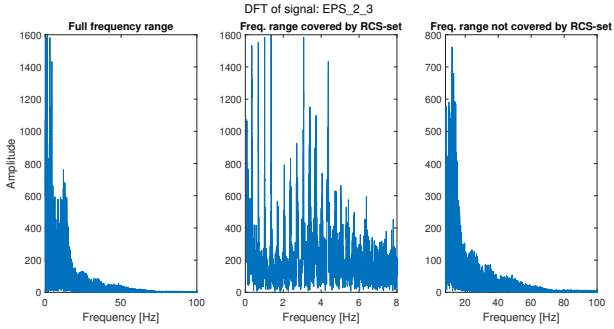


Figure 4.1: *The DFT of the HS-signal EPS_2_3, with three different zooms.*

Frequency ranges found relevant

The effects on the DMG when downsampling at different frequencies is shown in Figure 4.2. The results show a steep decline in captured damage when downsampling below 200 Hz. That is, frequencies above what is available in the RCS-set has a large contribution to the DMG. It is concluded that an appropriate frequency to down- and upsample to is 200 Hz. The result of the low-pass filtering

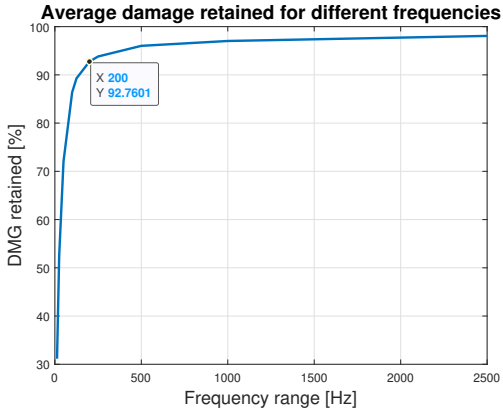


Figure 4.2: *A plot that shows how downsampling affects the accumulated damage of strain signals*

at different frequencies is shown in Figure 4.3. It shows how the colored noise adds to the lower frequencies and their peaks and valleys, and thus increases the accumulated damage. The results shown in Figure 4.1 - 4.3 reveals that the input signals themselves may not contain enough high frequency contents to accurately recreate the strain signals. The difference between a low-pass filter with 40 and 100 Hz cut-off frequency is comparably low, and confirms what is shown Figure 4.2 - that the decline of DMG is steeper the lower the frequency is.

4.2 Signal relationships

Overall correlations

An example of the overall correlations with DMG are shown in Figure 4.4. The plot show correlations that have passed the significance test. The results show that there is a large variance between recordings, which results in low correlation scores. The results suggest that the angular settings, the angular orientation of the complete carriage and the angular speed of the cutter wheel has an effect on the accumulated damage. Since the EPS_2_7 sensor is placed on one side of the main frame, it is reasonable that the swing angle setting has an effect on its output. The slope angle shows a very small trend, but it is also reasonable that a change of carriage orientation affects the strain signal since it changes the conditions of the gravitation. That the angular speed of the cutter wheel increases

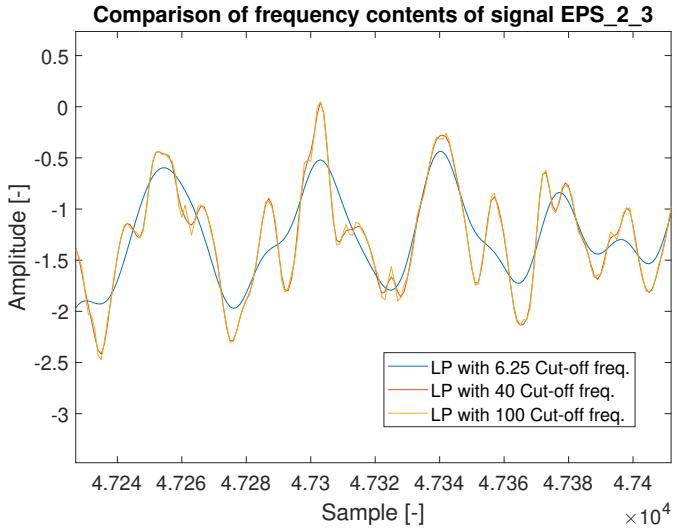


Figure 4.3: *A zoom-in on the HS-signal EPS_2_3 in the time domain, after being filtered with different cut-off frequencies.*

the accumulated damage is perhaps the most unsurprising, since it results in harder impacts with the rock. Given the results, it is preferable to evaluate how these overall correlations may affect the modelling. The results of the EPS_2_7 sensor is representative for the remainder of the sensors.

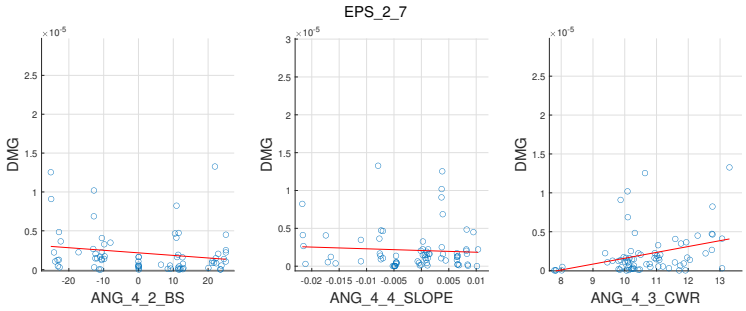


Figure 4.4: *The correlation for EPS_2_7 over all available data files. Only correlations passing the significance test is shown.*

Coherence analysis and mutual information

A selection of the results from the coherence analysis described in 3.4 is shown in Figure 4.5 and 4.6. Figure 4.5 shows the first level of the coherence ranking, while Figure 4.6 shows the second level. To note is that the ranking changes between the two levels. For the EPS-signals located on the thrust-frame there

exists a clear pattern, where the pressure sensors PRE_3_7 - 3_8 contain the most information about the EPS-signals. The full rankings for all combinations are not shown here but were presented in the same manner. The rankings generally show that the swing cylinder pressures are highly informative for all sensors, especially for those placed on the thrust-frame.

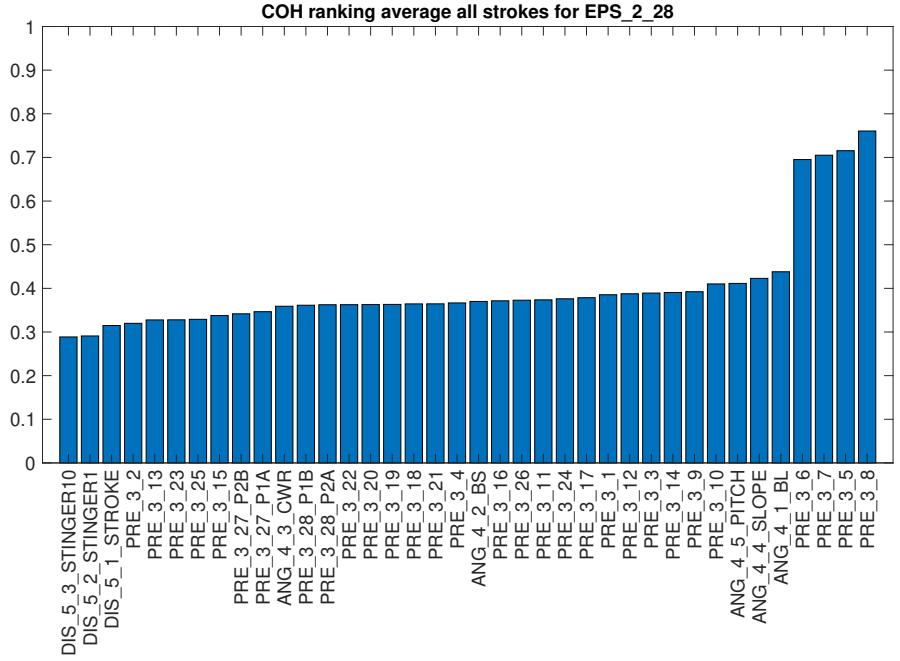


Figure 4.5: *The coherence ranking for EPS_2_28 over all strokes*

The mutual information rankings are presented in the same way as the coherence analysis and a selection is shown in Figures 4.7 - 4.8.

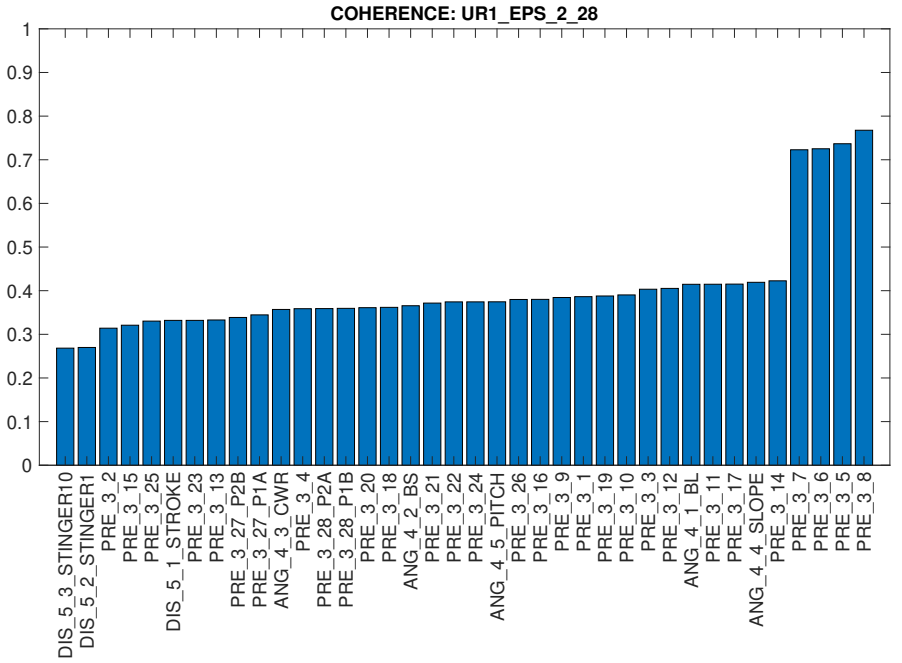


Figure 4.6: *The coherence ranking for EPS_2_28 on the stroke UR1*

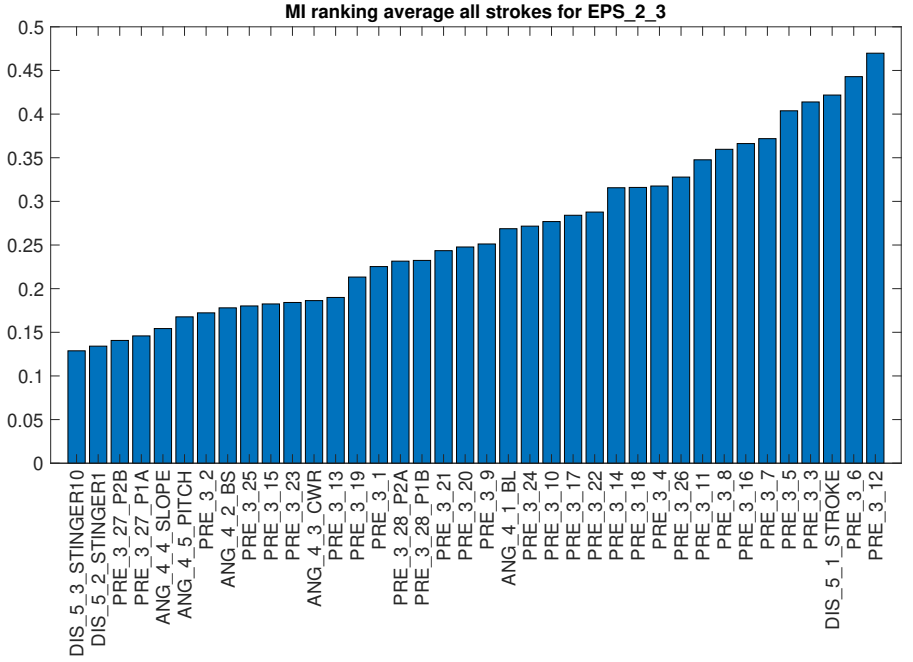


Figure 4.7: *The mutual information for EPS_2_28 over all strokes*

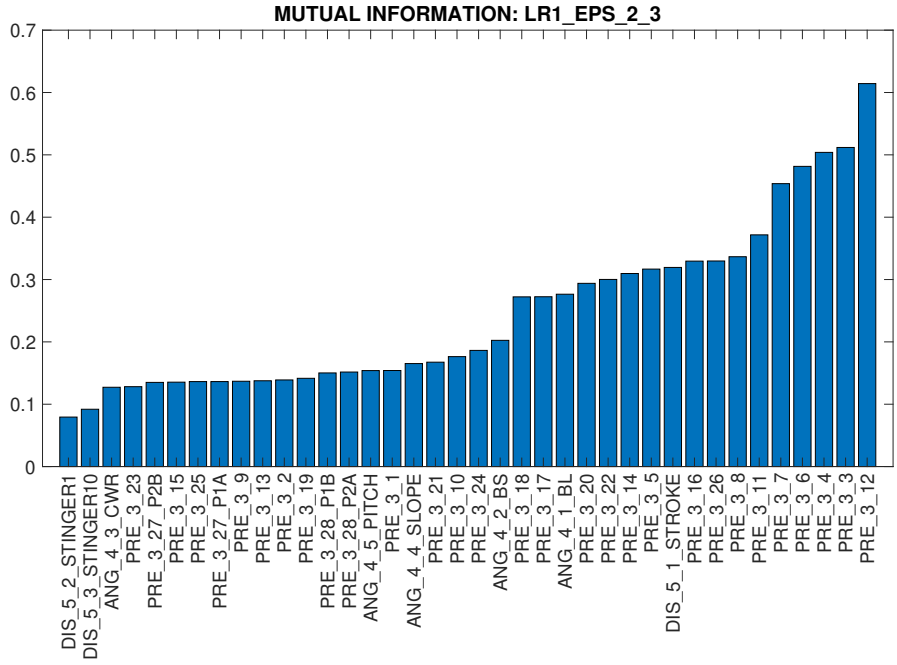


Figure 4.8: *The mutual information ranking for EPS_2_28 on the stroke UR1*

5
Model Results and Discussion

In this chapter the results of the various investigations are presented in chronological order. The results are discussed and intermediate conclusions are presented.

5.1 Direct linear models for all dynamics

5.1.1 Overall trend investigation

The test of the impact of overall trends and static levels in signals yielded that including the static levels caused an off-set in the amplitude, resulting in higher MSE. The outputs from the two models are shown in Figure 5.1 along with a reference signal. There was no difference in the residual analysis of the two models. The reason why the inclusion of the static levels of, for example, the cutter wheel rotational speed (CWR) does not improve the model is believed to be that the higher impact forces caused by the increase in CWR is already represented by an amplitude increase in the pressure sensors used as input signals. While the investigation was not made with the specific purpose of determining whether overall trends could be used to sift input signals, the results suggest that it is not preferable since what caused the overall trends did not impact the results in a positive way.

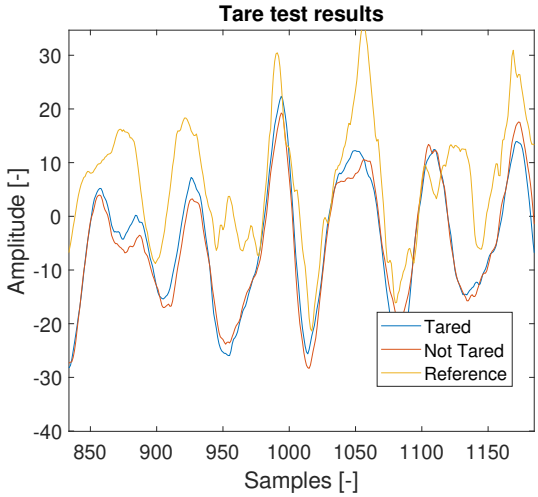
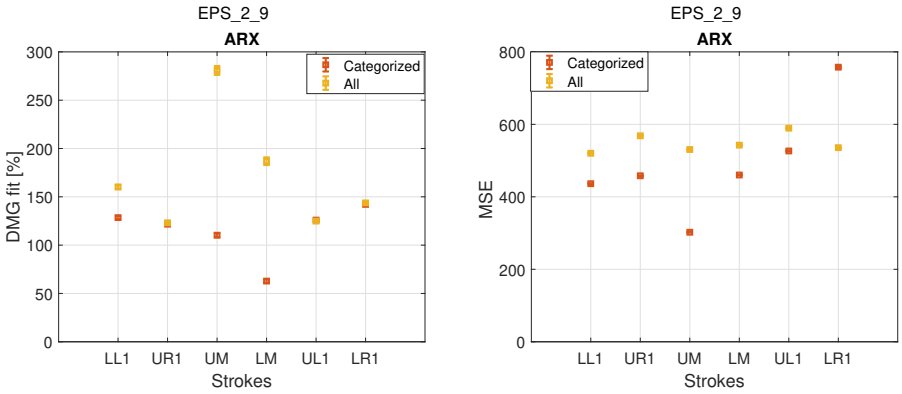


Figure 5.1: *A zoom-in on the tared, non-tared and reference signals, showing that the main difference is a shift in amplitude. The tared and non-tared signal is the average of 500 simulations.*

5.1.2 Result of stroke categorization during modelling

The results in Figure 5.2 shows an overall performance increase when using stroke categorization. Worth noting is the MSE of the LR1 stroke, which is higher for the categorized model. This deviation is explained by the variations in the excavation procedure between validation files. That is, since variation in the validation files exist, the overall result is what matters. Since the results show that categorization improves the model performance, it is preferred in further evaluations. The improvements from the categorization are believed to be because the individual modelling goal are simplified and narrowed down. It also indicates that there are relationships between the stroke classes are not measured and provided as an input or that a linear model cannot recreate it.

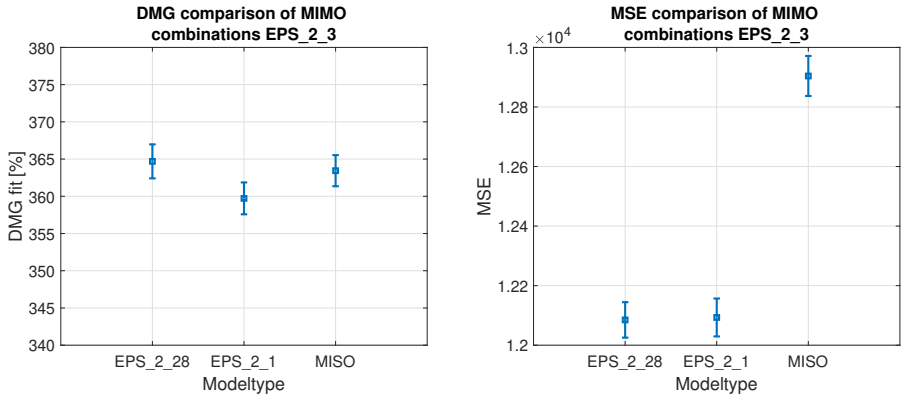


(a) DMG *(b) MSE*

Figure 5.2: *DMG and MSE comparisons between models trained on specific categorized strokes and a model trained on all available data.*

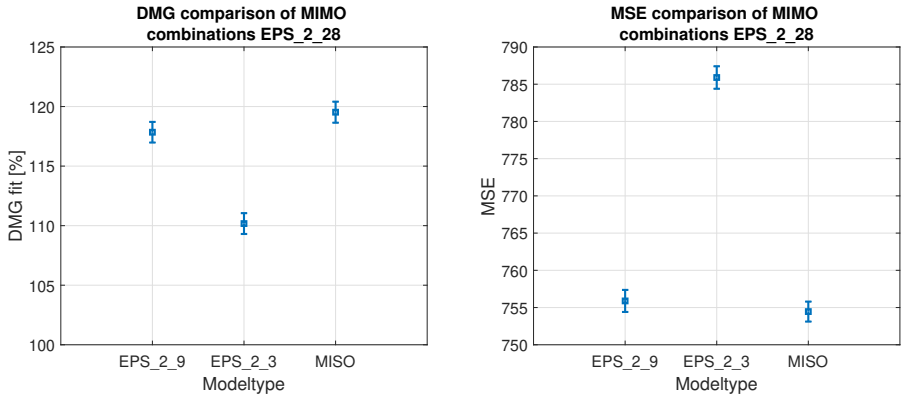
5.1.3 Result of MIMO and MISO comparison

The test on MIMO models ability to better capture the noise dynamics yielded questionable results. Figure 5.3 suggests that the DMG estimate could be improved by using MIMO-models, as well as the MSE. However, since the complexity of the model increased and the improvements are very small, in addition to the simulation results in Figure 5.5, the test results are not convincing enough. The same arguments can be applied on the results for EPS_2_28 MIMO-models, depicted in Figure 5.4 and 5.6.



(a) DMG *(b) MSE*

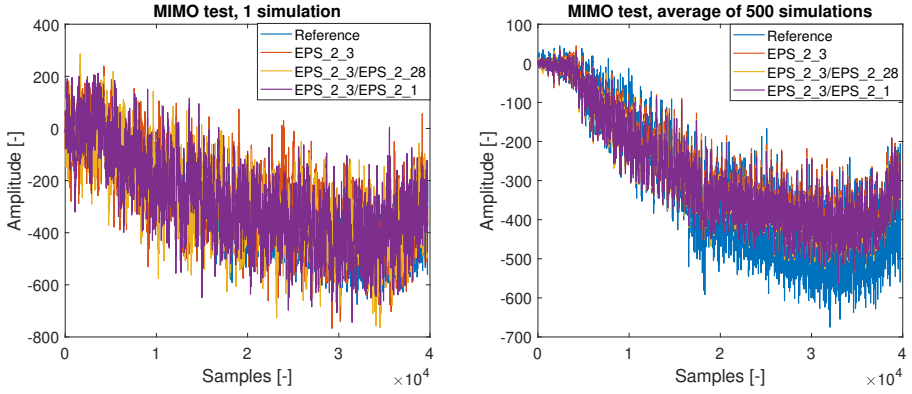
Figure 5.3: *Comparison of DMG estimate accuracy and MSE depending on MIMO combination.*



(a) DMG *(b) MSE*

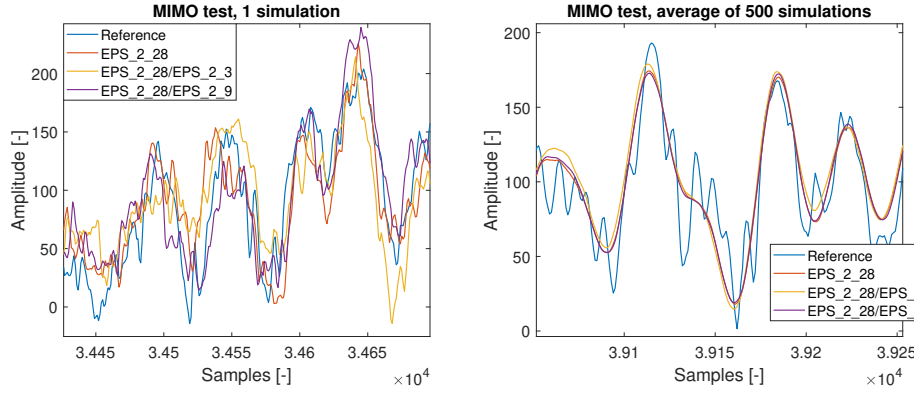
Figure 5.4: *Comparison of DMG estimate accuracy and MSE depending on MIMO combination.*

There is a large difference in the magnitude of both the DMG and MSE when comparing the results for EPS_2_3 and EPS_2_28. The reason for this is apparent when inspecting the time plots, as in Figure 5.5 and 5.6. Figure 5.5.b shows that the estimates does not follow the trend of the reference, which causes the model to increase the impact of the noise to minimize the loss function. This give less credibility to the results in Figure 5.3, but given the results for the EPS_2_28 the conclusions still holds. Figure 5.6 shows a zoomed-in view of the time plot of the EPS_2_28 estimate, which managed to catch the trend.



(a) 1 Simulation *(b) Average of 500 simulations*

Figure 5.5: *The time plot of the realization of the EPS_2_3 estimate, given different MIMO combinations.*

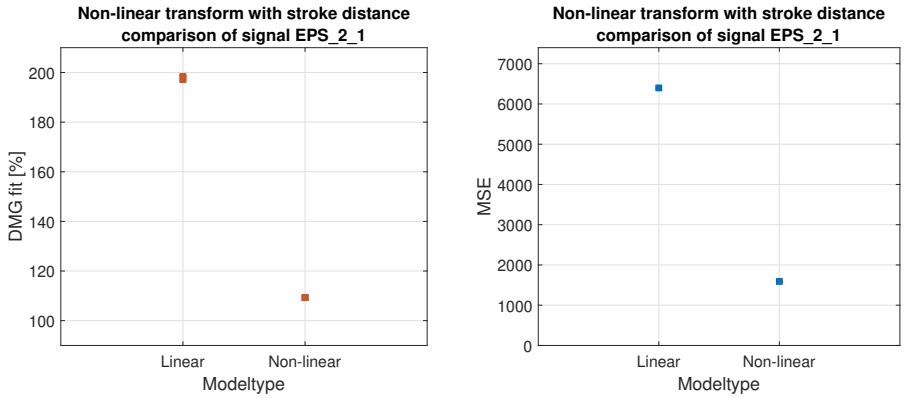


(a) 1 Simulation *(b) Average of 500 simulations*

Figure 5.6: *Zoomed-in time plot of the realization of the EPS_2_28 estimate, given different MIMO combinations.*

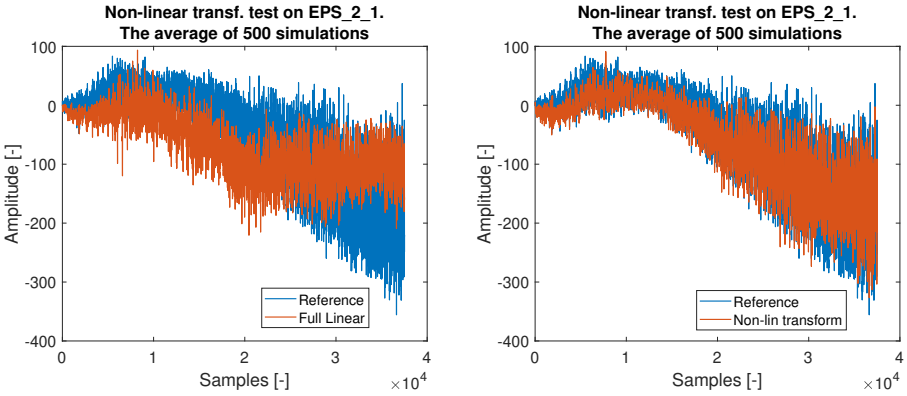
5.1.4 Results of non-linear signals transformations

When using non-linear transformations of the highest ranked signals as a complement to the non-alterered signals, the estimates was improved for the sensors placed on the main frame. The MSE and DMG of EPS_2_1 is shown in Figure 5.7. A comparison of how well the signals trend is followed by the estimate is shown in Figure 5.8.



(a) DMG *(b) MSE*

Figure 5.7: *Plot of the difference in DMG and MSE between a regular model and a model using the non-linear transform*

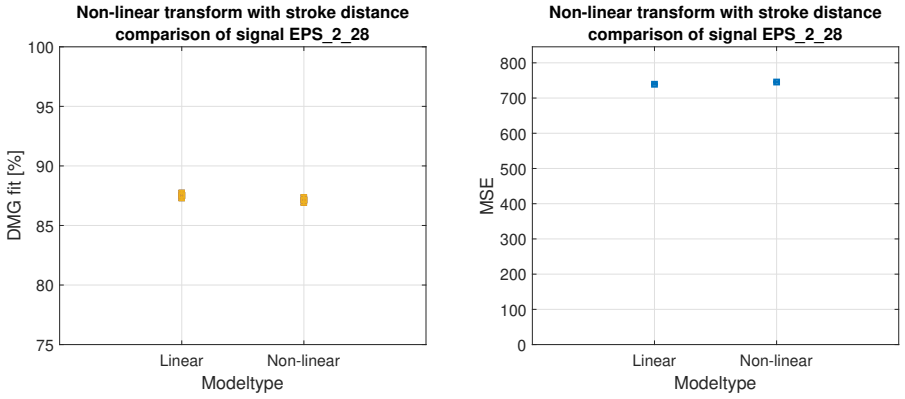


(a) Non-altered *(b) Non-linear*

Figure 5.8: *Plots of the resulting the estimated signal with and without non-linear transforms. The model using non-linear transforms manages to catch the trend better.*

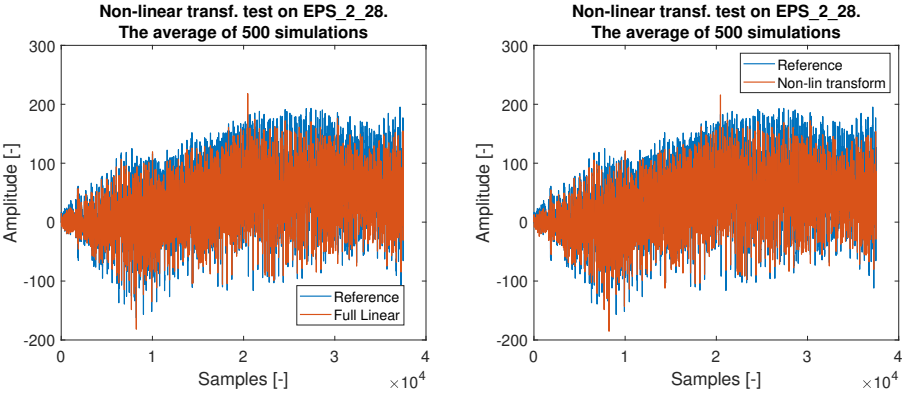
While the EPS_2_1 estimate was improved a lot, the EPS_2_28 estimate did not

show any significant improvement, as shown in Figure 5.9 and 5.10. The result suggests that a non-linear transformation with the stroke distance improves the estimate of the main-frame sensors, and that the non-linearity enters at the guide-rail connection. The results for the EPS_2_28 sensor is not so surprising. Since the most informative input signals for the EPS_2_28 sensor are pressures of the swing cylinders, which are connected to the mechanical structure on which the EPS_2_28 sensor is fastened.



(a) DMG *(b) MSE*

Figure 5.9: *Plot showing how the non-linear transform affected the EPS_2_28 sensor. It shows no significant improvement.*

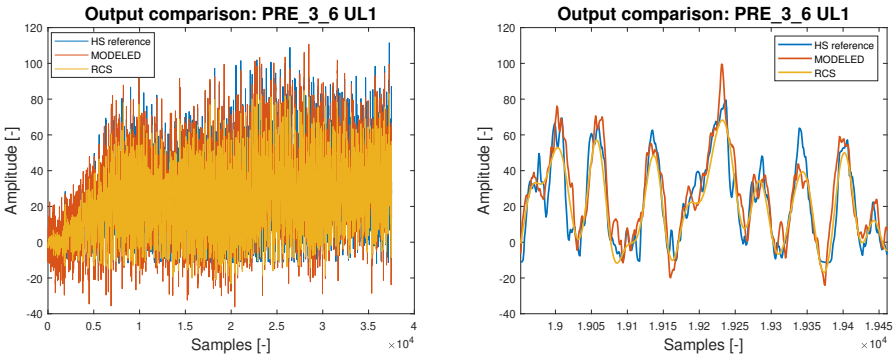


(a) Non-altered *(b) Non-linear*

Figure 5.10: *Plots of the resulting the estimated signal with and without non-linear transforms. There are no obvious differences in the estimated signal.*

5.1.5 2-step linear models

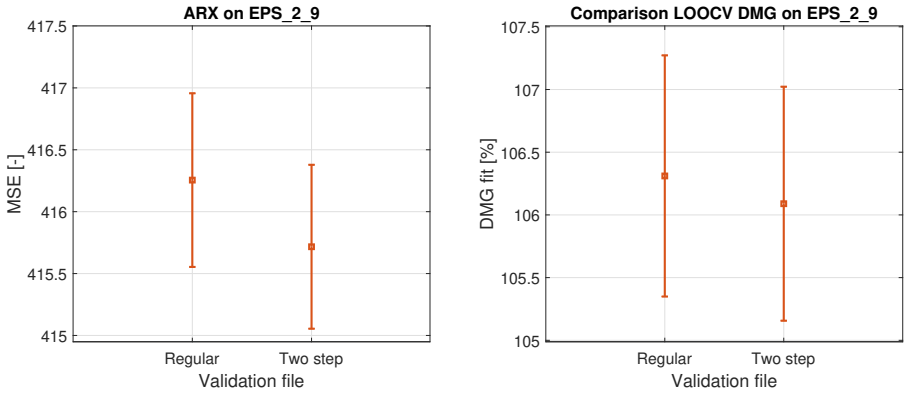
The plots in Figure 5.11 shows the SISO models estimate of the pressure sensor PRE_3_6 in the HS-set, using the RCS-set signal. The modeled signal captures the amplitudes more accurately but at the cost of overestimating the pressure variations in the start. The overestimation of the variations in the start is a recurring problem when simulating the models using noise. Given the relative size of the variations, the contribution to the total DMG estimate is marginal.



(a) Output comparison *(b) Zoom of the plot*

Figure 5.11: *Plots showing the modeled HS-set PRE_3_6 signal compared to a reference and the original RCS output.*

A comparison between the 2-step model and the regular model are shown in Figure 5.12. The two step model generates a slightly lower MSE but both models lie within each other's confidence interval. It reveals that while the SISO model produces a more peaky input signal, it does not improve the final results. It is thus concluded that the MISO model does the job equally well, and that 2-step models are not preferred. Since the test is done on the EPS_2_9 sensor, which along with EPS_2_28 provides the best modelling conditions given the available input signals, no improvements are expected if the procedure is repeated on main-frame sensors or F_lift_tot.

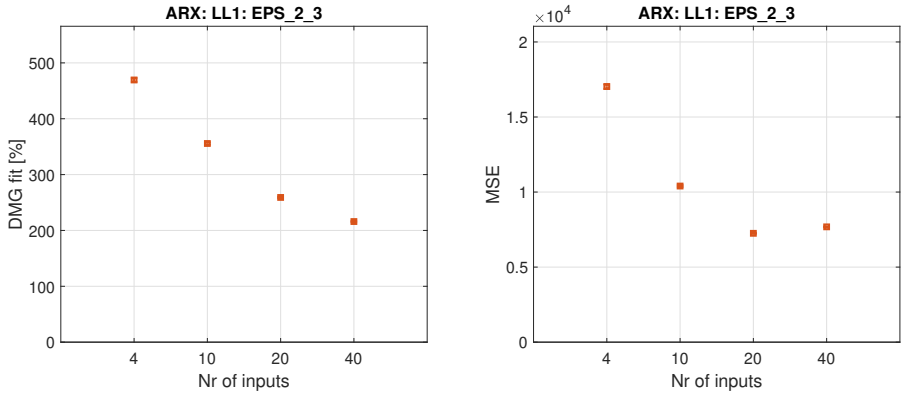


(a) *MSE* ***(b)*** *DMG*

Figure 5.12: *MSE and DMG comparison between a two step model and a regular model.*

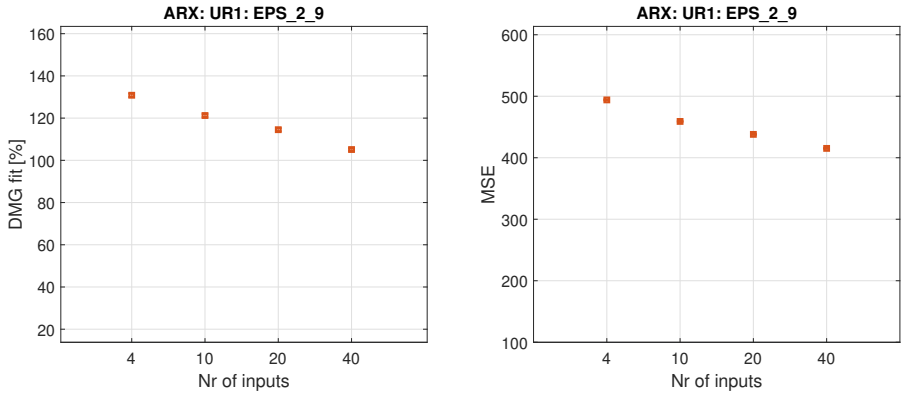
5.1.6 Number of input and input lags for linear models

Below are the results from comparing MISO models with a different amount of inputs. In Figure 5.13 - 5.14, plots of the MSE and DMG fit are displayed.



(a) DMG *(b) MSE*

Figure 5.13: *DMG fit and MSE for the EPS_2_3 estimate as a function of number of input signals.*



(a) DMG *(b) MSE*

Figure 5.14: *DMG fit and MSE for two different stroke classes and two different output signals*

The results show that more inputs improves the DMG fit of the model and lowers the MSE, regardless of stroke class or output signal. A thing to note in Figure 5.13. is that the linear models again have a hard time modelling main-frame EPS-signals.

When varying the number of input and output lags, changes in the residual analysis was apparent. As shown in Figure 5.15, generally, more dynamics are captured with an increase in input lags up until the maximum tested. The same holds for output lags. The residuals are plotted with the same reference threshold for whiteness because they were virtually the same. A saturation of performance increase was found when comparing the model with 40 input lags and 10 output lags to a model with 80 input lags and 10 - thus the 80/10 model was left out of the figure. The 40/10 (and 80/10) model shows the best overall performance on the cross correlation, but falls behind the 40/20 model when reviewing the auto-correlation. It is thus concluded that the more inputs used and the more lags of the used inputs improves the results. It should also be noted that most of the residuals are outside the thresholds, which suggests that the ARX-model structure is not sufficient for modelling the sensor.

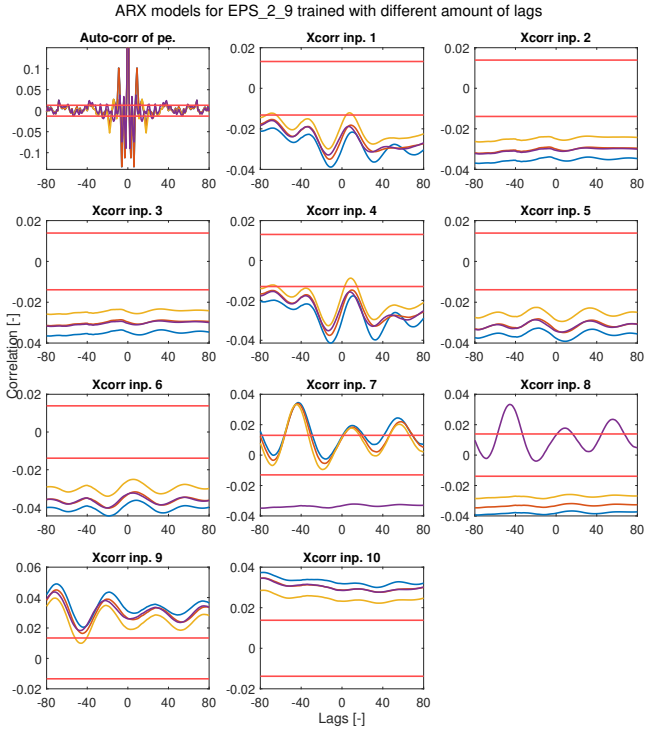
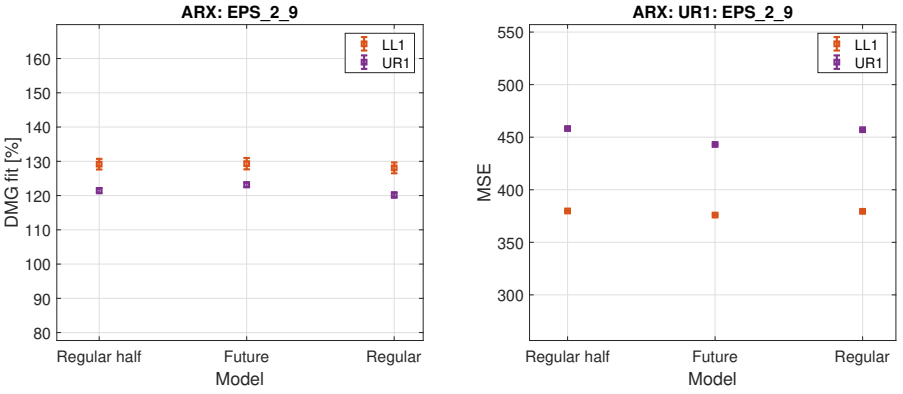


Figure 5.15: *Plot of the residual analysis made to decide the amount of input and output lags. Blue is the 15/3 model, red is the 40/3 model, yellow is the 40/10 model and purple is the 40/20 model. The red horizontal lines are the thresholds, inside which any correlations can be discarded.*

5.1.7 Future inputs for modelling

In Figure 5.16, a comparison of the regular models and models with future lags are shown. With regards to the MSE and DMG, the results show no noticeable improvements. The assumption that the HS-signals precede the RCS-signals thus holds.



(a) DMG *(b) MSE*

Figure 5.16: *Comparison of different models on two strokes, showing no significant improvement. Regular half = 40 lags, Future = 80 lags where 40 of them are future inputs, Regular = 80 lags.*

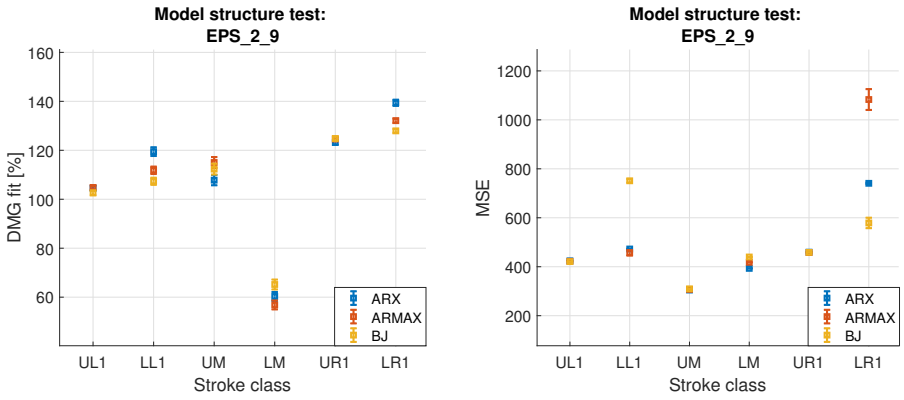
5.1.8 Best model structure for each sensor

The results of the sensors placed on the thrust-frame, with regards to MSE and DMG, are shown in Figures 5.17 - 5.19, and the results of the sensors placed on the main-frame are shown in Figures 5.20 - 5.22. The OE model structure under performed on most structures and was thus left out in the plots.

When reviewing the results, there are some outliers in play. For example, when looking closer at the EPS_2_3 sensor results, and more specifically the LM stroke-class file used for validation vs the model's estimate there is a sign change, as shown in Figure 5.23. Effects like these are caused by inconsistencies in the data set, and since they exist, the model that is viewed as the best for each sensor is the model that provides an overall best result. This particular inconsistency was a jack pressure sensor that had an inverted behaviour.

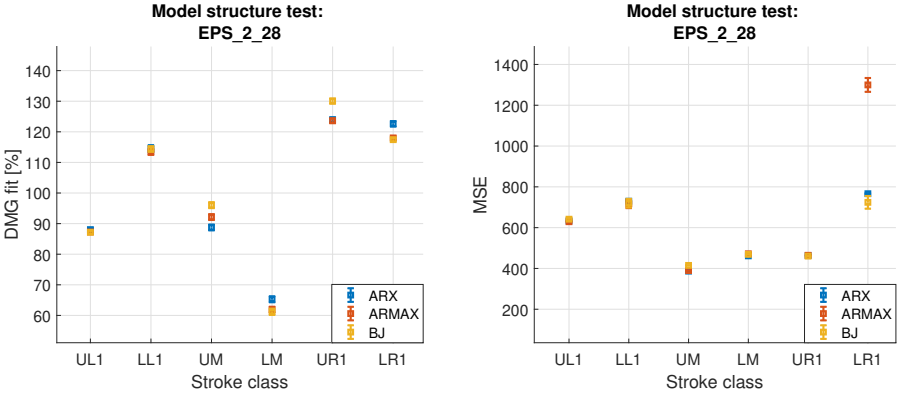
When comparing the degree of success of the structures for the sensors EPS_2_28 and EPS_2_9 on the thrust-frame, except for the poor performance of the OE-model, no model structure is significantly better or worse than the other. Since the ARX-model structure is the simplest and most efficient, the results suggest that it is preferred in the case of the mentioned sensors. Regarding the performances of the models on the derived F_lift_tot signal, it is apparent that the BJ

model structure performs the best. It is important to note the size order of the MSE for the models evaluated on the thrust-frame, as it is significantly lower when compared to main-frame models. The result does reveal difficulties when modelling main-frame sensors with linear models, even though two non-linearly transformed signals were used. However, since the models was trained under the same conditions, the most fitting may still be selected. The results of the model structures when estimating signals on the main-frame have a larger variance. Judging by the DMG and MSE the best candidate is either ARMAX or BJ. The residual analysis of the ARMAX and BJ models are shown in Figure 5.24 - 5.29. The residual analysis reveals that the BJ structure manages to catch the dynamics better than ARMAX on EPS_2_7. BJ is thus the choice for the EPS_2_7 sensor. The residual analysis for EPS_2_1 and EPS_2_3 revealed that there is barely any difference between BJ and ARMAX. However, since BJ scores slightly more consistently on MSE and DMG, the BJ model is also the choice for EPS_2_1 and EPS_2_3.



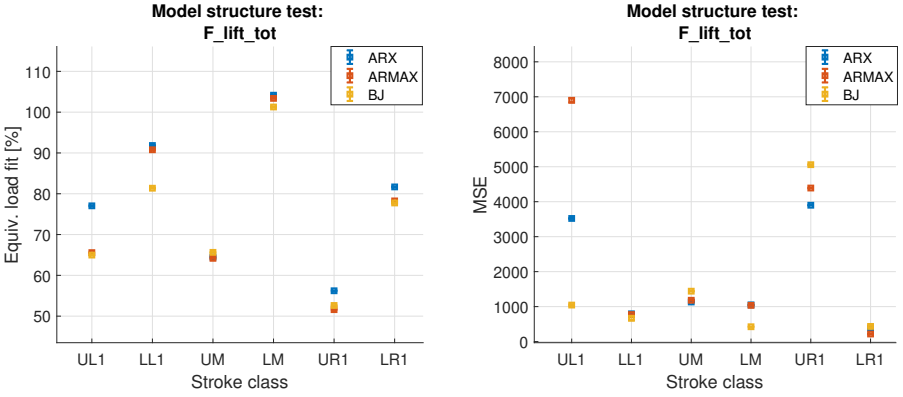
(a) *Output comparison* ***(b)*** *Zoom of the plot*

Figure 5.17: *DMG and MSE of EPS_2_9 models*



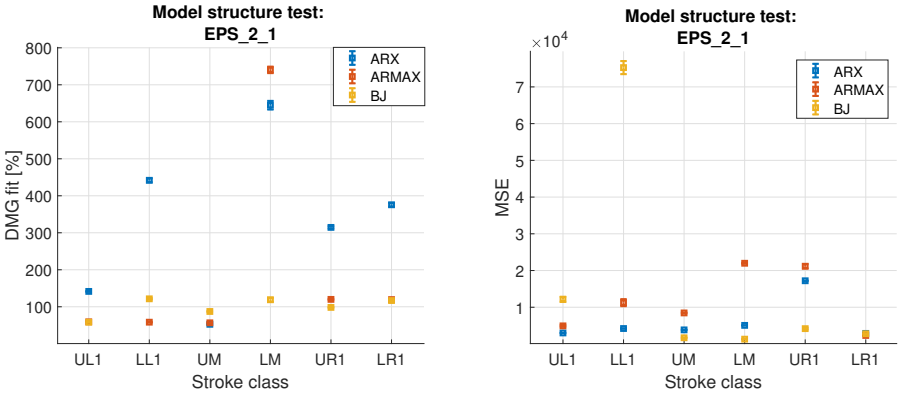
(a) *Output comparison* ***(b)*** *Zoom of the plot*

Figure 5.18: *DMG and MSE of EPS_2_28 models*



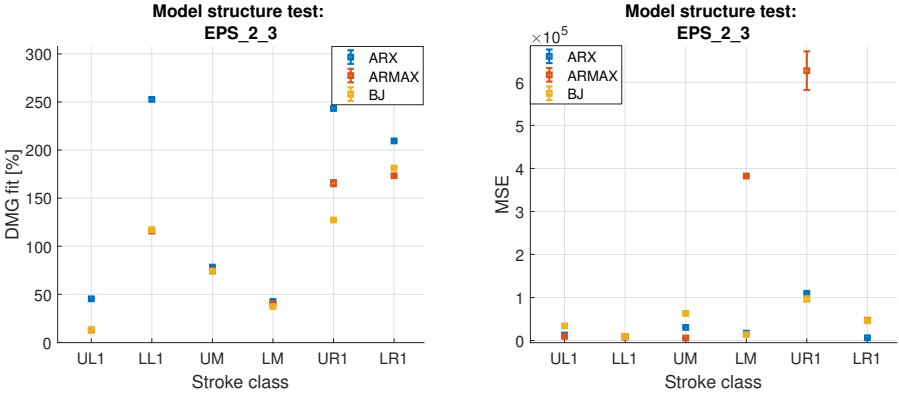
(a) *Output comparison* ***(b)*** *Zoom of the plot*

Figure 5.19: *EQUIV and MSE of F_lift_tot models*



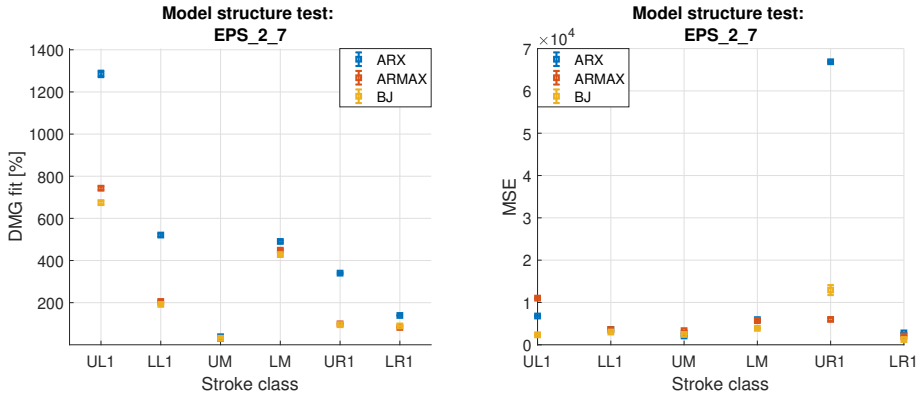
(a) *Output comparison* ***(b)*** *Zoom of the plot*

Figure 5.20: *DMG and MSE of EPS_2_1 models*



(a) *Output comparison* ***(b)*** *Zoom of the plot*

Figure 5.21: *DMG and MSE of EPS_2_3 models*



(a) Output comparison *(b) Zoom of the plot*

Figure 5.22: *DMG and MSE of EPS_2_7 models*

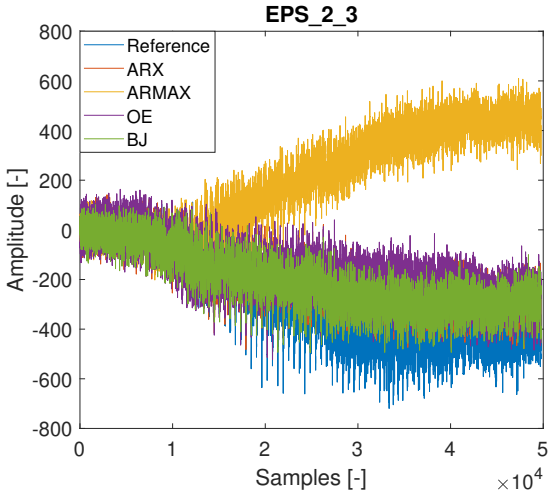


Figure 5.23: *Plot illustrating how different models follow the trend differently as a result of weighting inputs differently and changes in these.*

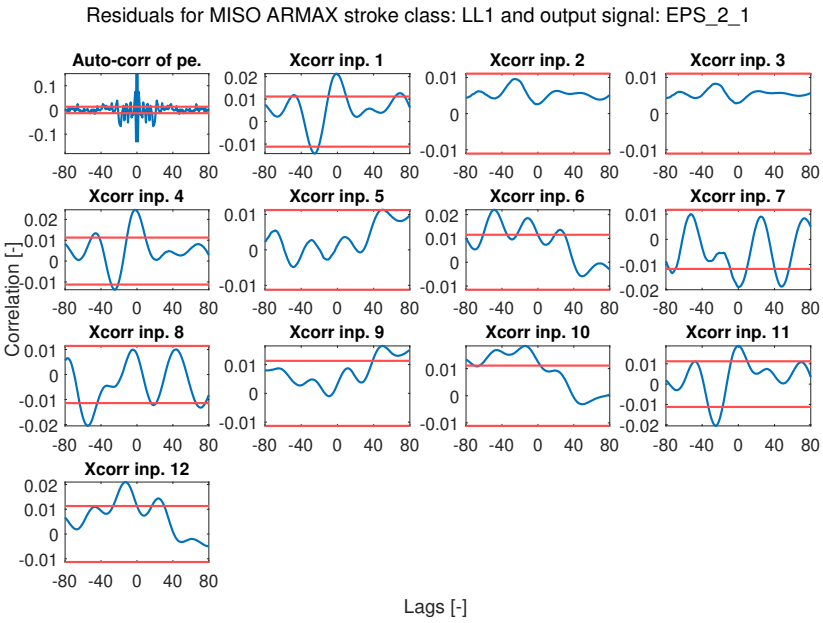


Figure 5.24: *Residual analysis of the ARMAX model for the EPS_2_1 sensor.*

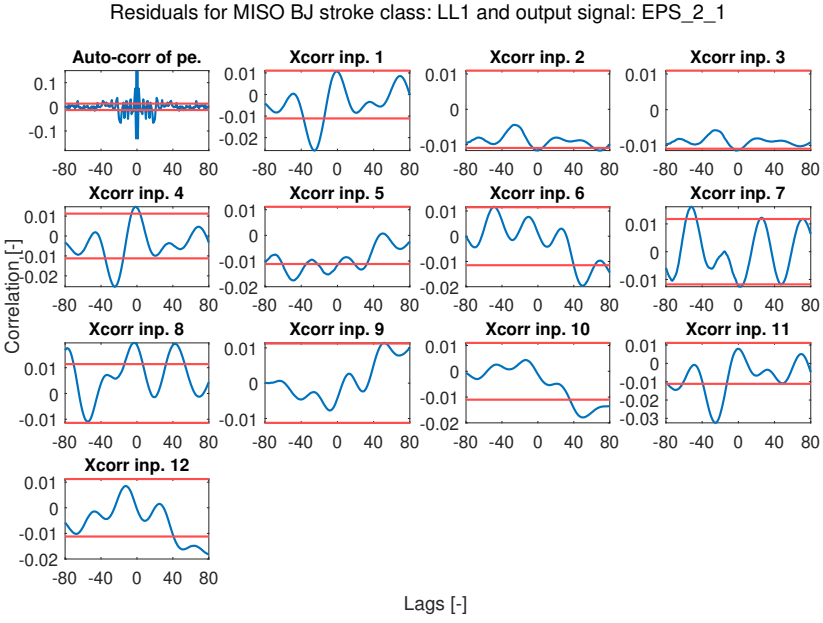


Figure 5.25: *Residual analysis of the BJ model for the EPS_2_1 sensor.*

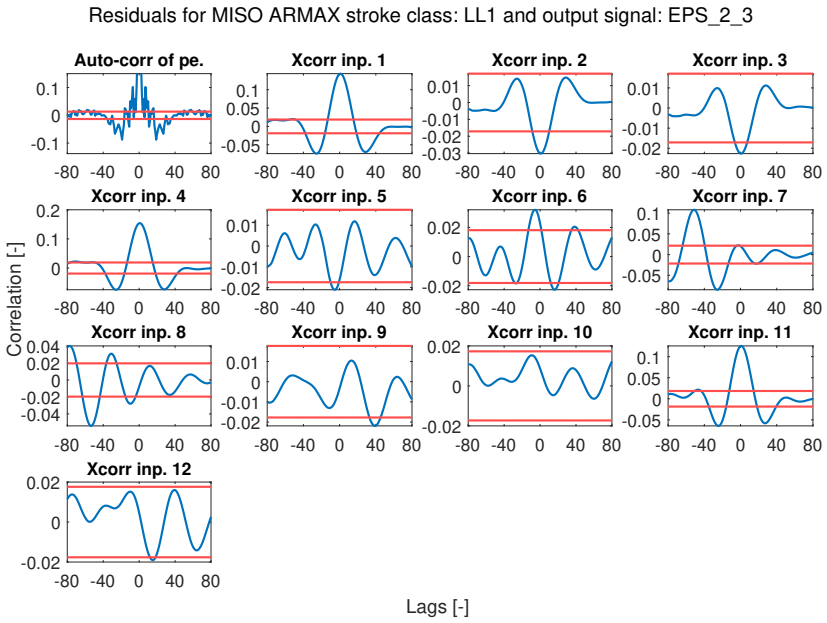


Figure 5.26: *Residual analysis of the ARMAX model for the EPS_2_3 sensor.*

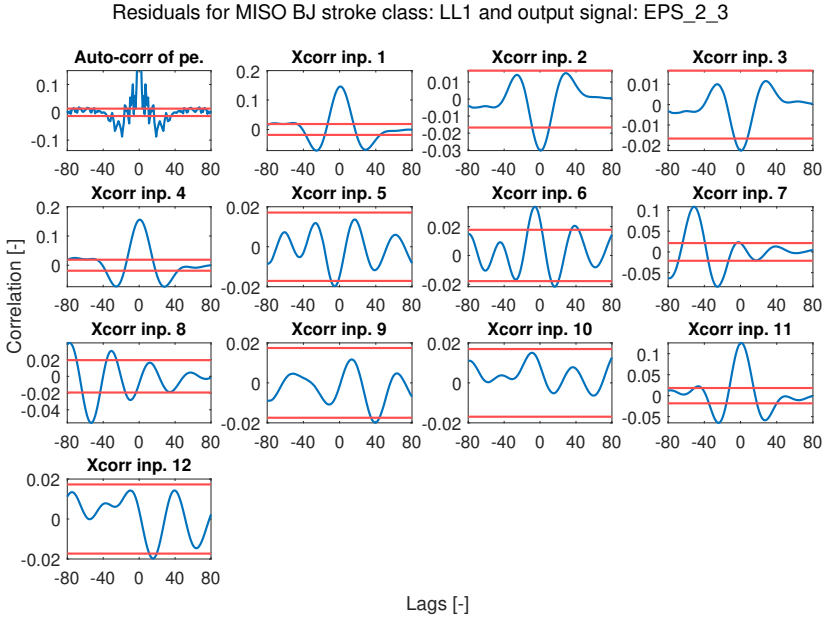


Figure 5.27: *Residual analysis of the BJ model for the EPS_2_3 sensor.*

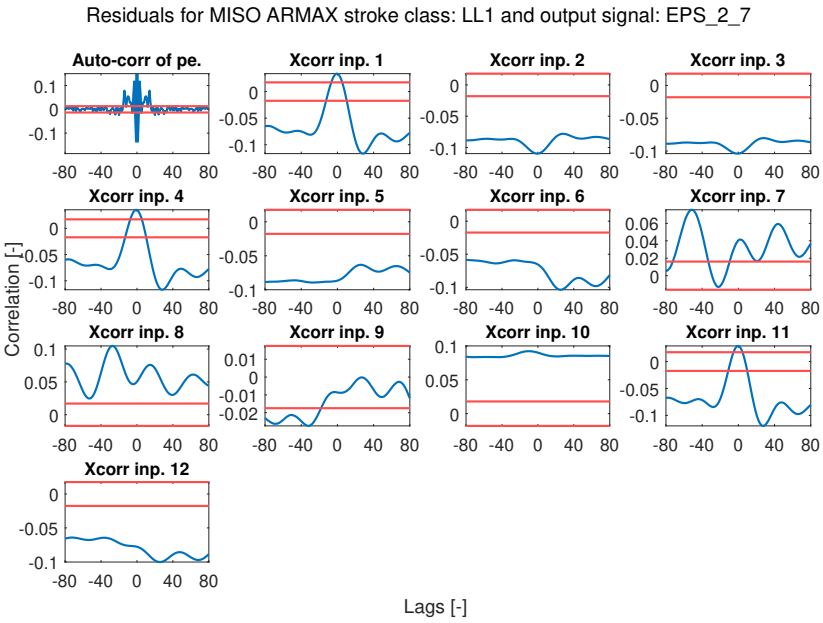


Figure 5.28: *Residual analysis of the ARMAX model for the EPS_2_7 sensor.*

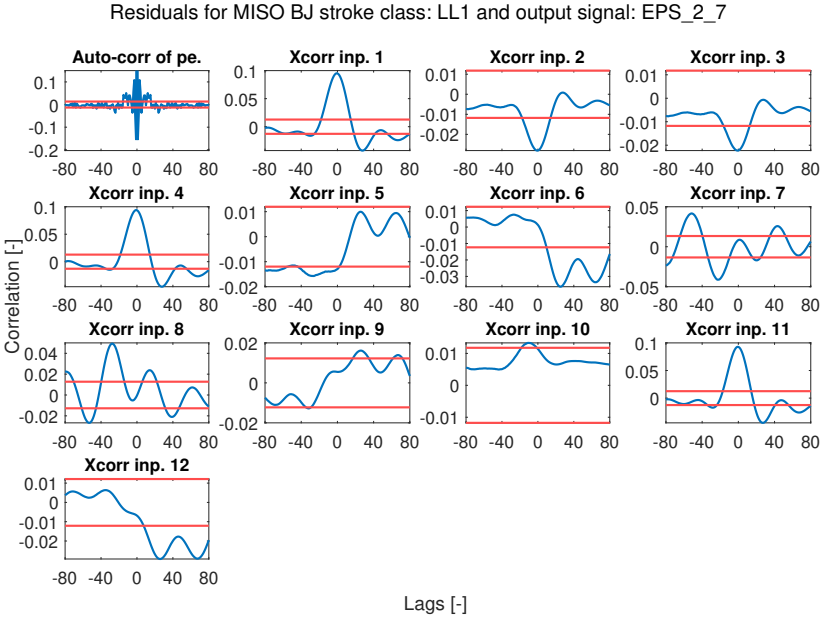


Figure 5.29: *Residual analysis of the BJ model for the EPS_2_7 sensor.*

5.1.9 Regularization tuning

The regularization proved to be highly important to help the model follow the reference trend and capture the input signal's dynamics. The results are, however, somewhat contradictory with the goal of the model. Figure 5.30 shows the impact of the λ-value with regard to capturing the trend of the sample signal and Figure 5.31 - 5.33 shows the impact with respect to the residuals. It is obvious that $\lambda = 1$ yields the best trend following, but does also perform the worst when reviewing the auto-correlation. The results shows that a smaller λ is prone to over-fitting, as it performs the worst when the cross-correlation is considered. Given the objective $\lambda = 1$ performs the best, however, $\lambda = 100$ is a clear winner if the trend is not considered.

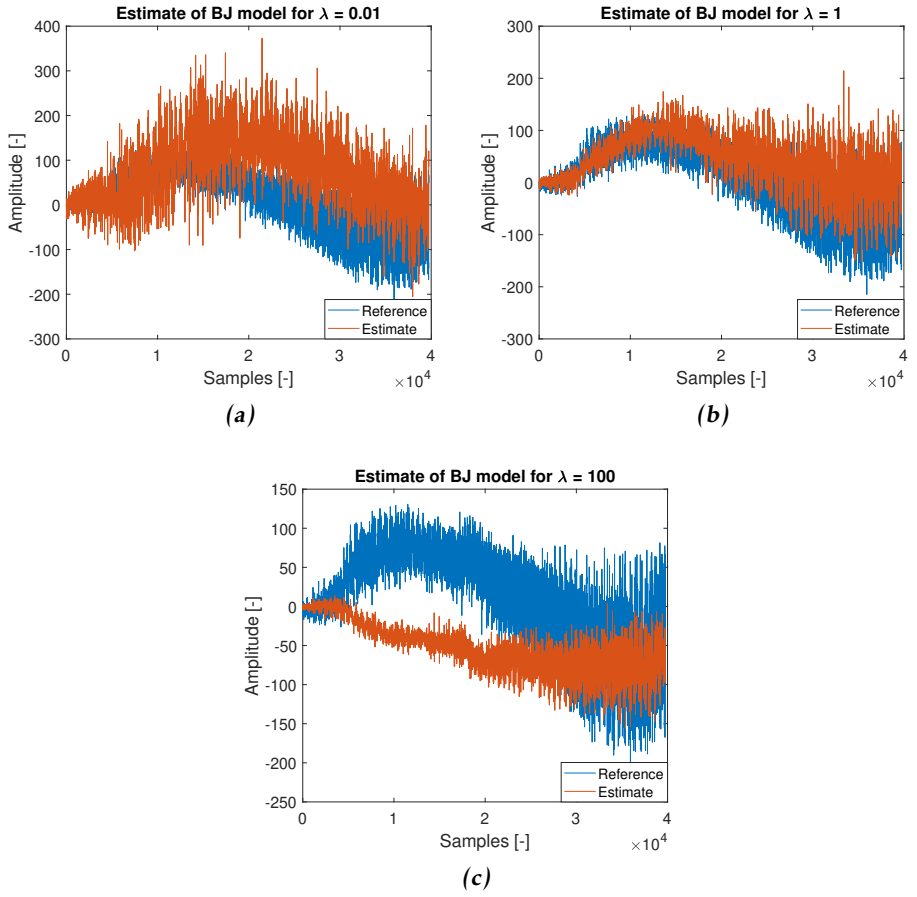


Figure 5.30: *Overview of how well the estimate manages to capture the trend as a function of the λ-value.*

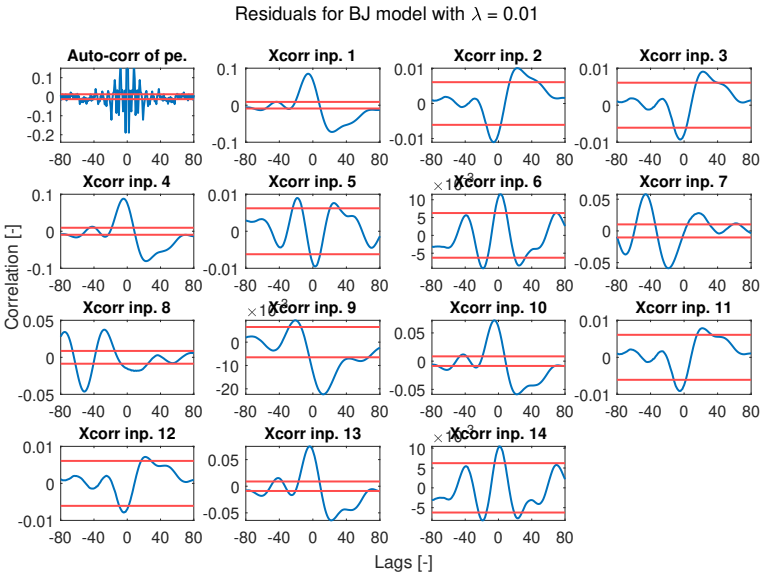


Figure 5.31: *Residual analysis of the regularization test of a BJ model with* $\lambda = 0.01$.

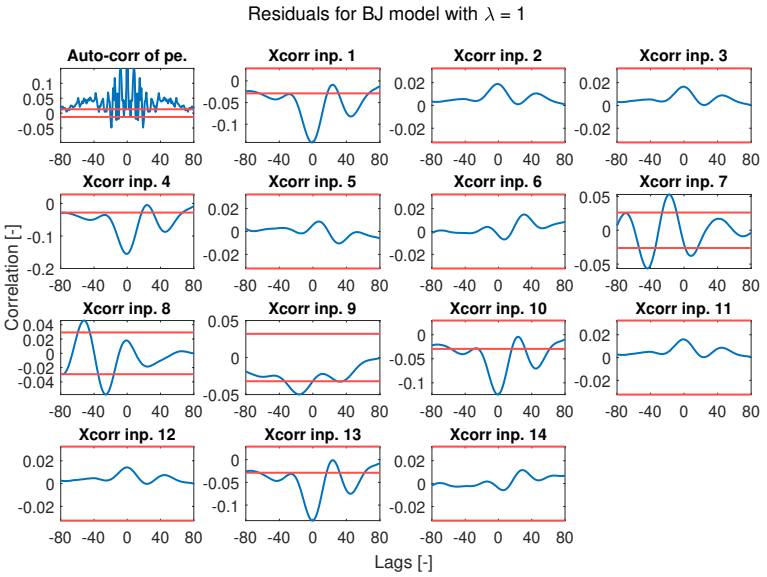


Figure 5.32: *Residual analysis of the regularization test of a BJ model with* $\lambda = 0.1$.

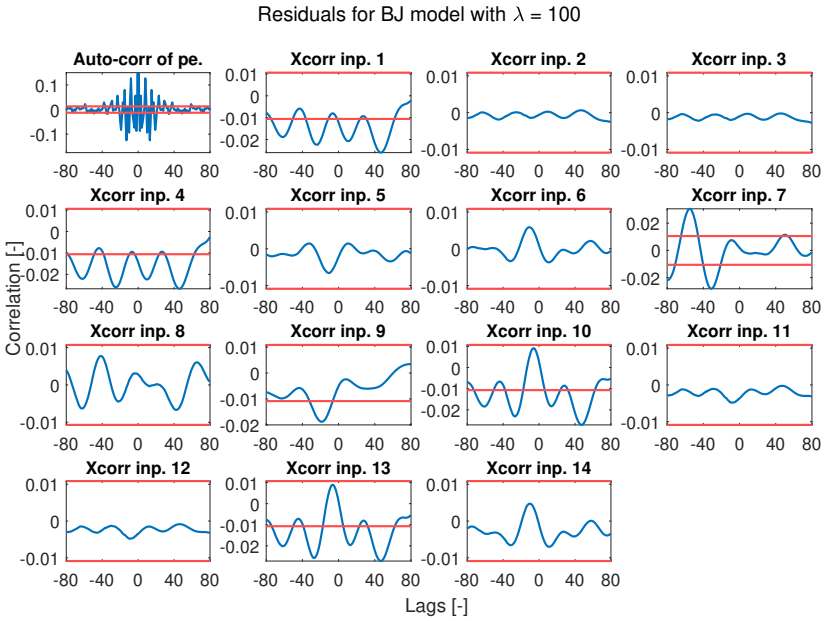


Figure 5.33: *Residual analysis of the regularization test of a BJ model with* $\lambda = 100$.

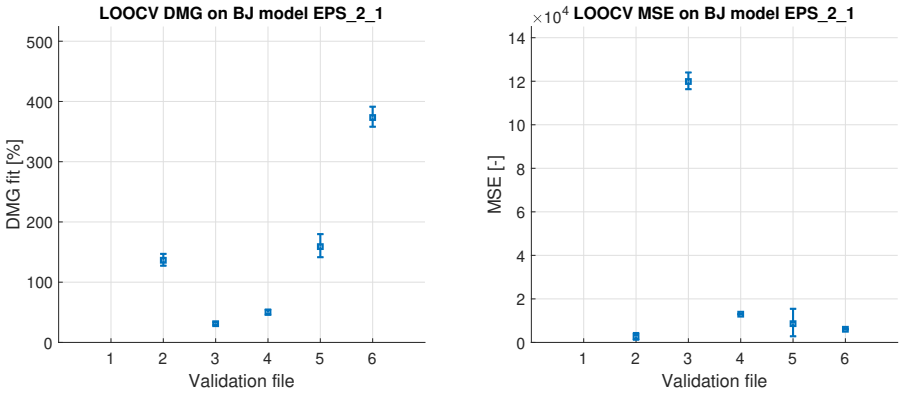
5.1.10 LOOCV evaluation results

The LOOCV evaluation of the EPS_2_1 model shows, in Figure 5.34, that it is very hard for linear models of main-frame sensors to capture the trend while also providing a good DMG estimate. Figure 5.35 - 5.37 reveals that the thrust-frame sensors are somewhat more consistent, with some very accurate estimates. However, while they are more consistent than the main-frame sensor estimate, the result is not satisfactory. Some LOOCV evaluations was conducted twice, with the inclusion and exclusion of the low tunnel recordings. Generally, one can see that the exclusion of the low tunnel recordings increases the performance of the DMG estimate, but while doing so also increases the MSE. Figure 5.38 shows validation file 2 of the LOOCV for EPS_2_28, which showed a large improvement on DMG estimate but also a large increase of MSE. The plot reveals that the cause of the larger MSE is an off-set. The off-set is believed to be introduced because, while the low tunnel recordings make the DMG estimate worse, the deviations of the recording forces the model to be more general. Thus, the inclusion of the low tunnel recordings contributes as a sort of regularization. The result of the comparison also suggests that the improvements of the categorization of angular settings are at least partly caused by that fact that the cutter wheel boundary conditions with the rock are categorized at the same time.

When reviewing the results of the F_lift_tot, especially the UR1 stroke class, its

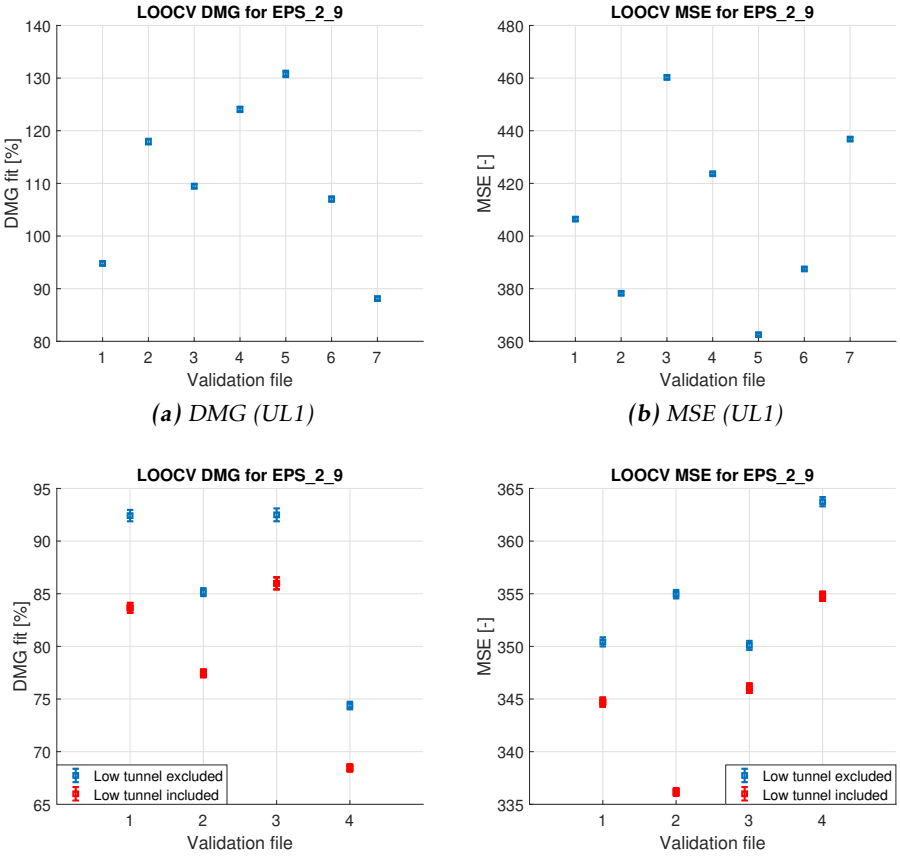
found that there are large differences between validation files 1-3 and 4-6. This was found to be because of a faulty pressure sensor. Thus, in further investigations these recordings was left out as they serve as bad validation files and has a negative impact as training files. The same goes for validation file 2-4 in the LM stroke class. While the impact is not as clear in the results, these files are also left out in further analysis. As the exclusion of these files substantially reduces the amount of available data, the LM stroke class is not further evaluated.

One may derive a pattern, where different sensors are more or less excited depending on the stroke class. When viewing the performance of the EPS_2_9 and EPS_2_28 models in Figure 5.35 - 5.36, the DMG estimate is more accurate and consistent and the MSE is lower for the LM-class. Whether this is caused by large variations of structure excitation in the training data set or simply because the strain signals are easier to model during LM strokes is hard to say from the shown plots. The topic is further evaluated in Section 5.2.2.



(a) DMG *(b) MSE*

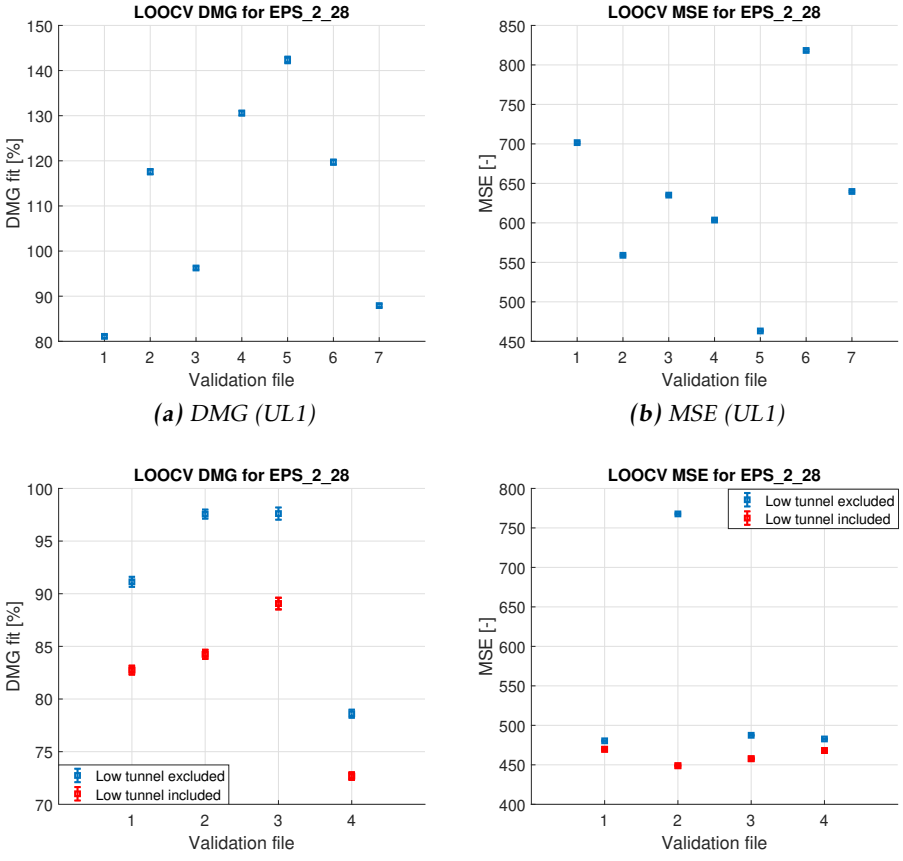
Figure 5.34: *A plot of how the DMG and MSE inconsistently varies between validation file of the same stroke class. Validation file 1's estimates are not shown because the DMG fit % and MSE are too high.*



(a) DMG (UL1) *(b) MSE (UL1)*

(c) DMG (LM) *(d) MSE (LM)*

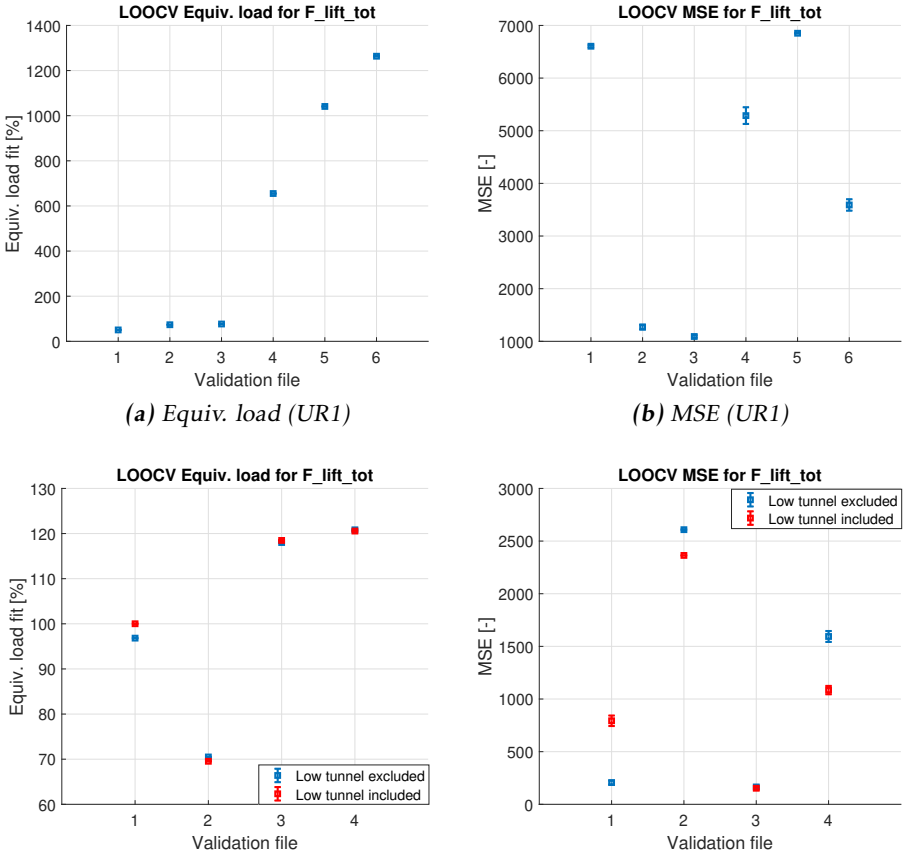
Figure 5.35: *The results of the LOOCV evaluation of EPS_2_9. The LM stroke class compares the LOOCV when including and excluding the low tunner recordings.*



(a) DMG (UL1) *(b) MSE (UL1)*

(c) DMG (LM) *(d) MSE (LM)*

Figure 5.36: *The results of the LOOCV evaluation of EPS_2_28. The LM stroke class compares the LOOCV when including and excluding the low tunner recordings.*



(a) *Equiv. load (UR1)*

(b) *MSE (UR1)*

(c) *Equiv. load (LM)*

(d) *MSE (LM)*

Figure 5.37: *The results of the LOOCV evaluation of F_lift_tot. The LM stroke class compares the LOOCV when including and excluding the low tunner recordings.*

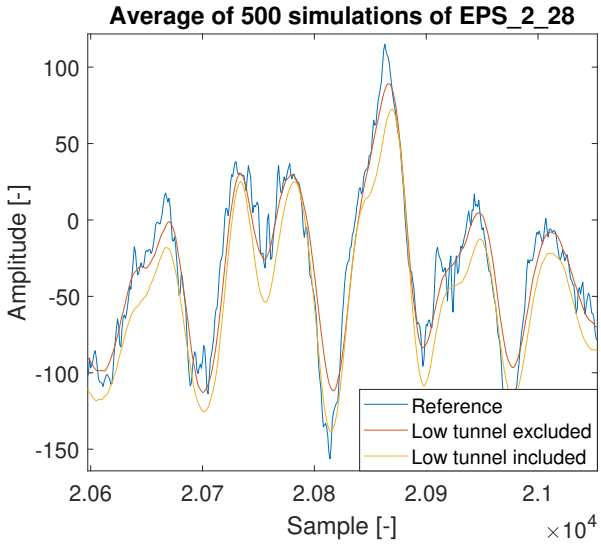


Figure 5.38: *Plot of how the average of 500 simulations of EPS_2_28 changes when including and excluding low tunnel recordings, compared to a reference signal*

5.2 Linear models for fast dynamics

5.2.1 Regularization tuning of detrended models

The residual analysis results from tuning the regularization parameter λ on models created from the detrended data set are shown in Figure 5.39. The results show that $\lambda = 100$ performs worse in both auto and cross correlation. $\lambda = 1$ and $\lambda = 0.01$ perform very closely in regards to the cross correlation, while $\lambda = 0.01$ has slightly less auto-correlation of the residual. This holds across all stroke types. When taking into account the DMG and MSE there is a slight performance increase when using $\lambda = 0.01$ and therefore this will be used in further tests on detrended data.

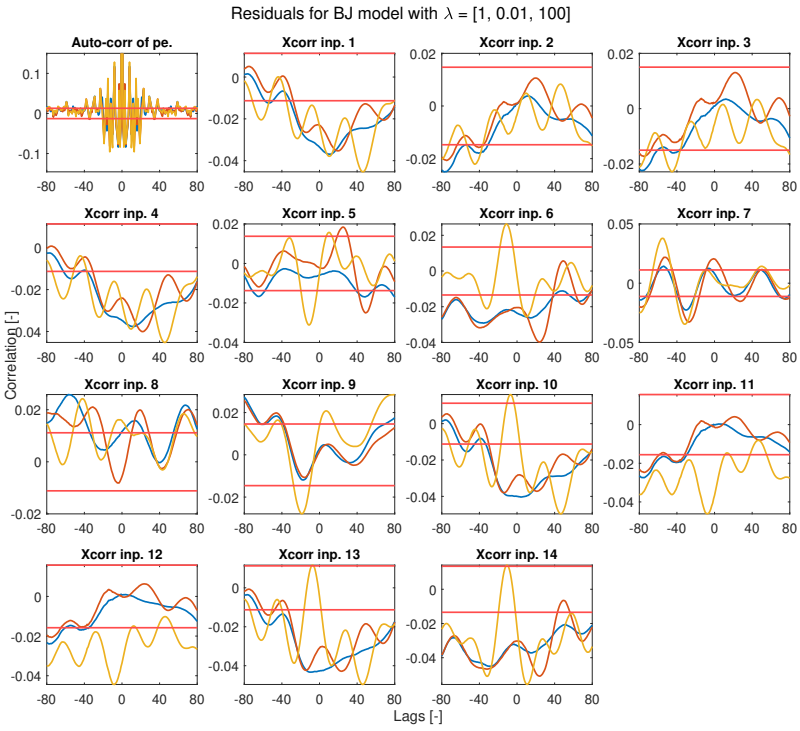
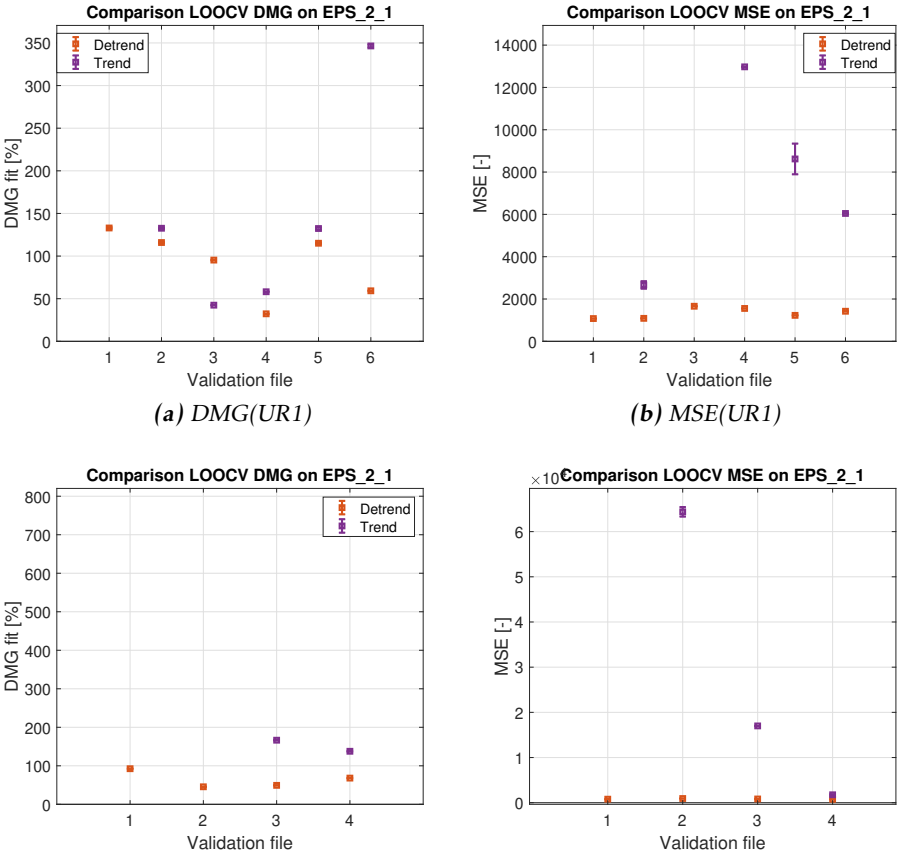


Figure 5.39: *Plot of the residual analysis for the regularization tuning of detrended models Blue is $\lambda = 1$, red is $\lambda = 0.01$ and yellow is $\lambda = 100$*

5.2.2 LOOCV evaluation of detrended models

Figure 5.40-5.43 show the DMG and MSE for the LOOCV test on detrended data compared to the results for the non-detrended data. When the model does not have to take the trend into account it performs much better in terms of MSE. This is in line with the expected results, since it has been shown that linear models have a hard time capturing the trend of EPS signals, especially the ones located on the main frame. The DMG fit in turn has a much smaller variance between validation files. It should be noted that although detrending the data considerably improves the main-frame results, there is only a slight performance increase on the thrust-frame sensors. With the DMG loss during the detrending procedure shown in Figure 3.4 taken into consideration, one may argue that the slight improvements of the detrending of the thrust-frame sensors are cancelled. However, given that the main objective is to estimate the DMG, it is overall preferable to detrend the data since the MSE is generally lowered, and the models are thus producing a more reliable DMG estimate. Even after the detrending, there still exists a difference in the ability to catch the dynamics fully when comparing the main-frame sensors and the thrust-frame sensors. If one reviews Figure 5.41d and 5.42d, the MSE of the detrended signal EPS_2_1 is almost the double of the detrended signal EPS_2_9 while the time signals are of approximately the same size. This means that the models still have troubles to accurately follow the reference when main-frame sensors are modelled.



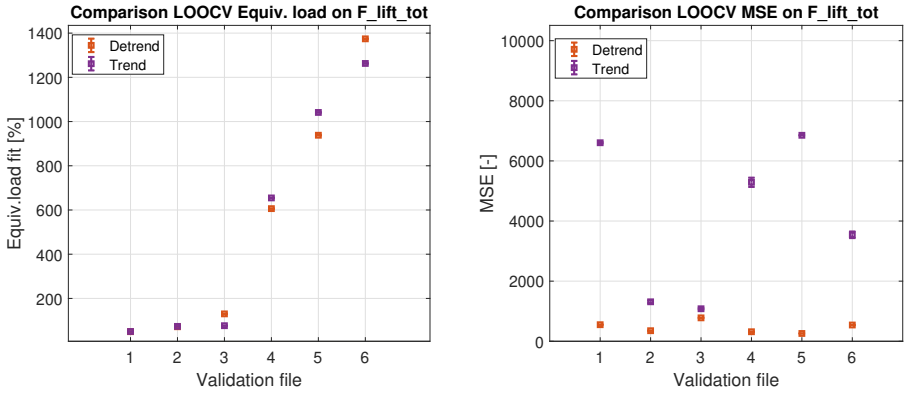
(a) DMG(UR1)

(b) MSE(UR1)

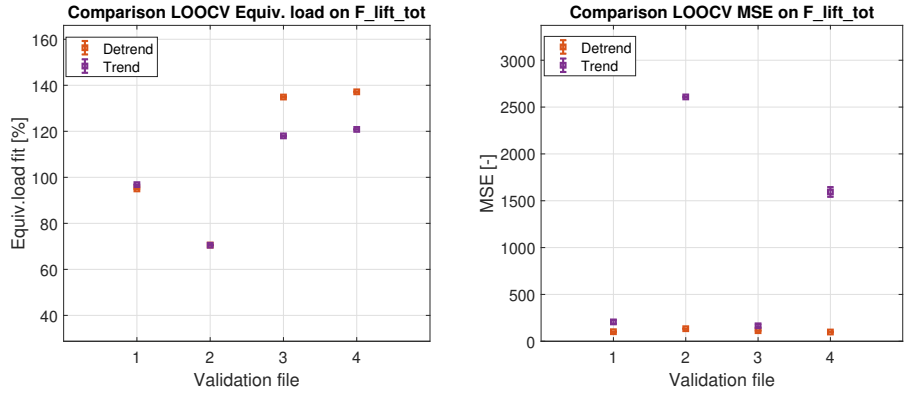
(c) DMG(LM)

(d) MSE(LM)

Figure 5.40: *A plot of how the DMG and MSE varies between validation files of the same stroke class for the detrended data. Some validation file estimates are too high and not included in the graph.*

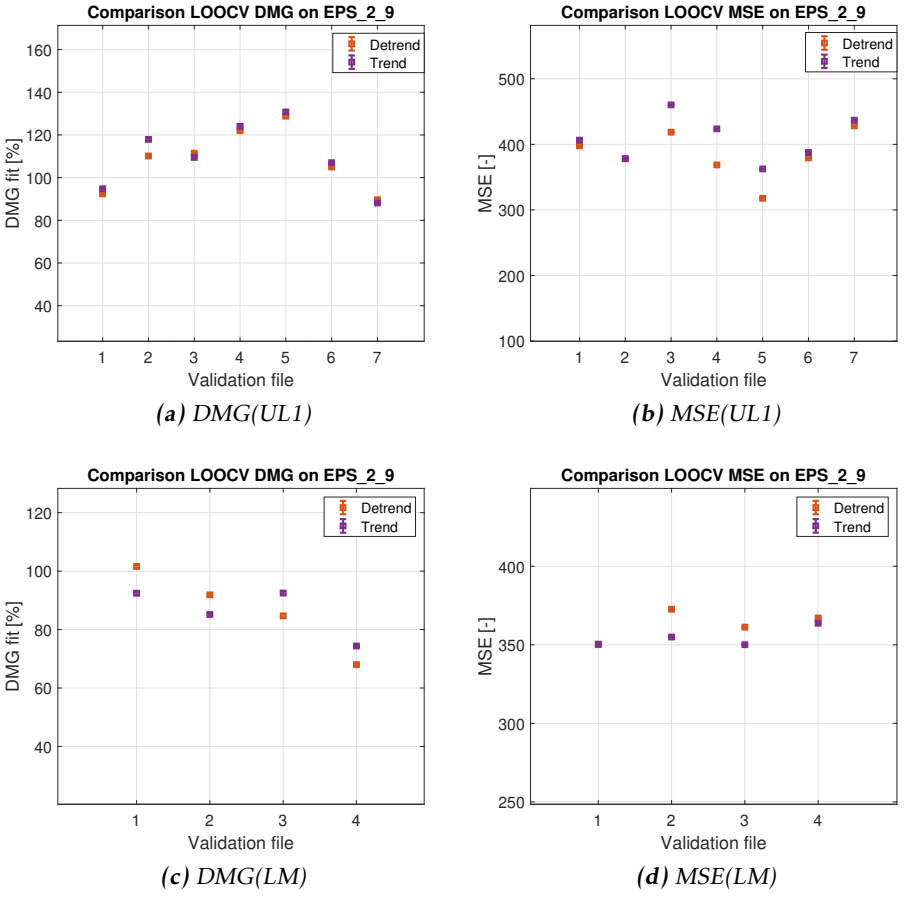


(a) *Equiv.load(UR1)* ***(b)*** *MSE(UR1)*



(c) *Equiv.load(LM)* ***(d)*** *MSE(LM)*

Figure 5.41: *A plot of how the DMG and MSE varies between validation files of the same stroke class for the detrended data.*



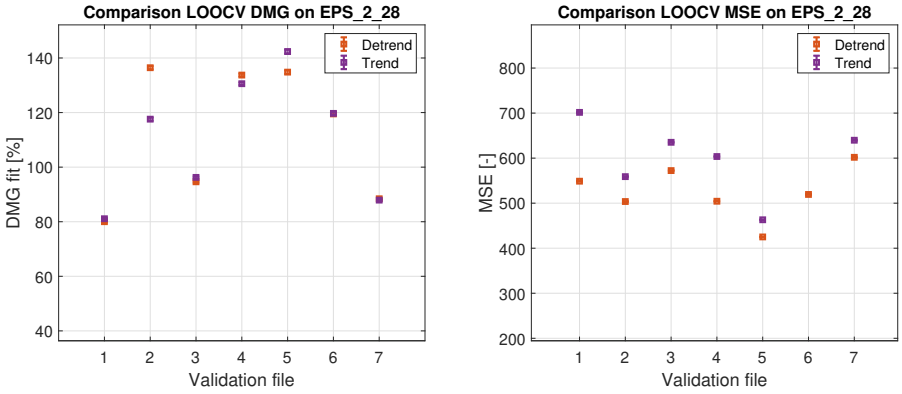
(a) DMG(UL1)

(b) MSE(UL1)

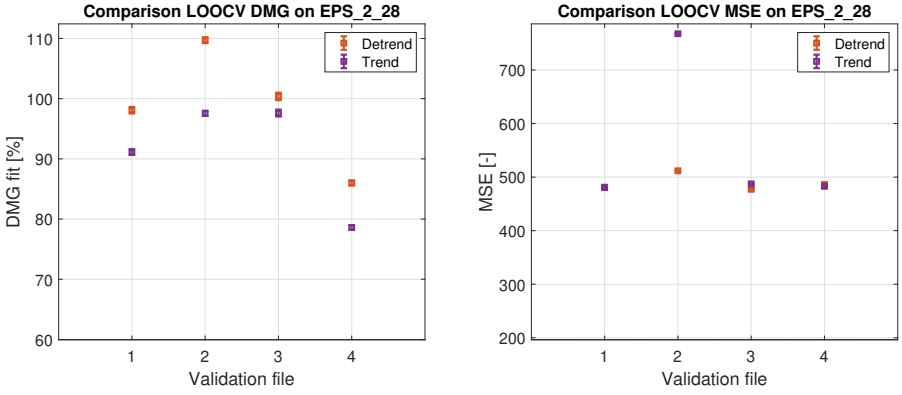
(c) DMG(LM)

(d) MSE(LM)

Figure 5.42: *Plots showing the difference in DMG and MSE for detrended and trend data on different validation files for EPS_2_9.*



(a) *DMG(UL1)* ***(b)*** *MSE(UL1)*



(c) *DMG(LM)* ***(d)*** *MSE(LM)*

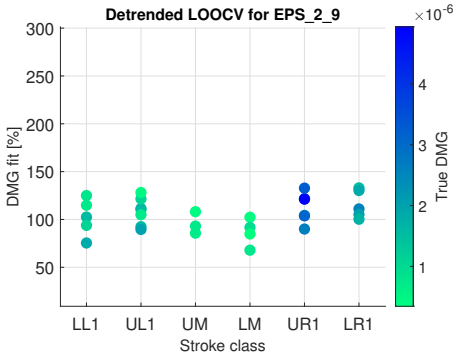
Figure 5.43: *Plots showing the difference in DMG and MSE for detrended and trend data on different validation files for EPS_2_28.*

LOOCV evaluation of all models

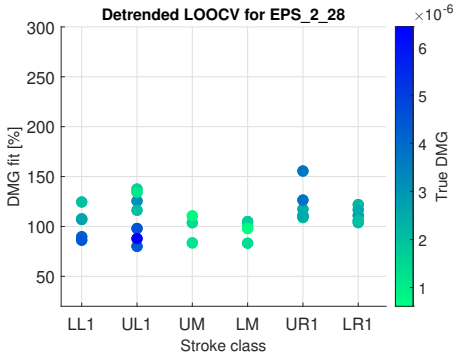
A comparison of the overall performance for each of the chosen EPS signals is shown in Figures 5.44a - 5.44d. The plots show how the results for an EPS signal vary depending on the validation file for each stroke class. The color represents the true accumulated damage for the specific validation file. The results for the sensors located on the mainframe seen in 5.44c - 5.44e are windowed as they contain outlier validation files which means they each have 1 to 2 files that give values way outside the reasonable range and are therefore discarded to make the plots readable. The results show that the sensors located on the main-frame have a much larger variance between validation files which is in line with earlier observations. As mentioned in Section 5.1.10 the results for F_lift_tot contains fewer validation files and no files for the stroke LM. While the result varies a lot, one can derive that a smaller variance of the true DMG of the training or validation files results in a more consistent estimate.

The reason why there are large variations in the accuracy of the estimate on sensors and stroke classes with a large variation of true DMG is believed to be caused by the noise model. During training, the noise model becomes the average of what would be suitable for high excitation and low excitation scenarios. This results in a noise model that overestimates validation files with low excitation and underestimates files with higher excitation. That is, it is concluded that the noise model gain holds a relationship to the level of system excitation.

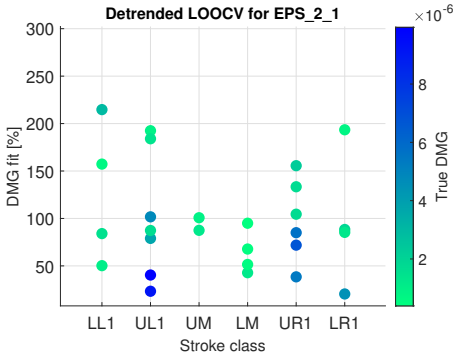
Figure 5.45c - 5.45b, shows the DMG estimations of EPS_2_9 using 3 different methods. Figure 5.45c are the DMG estimate for EPS_2_9 when simulated without noise, it mirrors previous results and again shows that the noise is important to better catch the peaks of the signals. Figure 5.45d shows the damage estimate when the outputs are simulated without noise, and then scaled with a factor of 1.15. Figure 5.45b shows the best case scenario, that is when high frequency data are available as input and thus represents the upper boundary of performance. Since the only difference is that the 4 top signals contain higher frequency information, it is apparent that what hinders the performance of the models the most is the low sampling frequency. One can also see that the DMG of EPS_2_9, which is placed on the right swing cylinder ear, is larger on strokes performed on the right side of the tunnel face, and DMG is the largest on the upper strokes. Figure 5.45c and 5.45d suggest that while the different scalings performs similarly, the realization of noise is the better option.



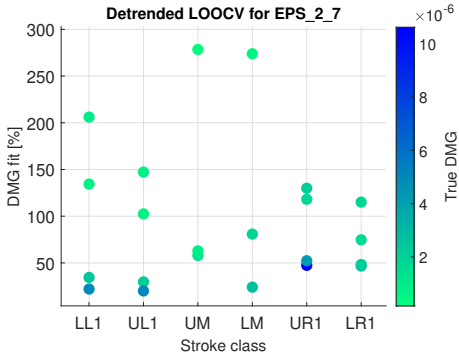
(a) EPS_2_9



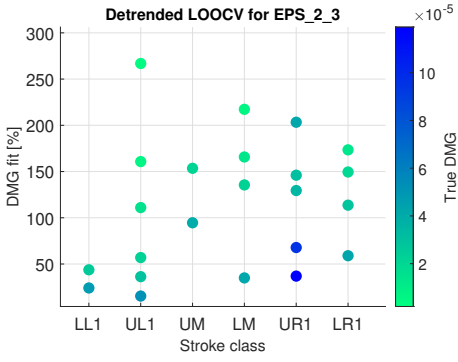
(b) EPS_2_28



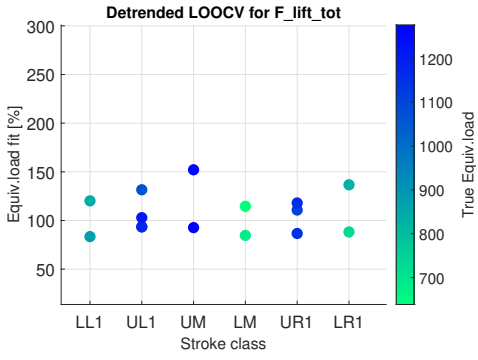
(c) EPS_2_1



(d) EPS_2_7

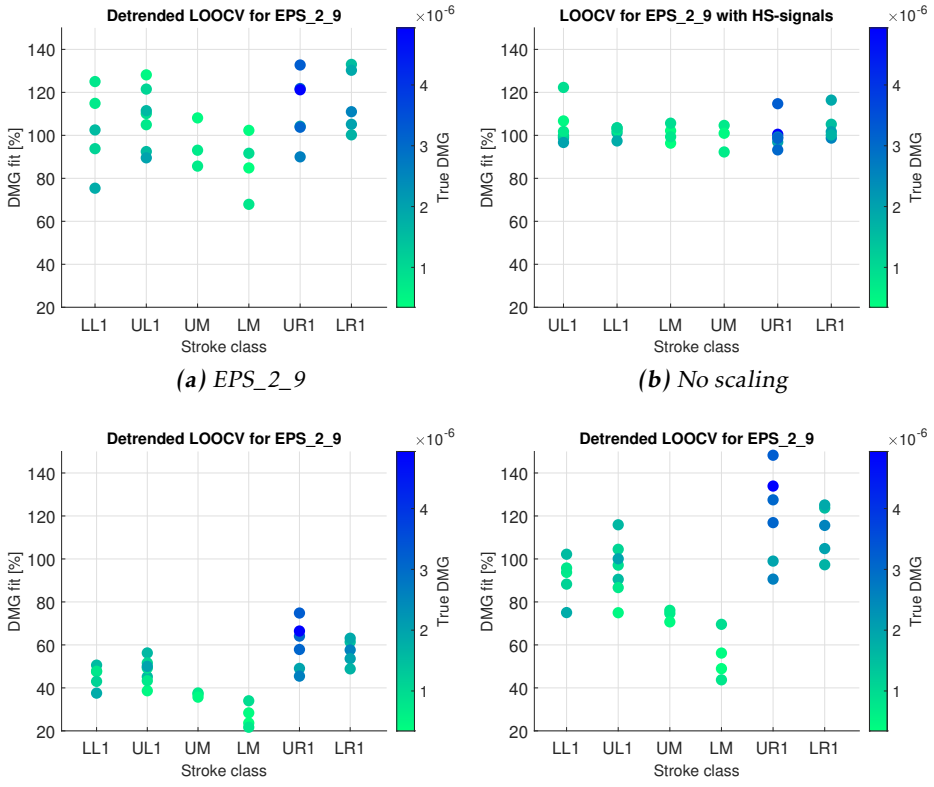


(e) EPS_2_3



(f) F_lift_tot

Figure 5.44: *LOOCV plots showing the estimation performance for all chosen EPS signals and F_lift_tot, each point is a different validation file. The color represents the true DMG of each validation file.*



(a) EPS_2_9

(b) No scaling

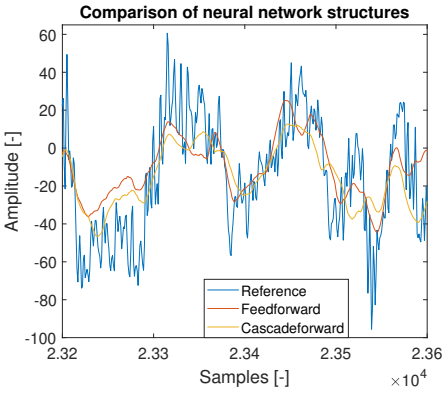
(c) No scaling

(d) Output signal scaled

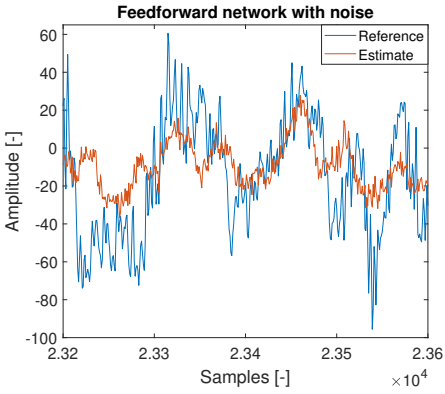
Figure 5.45: *LOOCV plots showing the estimation performance of EPS_2_9 with different scaling methods when not estimating the noise. Each point is a different validation file. The color represents the true DMG of each validation file.*

5.3 Non-linear models for fast dynamics

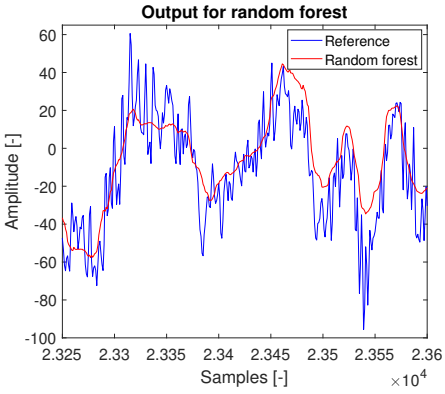
The first test was conducted to determine the better structure, and the results of the two tested structures are shown in Figure 5.46a. The feedforward structure performed better with regards to both MSE and DMG, even though the cascade-forward structure was trained with more parameters. Figure 5.46a shows that neither of the structures manages to capture the peaks of the reference, and the lack of the higher frequency content in the input signals affects the performance negatively. The test using more hidden layers and nodes gave roughly the same estimate shown in Figure 5.46a, and thus no increase in performance. The estimate in Figure 5.46b was obtained when training the neural network using a generated noise signal as input. It is apparent that the signal contains higher frequencies, but without any significant increase in performance. Figure 5.46c and 5.46d shows the estimate of the ensemble learners, which does a fairly good job of following the slower dynamics, but does not manage to catch the peaks and valleys. Figure 5.46e shows the estimate of a BJ model of the strain sensor. It serves as a good reference when comparing the performances, and shows that the linear model performs the best with respect to reference following. This is translated to DMG and MSE as well, where the linear model are superior. While this result show some promise for the non-linear models, it is concluded that the linear models performs the best, and the non-linear models are thus not further evaluated. It is believed that the feedforward network has trouble to utilize the noise input as good as the linear model, because of the way it is evaluated and trained - SGD that is. The reason behind the lower performance of the neural network could also be the technique used to select the input signals. That is, a technique based on the frequency contents may be more appropriate for the selected task.



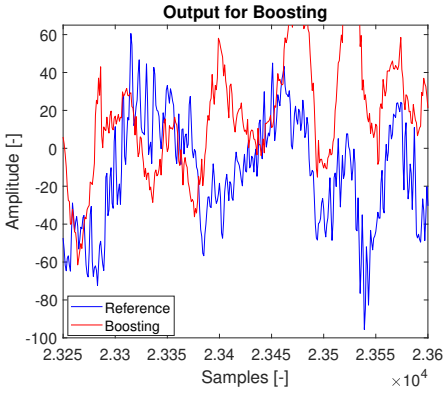
(a) Comparison of Feedforward and Cascadeforward networks



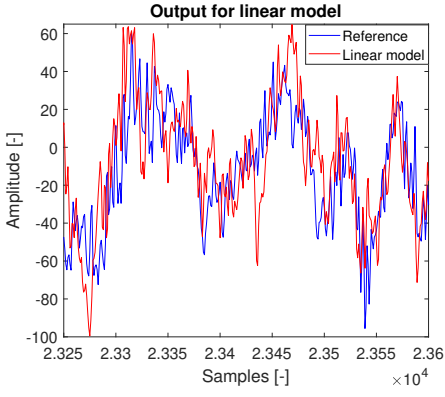
(b) Feedforward realized with noise input



(c) Random Forest



(d) Boosting



(e) Corresponding linear model

Figure 5.46: *Time plots of the generated outputs from the non-linear and corresponding linear model compared to the same reference. Estimations shown are done on the output EPS_2_1 and stroke class LR1*

6

General Discussion and Conclusions

6.1 General Discussion

The definition of the start of a stroke as a specific stroke length could result in some idle data points, where the cutter wheel was not in contact with the rock. This resulted in no strain variations. When simulating the models with noise, these portions of the validation files were estimated as noisy and thus contributed to the DMG estimate. The contribution to the DMG estimate was however deemed insignificant, as the amplitude of the noise was comparably low. The number of idle data points was small and it is thus assumed that it had an insignificant effect on the results.

The coherence measure was used by taking the average over the frequency span of the RCS-set. The frequency span may be selected more carefully if one wants. That is, one could look at the spectrum of the estimate and the reference, and if a certain frequency span is missing, the coherence analysis could be performed on that specific frequency span to easier pinpoint specific signals.

At some intermediate results of the methodology, it was concluded that the reason for some outliers were the fact that signals used as input behaved differently in the validation file. This can be seen in Figure 5.23. A possible way to provide more robust signals could be to use rod and piston pressure sensor pairs to derive the force exerted on the piston instead. This would however reduce the identifiability of each signal.

It should be mentioned that the damage estimated in the LOOCV-evaluation are the DMG after the signals have been resampled and high pass filtered, meaning that there are additional DMG loss and uncertainty added to the presented results. However, the estimated DMG is still compared to a True DMG derived

using the processed signal, since it is what the model tries to mimic. Should two DMG estimates produced by the same model, or two DMG estimates produced by models using the used methodology as described be compared it is deemed possible to truthful conclusions. While on the topic of the DMG estimate, it is necessary to mention that an improvement of the estimates can be expected if more of the available input signals and more lags of the inputs are used and even more so if they are sampled with a higher frequency.

Throughout the methodology, it was repeatedly shown that the main-frame sensors are hard to model. This problem was somewhat relieved by using a non-linear transformation of some input signals. A further investigation of suitable non-linear transformations could possibly improve the performance of the models of the main-frame sensors.

The results show that the key to a consistent estimate is an adequate noise model. With this in mind, it can be expected that the MIMO models should perform better than they did. The results of the MIMO models thus suggest that the non-existing high frequency components of the input signal holds a sensor unique relationship to the output. The LOOCV results later suggested that this relationship varies with the level of excitation of the system as well. This is particularly visible on the main-frame sensors, where the noise model is used to follow the reference to a larger extent.

The results show that the thrust-frame sensors are easier to model than the main-frame sensors. Instead of trying to model the main-frame sensors directly, one could examine if there exists any relationship between the DMG of the thrust-frame sensors and the main-frame sensors and create a model of that relationship instead.

6.2 Conclusions

This thesis asked three questions, the first was if available signals contain valuable information. It can be concluded that several of the available signals carry relevant information for the task at hand, and thus serve as good inputs signals to models of the strain signals. It is generally so, that the relevance of the input signals can be mechanically/physically motivated. That is, the placement of the modelled sensor relative to the input signals are important features. Several signals can, as mentioned, serve as good input signals to the models. The coherence measure serves as a good option to find input signals in the case of linear models. It is hard to determine if it is the non-linear model structure or the mutual information measure that makes the non-linear models inferior to the linear models. All input signals do however lack key high frequency components, which overall reduces the performance of the models.

The second question was whether the available input signals could be used to model the strain and force signals. The results show that the strain and force signals can be more or less accurately modelled. The performance of the models

are highly dependant of the available input signals, and if they are in proximity to the modelled sensor. The structure of the boring machine plays an important role in the difficulty of the task. Non-linearities between several of the most informative signals and the modelled sensors are introduced through the mechanical connection between the thrust- and main-frame. If appropriate non-linear transforms of the used input signals are applied, most of the sensors can be modelled arbitrarily well. Generally, regularized and categorized linear MISO models on detrended signals with overall trends removed performs the best - even though some DMG are lost during the detrending. It can be concluded that the categorization helps in two regards - categorizing the excavation settings and categorizing the boundary conditions with rock. The models are most successful on the thrust-frame sensors, and more input signals with more lags can be expected to increase the performance if the inputs' behaviours are properly captured in the training set.

The third question asked if the recreated strain and force signals could be used to accurately derive the DMG and equivalent load. The results show that the DMG can be accurately derived on certain sensors, but lesser so on some sensors. The equivalent load estimation is also accurately derived. The result of the equivalent load has lower robustness because of smaller amounts of available data. When using the estimated strain and force signals to derive a DMG or equivalent load estimate, the quality of the modelled signals are put to test and because of the difficulty during the modelling of the main-frame sensors, the DMG estimates of these are unreliable. However, the estimates of the thrust-frame sensors are successful, and the largest difficulty lies within recreating a non-existing part of the input signals. So given the complexity of the task and possible inaccuracies introduced by the pre-processing, the thrust-frame sensors are successfully estimated.

6.3 Future Development

In this section, some suggestions for future developments are made.

Possible improvements of used methodology

As mentioned in the discussion, a possible improvement to the methodology could be to do a more thorough investigation of non-linear transforms of input signals. One could write a script which creates a selection of non-linear transforms of input signals, and then evaluates the success of the different transforms by applying the coherence analysis.

It is likely that the impact on the DMG from the high frequency components are increased when larger excitations occur. That is, the true noise model should differ between high and low excitation sections. Because of this, a possible improvement could be to split the stroke in two based on the stroke length and categorize the models even further to get a better high frequency content description for both the low and high excitation sections.

Possible implementations of the models

The DMG fit for the estimated thrust-frame sensors was relatively consistent and centered around 100%. Even though the sample size is roughly 7 or less for each stroke class on the thrust-frame sensors, one could assume a gaussian distribution and fit the distribution of the DMG fit for each model. When using the Mobile Miner, the mean of several DMG estimates of consecutive strokes could be derived, and then the confidence interval based on the fitted distribution can be used to find if variations are explained by model inaccuracies or by an actual increase in DMG. The cause can then be identified and avoided in future excavation settings. The mean can be derived iteratively, to find differences online, or on an offline data set to find differences between specific periods of the excavation settings. Preferably, the DMG estimate should be normalized with either the stroke length or the stroke length per unit of time. Such a normalization would put the DMG estimate in a DMG to productivity comparison and context. The start of a stroke could then either be determined using the method used in the thesis, or a specific level of excitation of one or more measured RCS-signals.

A measure of how sure one can be of the estimated mean is Hoeffdings inequality, which is stated in (6.1). It assumes that each estimate is bounded, independent and identically distributed. It presents an upper bound on the probability that the sample mean deviates a certain amount from the true mean. b_1 and a_i are the upper and lower bounds of the i:th estimate X_i, respectively. n is the number of total estimates, $\bar{X}$ the mean, and t the comparison threshold [10]. As an example, take the maximum % deviation of the models for EPS_2_9 and stroke class UL1, which is approximately 20%. In this application, $b_i - a_i$ may then be derived as $b_i - a_i = 0.4X_i$ and t iteratively as $t = \xi\bar{X}$, where ξ are chosen as the degree of deviation from the true mean.

$$P\{|\bar{X} - \mu| \geq t\} \leq 2exp\Big(\frac{-2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\Big) \tag{6.1}$$

Since the suggestion of implementation presented above relies on an relatively accurate mean estimate, especially if small differences are to be detected, it should be noted that before any conclusion are drawn one must question the reliability of the mean estimate - and preferably use (6.1) when doing so. That is, several strokes within the same stroke class using the same settings, before any change are introduced, will increase the reliability of the estimate, and enable conclusions to be drawn with an increased certainty. It is also a necessity when using (6.1) because of the assumptions made. ξ may be tuned as one like, and since Figure 5.45a shows that the certainty of DMG estimates with similar corresponding true DMG are somewhat clustered, the factor of 0.4 in $b_i - a_i = 0.4X_i$ may be conservative and could be lowered.

Modelling of accumulated damage using extracted features

Modelling the strain signal has the advantage that the signal has a physical interpretation before it is used to derive the DMG. Another approach to create a model to estimate the DMG, is to find and extract appropriate features of the available data and use these to predict the DMG directly. That is, the true DMG is derived from the measured strain signals for each stroke, and then features of corresponding RCS-signals are extracted to find relationships. Given that the DMG is highly dependent on the frequency and amplitude of the strain signals, an example of a relevant feature to extract from the available data set is the variance of the inputs and max- and min-values.

Bibliography

[1] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550, 1994.

[2] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[3] Lennart Ljung Tianshi Chen. What can regularization offer for estimation of dynamical systems? *IFAC Proceedings Volumes*, 46(11):1–8, 2013.

[4] Turan Dirlik. *Application of computers in fatigue analysis*. PhD thesis, University of Warwick, 1985.

[5] George Ellis. *Control system design guide: using your computer to understand and diagnose feedback controllers*. Butterworth-Heinemann, 2012.

[6] Ramon Fuentes, Elizabeth Cross, Andrew Halfpenny, Keith Worden, and Robert J Barthorpe. Aircraft parametric structural load monitoring using gaussian process regression. In *EWSHM-7th European workshop on structural health monitoring*, 2014.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[8] Fredrik Gustafsson, Lennart Ljung, and Mille Millnert. *Signal Processing*. Studentlitteratur, 2010.

[9] Anders Hansson and Lennart Ljung. Some topics in system identification. `http://www.control.isy.liu.se/en/student/tsrt92/file/SystemIdentification.pdf`, 2020. Accessed: 2021-01-30.

[10] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

[11] Barbara Illowsky and Susan Dean. Testing the Significance of the Correlation Coefficient, 3 2021. URL https://stats.libretexts.org/@go/page/800. [Online; accessed 2021-04-15].

[12] Erik Jakobsson, Erik Frisk, Robert Pettersson, and Mattias Krysander. Data driven modeling and estimation of accumulated damage in mining vehicles using on-board sensors. In *annual conference of the prognostics and health management society 2017, PHM17, October 2-5, St. Petersburg, Florida, USA*, pages 98–107. phmSociety, 2017.

[13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112, pages 232–242. 2013.

[14] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[15] Quoc V Le, Jiquan Ngiam, Adam Coates, Ahbik Lahiri, Bobby Prochnow, and Andrew Y Ng. On optimization methods for deep learning. In *ICML*, 2011.

[16] Ljung Lennart. System identification: theory for the user. *PTR Prentice Hall, Upper Saddle River, NJ*, 28, 1999.

[17] Lennart Ljung. System identification toolbox: User's guide, 1995.

[18] Lennart Ljung and Torkel Glad. *Modeling and Identification of Dynamic Systems*. Studentlitteratur, 1:2 edition, 2016.

[19] Lennart Ljung, Carl Andersson, Koen Tiels, and T Schön. Deep learning and system identification. In *Proc IFAC Congress, Berlin*, 2020.

[20] V. Mathew, T. Toby, V. Singh, B. M. Rao, and M. G. Kumar. Prediction of remaining useful lifetime (rul) of turbofan engine using machine learning. In *2017 IEEE International Conference on Circuits and Systems (ICCS)*, pages 306–311, 2017. doi: 10.1109/ICCS1.2017.8326010.

[21] Mathworks. Statistics and machine learning toolbox user's guide. https://se.mathworks.com/help/stats/index.html, 2020. Accessed: 2021-02-05.

[22] Masanori Matsuishi and Tatsuo Endo. Fatigue of metals subjected to varying stress. *Japan Society of Mechanical Engineers, Fukuoka, Japan*, 68(2):37–40, 1968.

[23] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[25] Sangram Patil, Aum Patil, Vishwadeep Handikherkar, Sumit Desai, Vikas M Phalle, and Faruk S Kazi. Remaining useful life (rul) prediction of rolling element bearing using random forest and gradient boosting technique. In *ASME 2018 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers Digital Collection.

[26] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[27] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

[28] J. Schoukens and L. Ljung. Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, 39(6):28–99, 2019. doi: 10.1109/MCS.2019.2938121.

[29] Li Tan. *Digital signal processing: fundamentals and applications*, pages 584–589. Academic Press, 2008.

[30] Li Tan. *Digital signal processing: fundamentals and applications*, pages 558–567. Academic Press, 2008.

[31] Zeyu Wang, Yueren Wang, Ruochen Zeng, Ravi S Srinivasan, and Sherry Ahrentzen. Random forest based hourly building energy prediction. *Energy and Buildings*, 171:11–25, 2018.