University of Pretoria

Thesis

Optimal sensor placement approaches for the design of inverse experiments by simulation

Compiled by:
Younghwan Chae: 11085160

Supervisor
Dr. Daniel N. Wilke

March 13, 2017

Department of Mechanical and Aeronautical Engineering

It is hereby declared that this is my own work and appropriate reference has been made to other people's work.

Abstract

This dissertation serves to present the research conducted on sensor placement optimisation (SPO) using sensitivity analyses of virtual experiments in order to design virtual inverse problems. Two classes of SPO methods are considered namely mode-based and mode-free approaches. The mode-based approaches make use of SIMPLS and SVD to extract useful data by examining the correlation between the target variables (characterising variables) and the sensor measurement variables, while the mode-free approaches eliminate the need of spending the extra time required to extract modes, which ultimately leads to successful sensor placement for solving inverse problems. The aim of the mode-free approach is to maximise the variance explained subject to uniqueness of the information of each sensor. Both approaches aim to maximise the potential of an experimental setup to solve an inverse problem by using the right number of sensors and placing them at the optimal spatial positions. SPO is not only capable of designing an experiment but it is also capable of classifying the well-posed or ill-posed nature of an existing experiment that can be modelled, which saves both time and cost. The approach followed in this study was to design a simple virtual inverse problem for which the well or ill-posedness of the problem can be controlled. Numerous virtual experiments were conducted that varied from well-posed to severely ill-posed to allow for rigorous testing of the various approaches. The effect of model error and stochastic noise on ability to reliably place sensors is also investigated.

Acknowledgements

It is my absolute pleasure to acknowledge several people who have been a great help throughout my Masters course in Mechanical Engineering.

First and foremost, I would like to thank my academic supervisor, Dr. D. N. Wilke, for accepting me as his student. It has been a great pleasure and privilege having you as my supervisor since my final year as an undergraduate. Thank you for sharing valuable engineering inputs countless times and willingly spending immeasurable amount of hours for discussions. Your positive and passionate thoughts and approaches for tackling problems always encouraged and guided me to find a better direction for the study.

Secondly, I would like to thank the Centre for Asset Integrity Management (C-AIM) group at the University of Pretoria for accepting me and also to Prof. P. S. Heyns for financially assisting me since 2015. Thank you, Mr. G. Harley, for managing various meetings and administrations in the group. You really helped us to bring about productive work in the workplace.

Thirdly, I would like to give thanks to the National Research Foundation (NRF) in South Africa for financial assistance during my Masters. It certainly has helped me focus on the research and not worry about financial issues during the study.

Finally, I would like to thank my parents and brother in South Korea for trusting and encouraging me to finish my work successfully. Without their love and support I could never be where I am right now. My cousin, Haeun Noh, also deserves great credit and thanks for final editing of my thesis and always supporting me. This acknowledgement would not be complete without my colleague and officemate, Dominic Kafka, who has been an amazing, reliable friend for seven years and I thank him for sharing ideas and having work discussions with me.

Without the above-mentioned people, the completion of this thesis would not have been possible. Please accept my deepest gratitude and appreciation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The accuracy of the sensors, the variations of sensor types, sizes and application have improved over recent years [1]. Sensing various properties have become possible, for example, displacement, level, velocity, acceleration, force, strain, pressure, light, humidity, temperature and radiation. Hence, sensing has become accessible in many applications in various fields due to advancements in technology. Examples include GPS in a vehicle, gas bio-sensor and environmental monitoring [2, 3, 4]. In structural engineering, sensors are often used for vibration monitoring, nondestructive testing (NDT), material characterisation and crack identification [5, 6, 7, 8]. Traditionally, sensors were placed sparsely solely on engineering judgement and intuition. In cases where engineering judgement and intuition are limited, an exhaustive sensor placement strategy is often employed [9]. This often leads to time-consuming and expensive sensing solution strategies.

This thesis serves to present the results and effectiveness of sensor placement optimisation (SPO) for inverse experimental design [10]. Inverse experimental design is the construction of an experiment to estimate model parameters, i.e., the aim is to construct an experiment that will generate experimental data to identify model parameters, a.k.a. inverse problem. An inverse problem is concerned with computing causal variables that result in a known response for a given model or process. Therefore, the output of a model or process is known while the input needs to be determined. This is in contrast to the usual specification of the causal variables and computing the response, which is often referred to as the forward problem. Inverse problems are usually cast as a minimisation problem, with the aim to minimise the difference between the known response and estimated response for a specified set of causal variables. This error will be referred to as the inverse error to ease the discussion that follows. In the context of SPO, the casual variables of an inverse problem refers to the information we aim to infer from the sensors, referred to as target variables in this study, while the response is sensed using sensors, referred to as sensor measurement variables.

As a first approach, we could merely minimise the inverse error for different choices of sensor combinations and find the combination that solves the inverse problem most accurately with computational efficiency using some minimisation strategy. However, this approach is not tractable as the computational cost to compute the inverse error requires an inverse problem to be solved often multiple times, as the inverse error function is often multi-modal. Hence, immense efforts were put in to develop formulations for optimal sensor placement over the last four decades. Initial efforts were contributed by control engineers [11]. The aim of sensor placement optimisation (SPO) within the context of inverse experimental design is to identify the relevant information to solve an inverse problem. This is usually done virtually by virtual experimental design using simulation to replace the physical experiment. This means that SPO should be capable of indicating which sensors at which spatial locations should give the best sensitivities w.r.t. estimating model parameters. Hence, SPO does not only place sensors but it also assesses the quality of the experiment before it is physically constructed and assembled albeit only virtually. More importantly, when sensors are optimally placed, this allows for experiments to be quantitatively compared. In addition, engineers are enabled

to critically assess the implications of optimal sensor placement, sensor types within the context of model identification.

The formulations that support SPO are mostly based on sensitivity analysis [12] conducted in the form of a sensitivity matrix such as the Fisher information matrix (FIM) J^TJ [13]. Here, J denotes Jacobian, which is the first partial derivative of the residual vector, that defines the system, w.r.t. target variables. In SPO the system is usually given by the difference between the known response and estimated response i.e. the residual vector [14]. For inverse problems, FIM has been used to optimise the sensor placement by maximising the sensitivity of sensor measurements [15]. Instead of using the Jacobian of the residual vector, the mode shapes have also been used to construct the FIM [16]. Although the FIM can be constructed using numerous sources of information depending on the application, a salient feature of FIM is the fact that it captures the information measured by the sensors [17], which can then be directly used in the construction of some optimisation criteria. There exist various ways proposed to quantify the information in the FIM. Some optimality criteria and scalar values based on FIM for SPO are listed:

- Determinant (D-optimality): determinant of a matrix is defined as the product of eigenvalues of the matrix. Hence, maximisation of determinant of a FIM maximises information contained in the matrix [18, 19].
- Condition number: condition number can be defined as the singular value decomposition (SVD) [21, 20] ratio between the largest and the smallest singular values. Thus, minimising the condition number of FIM can maximise the sensor configuration [19].
- Norm or trace: maximising the norm [23, 24] or trace [25] of Fisher matrix results in the minimum possible values of estimation error [27].

Although FIM is a powerful strategy to examine the sensitivity of target variables w.r.t. measurement variables, its major deficiency is that the information obtained in the FIM is localised and therefore only applicable for small range target variables [26]. This can be problematic for sensor optimisation since the experiment for parameter identification can be carried out in different domains of the target variables. Although impractical, an approach that recomputes the FIM as the target variable is resolved, has been proposed [28]. FIM is not only restricted to sensor optimisation but also is widely used to model Gaussian processes (GP) [29, 30] as well as non-Gaussian processes [31, 32]. In this study, we will define these formulations as mode-free formulations which do not require any modes to be extracted from information or covariance matrices. On the other hand, we also investigate mode-based formulations that decompose information or covariance matrices into modes that allow us to place sensors optimally.

Due to the discrete nature of sensor placement, approaches to solve both mode-based and mode-free formulations often include some binary optimisation strategy. Binary optimisation strategies include meta-heuristic approaches [26]; exhaustive combinatorial search [36], simulated annealing (SA) [37] and genetic algorithms (GAs) [38, 39], with GAs the most widely used to optimally place sensors [40, 41]. Note that although sensor placement approaches centred around convex optimisation strategies have been researched [42], this study is restricted to binary GA optimisation strategies when large systems are considered as exhaustive search is intractable. In addition, a newly proposed sensor addition approach is also considered that scales well over problem dimensionality and found to be competitive when compared against both exhaustive search and GA strategies.

The existence of noise is unavoidable in real experiments and carelessly placed sensors can often capture excessive amounts of noise. Therefore, although sensors can deliver helpful information to solve inverse problems, if signal to noise ratio is too low the chosen sensors will perform poorly. The signal to noise ratio also varies with sensor type as the sensor accuracy varies. Hence, sensor types with variations in measurement accuracy may dominate the type and spatial location of a sensor. Fortunately, SPO can accommodate some of the

differences between real and virtual experiments to place practically relevant sensors and not only idealistic sensors in a noise free world.

The focus of this study is to propose and investigate various strategies to optimally place sensors. An example of an experimental setup that involves the characterisation of a material model is considered. Here, sensors need to be placed to capture information regarding the mechanisms in the material model that needs to be characterised at spatial locations not hampered by noise. This implies the signal to noise ratio should be as great as possible and at the same time, the sensors should be able to convey independent information to increase the chances for success in solving the inverse problem. A number of computationally tractable, yet, accurate and robust multi-purpose formulations (that act as proxies for solving the actual inverse problem), which avoid the localisation issues of the FIM by considering ranges of target variables are introduced. In order to avoid localised sensitivity issues, sample variance reduction strategies are considered [33], specifically, Latin hypercube sampling (LHS) [34] is considered in this study. These techniques increase the efficiency of sampling over a target variable domain which ultimately place sensors that are relevant over the sampled target variable domain.

Specifically, in Chapter 2 procedures to develop SPO formulations are presented. Various numerical techniques and statistical approaches are introduced to demonstrate their relationship to inverse problems. Depending on whether the proxies require eigenvalues and eigenvectors or not, SPO is divided into two parts:

1. mode-based formulations
2. mode-free formulations

Chapters 3-5 present and discuss the results of the mode-based formulations. In Chapter 3, various objective functions formulated using partial least square (PLS) [35] or singular value decomposition (SVD) [21] and numerous data scaling methods are proposed and their principles are discussed in depth. In Chapter 4, a combination of data scaling methods and objective functions are tested on a more complex problem to examine their generality and robustness. In Chapter 5, the promising proxies were tested using the simulation models with different mesh sizes to quantify the effect of model errors, which are errors between a simulation and reality that are always present to some degree, as each mesh is considered to represent a model error to some degree.

The mode-free formulations are discussed in Chapter 6. The mode-free formulations are developed by assembling the critical characteristics of the inverse problem assimilated in the previous chapters related to the mode-based formulations. Most of the investigations conducted for the mode-based formulations are repeated for the mode-free formulations and the results compared. Mode-free formulations aim to maximise the ranks of covariance matrix between the target and measurement variables as well as the variances of the sensor signals. In Chapter 7, the effects of stochastic noise to both the mode-based and mode-free formulations are presented and related to the well-posed nature of an experiment. Lastly, Chapter 8 introduces a modified version of the mode-free formulations called simplified mode-free formulation (SMF) and its performance is investigated.

Finally, the study is concluded in Chapter 9 as well as recommendations for future research given.

Chapter 2

Methodology, Procedures and Fundamental Investigations

In this chapter, the methodologies that were followed to develop the proxies for the inverse problems are presented using a simple cantilever beam example and the procedures of the complete investigations in the dissertation are also discussed in order to give the reader the full views and the logical steps behind the various investigations. This chapter also deals with the fundamental investigations which offer a clear understanding of the relationships between the inverse problem and multi-variate statistical techniques.

2.1 Methodology

Before discussing the methodology for developing proxies, the simple cantilever beam problem shown in Figure 2.1 is introduced. The aim of this experiment is to infer Young's modulus E from displacement u, stress σ and strain ϵ data [43]. This setup was deliberately chosen as it is impossible to infer Young's modulus from stress but strain and displacement are sensitive to Young's modulus. The model was built using finite element method (FEM) with eight-noded elements with reduced Gauss points (i.e. 2×2 Gauss integration points are used). The beam is 0.5 m high, 2 m long and 0.5 m thick. The 2D plane strain assumption was used with the Poisson's ratio, ν of 0.3. Young's modulus of the beam, E would usually be 70 - 200×10^9 Pa but is randomly manipulated between 10^5 - 10^7 Pa. The load applied at the tip is 1000 N at the 37th node downwards while the left side of the beam is completely fixed.

From each manipulation of the displacements, u are measured at each node and the stresses, σ and strains, ϵ are measured at each Gauss point indicated with crosses. This implies that every time E varies u, σ and ϵ get remeasured and the measuring was done over 100 times. Latin hypercube sampling (LHS) which is a random sampling strategy which focuses on the coverage of the sampling domain was implemented to generate random E values. Hence, it was possible to 'uniformly' evaluate the variable over its range [34]. In this

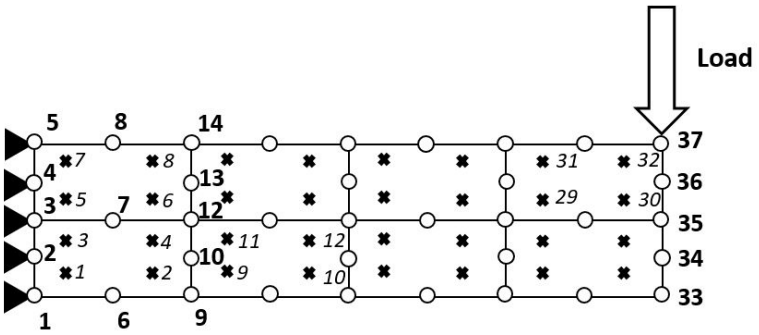


Figure 2.1: Simple cantilever beam bending problem.

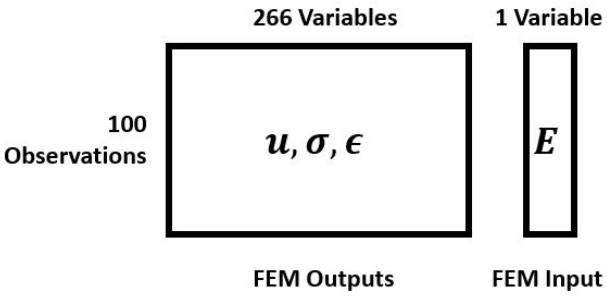


Figure 2.2: FEM measurement and target variables.

chapter, the finite element method (FEM) input variable E needs to be recovered from the inverse problem which we will refer to as the 'target variables' and the sensor measurement variables such as u, σ and ϵ are called 'measurement variables'.

Since the displacements, stresses and strains are measured at all the nodes and Gauss points, the measurement data contain 266 variables in total: (1) displacements, u: from the 1st to 74th variables, (2) stresses, σ: from the 75th to the 170th variables and (3) strains, ϵ: from the 171st to 266th variables.

- Displacement, u has 74 variables since there are 37 nodes and each node involves 2 variables: displacements in x-direction, u_x and y-direction, u_y. u_x always come before u_y for every node.
- Stress, σ has 94 variables since there are 32 Gauss points and each Gauss point involves 3 variables: normal stress in the x-direction, σ_x and y-direction, σ_y. Shear stress σ_{xy}.
- Strain, ϵ has 94 variables since there are 32 Gauss points and each Gauss point involves 3 variables: normal strains in the x-direction, ϵ_x, and y-direction, ϵ_y. Shear strain ϵ_{xy}.

The measurements were generated 100 times using random E values in the range ($10^5 - 10^7$ Pa). Therefore, the size of the measurement and target matrices is 100 by 1 and 100 by 266, respectively (see Figure 2.2).

The methodologies of the development of the proxies for the inverse problems are illustrated in Figure 2.3. The processes started by creating a model of the experiment such as the cantilever beam in FEM (Figure 2.1). By manipulating the target variables, data are generated that contain the measurements. The number of data points is controlled by the number of elements in the finite element meshes to discretise the structure as well as the accuracy of the simulation models. Next, a number of theoretically motivated proxies are tested. Since the proxies should be capable of finding few sensitive sensors from large numbers of sensors, it is considered as a classical variable reduction process combined with sensitivity analysis. In this process, the proxies dictate the criteria on which sensors are considered appropriate for the inverse problem results. It is critical to note the advantages of SPO using a model. Whereas searching for sensors using data would only explain the correlations between the measurement variables and the target variables, searching for sensors using models can explain the causalities between them since each of the target variables can be independently manipulated.

The identified sensors are placed on the models and virtual measurements are obtained. By performing inverse problems with the sensors, the target variables are recovered and they are compared to the actual target variables that were randomly generated. It needs to be made clear that the training data which are used to find sensors are different from the test data which are for testing the performances of the proxies. For this reason, the E values that are close to the limits are less likely to be trained as good as those in the middle of the range. This can lead to two distinct types of test datasets: (1) creating datasets populated

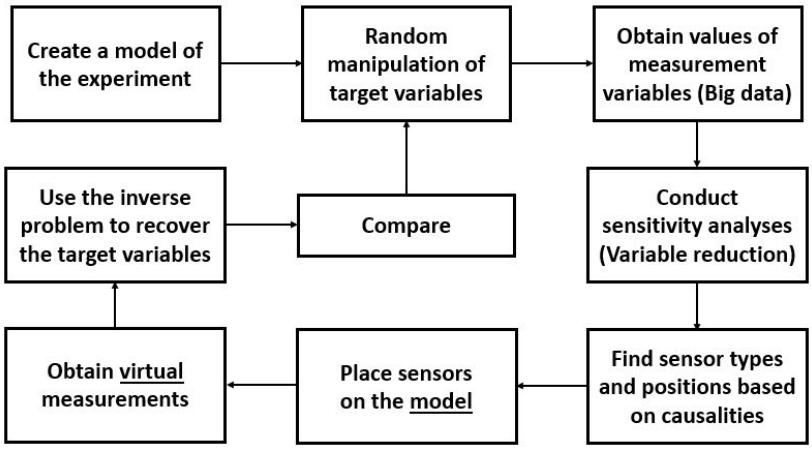


Figure 2.3: Methodologies for testing proposed proxies for the inverse problems.

mainly around the centres of the ranges using normal Gaussian distribution and (2) datasets populated evenly over the range using LHS. Both training and train datasets use LHS in this study since it tests the robustness of the technique. The main disadvantage of implementing the normally distributed target variables for training is that the values close to the boundaries rarely have chances for training. This also means the structures of the target variables are strongly dependent on the confidence one has about the ranges of the target variables in the training dataset.

2.2 Characteristics of inverse problems

In order to develop proxies to recover the essence of the measurement/target variable relationship of inverse problems, it is required to know the characteristics of the measurement/target variables for inverse problems. There are two main approaches that are considered in this paper: mode-based formulations and mode-free formulations. Mode-based formulation is as its name implies using numerical techniques which uses modes or eigenvalues while mode-free formulation does not require mode but it only uses basic statistical approaches and linear algebra techniques.

Since the aim of SPO is finding the optimal sensors which solve inverse problems, it is required to know what sensors are preferred for the inverse problems and how to distinguish the useful sensors from the unnecessary sensors. We consider the inverse problem errors as the optimisation objective function values and search for the sensors which would give the best recovery percentages (success rates) and minimum number of iterations to evaluate our proxies. This is done on a small problem as it is too computationally demanding for large problems that contain a large number of variables.

The cantilever beam problem with only one unknown target variable, E is therefore well suited. The number of sensors required to obtain the exact recovery is one. This avoids having to go through a large number of combinations to find the optimal sensor configuration. The number of possible combination which can be made choosing one sensor at a time is equal to the number of possible sensors ($C_1^{266} = 266$). Therefore, the inverse problem error based on optimising the inverse problem is tractable, requiring 266 inverse problems to be optimised. This process was done by minimising the error term, $\mathbb{E}$ which is given by

$$\mathbb{E} = \sum_{i=1}^{N_{variables}} \left(\frac{(X_{i,actual} - \bar{X}_i) - (X_i - \bar{X}_i)}{SD_i} \right)^2 = \sum_{i=1}^{N_{variables}} \left(\frac{X_{i,actual} - X_i}{SD_i} \right)^2, \tag{2.1}$$

where SD_i and $\bar{X}_i$ are the SD (standard deviation) and mean value of the measurement variables in training data. Note that both the training data and test data are generated in the

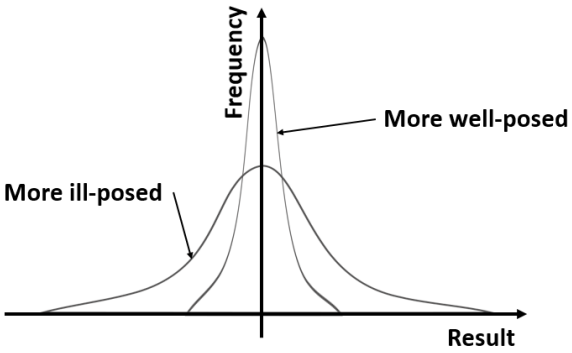


Figure 2.4: Illustration of well-posed and ill-posed problems.

same target variable ranges and in the results, the mean and SD values of them are almost identical). When finding the optimal sensors (training) the values of target variables are known. However, when testing the found sensors, it is assumed that the values of the target variables in the test data are unknown. Therefore, throughout the dissertation, the recovered target variables from the inverse problems are considered to be successful if the percentage errors between the actual values of target variables in the test data and the recovered values are less than 1%.

It must be emphasized that although having the heuristic cut-off value of 1% makes it convenient to indicate whether a sensor is appropriate or not for the inverse problems, the inverse problem results are not actually binary but relate to the ill-posedness of a problem. An ill-posed problem simply means it has a larger spread of answers than a well-posed problem (see Figure 2.4). Similarly, the variance of a good sensor would be less than a poor sensor. This is illustrated in the next section.

Note that the inverse problem uses the Z-score scaling as is often done in statistics. Z-score scaling is defined as

$$x_i = \frac{x_i - \bar{x_i}}{SD(x_i)}, \tag{2.2}$$

where $\bar{x_i}$ and $SD(x_i)$ are the mean and standard deviation of the ith design variable. Each variable is scaled with a different mean and SD to be in the equally distributed range as shown in Figure 2.5. The Z-score therefore normalises all variables to have the same mean and SD of zero and one, respectively. However, extremely small standard deviation may lead to normalisation of the variables which results in large Z-score value which makes a variable become "deterministic". Nonetheless, scaling is critical for minimisation of errors since there is a huge difference in the orders of magnitude among u, σ and ϵ. $X_{i,actual}$ is the actual measurement from the test data which X_i needs to be close to during the minimisation process. The error term, $\mathbb{E}$ is always a scalar function that can depend on a single variable (univariate) or multiple variables (multivariate) depending on the number of sensors to be placed. Since the cantilever beam problem only has one target variable and uses only one sensor, $\mathbb{E}$ in this case is an univariate function. The minimisation of the $\mathbb{E}$ term was done using Quasi-Newton [38, 44] and Nelder-Mead algorithms [45, 46] with maximum function evaluations and number of iterations of 5000. The initial guess was set to be the logarithmic mean of the range (i.e. 10^6 for $E = 10^5$ - 10^7 Pa) and this equally applies to all the inverse problems presented in this study.

2.2.1 Inverse problems with Quasi-Newton algorithm

Quasi-Newton algorithm [38, 44] generally requires less number of iterations compared to Nelder-Mead since it uses approximated Jacobian matrix whereas Nelder-Mead only searches for minima or maxima of functions using function values. The recovery success rates and numbers of iterations required for various sensors are shown in Figure 2.6(a). Each sensor has various numbers of iterations and recovery rates. Note that the recovery rates are measured

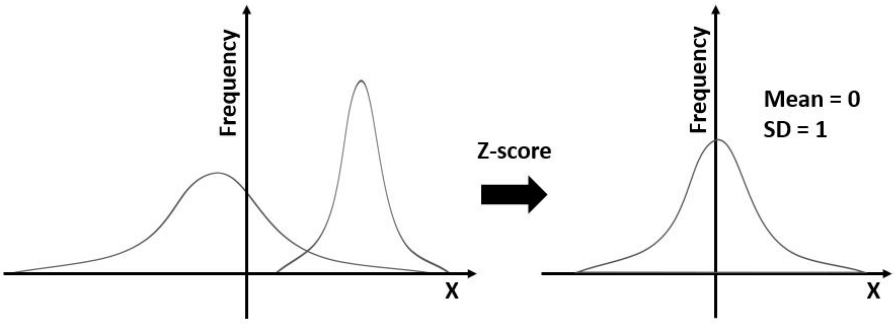


Figure 2.5: Effect of Z-score on data.

out of 100, 100% being the best possible success rate. The dotted lines on the figure indicates the three distinct measurement classes: u, σ and ϵ, respectively. It is noted that the numbers of iterations remain the same independent of the sensor selected. In turn, the recovery rates vary significantly between selected sensors. Consider that the recovery rates of the three classes of sensors (u, σ and ϵ) are studied separately in detail.

The displacement variables, u show that they have generally high recovery rates. This means that u is definitely a function of E in this problem (see Figure 2.6(b)). Note that the first ten measurement variables represent the fixed nodes of the cantilever beam. Therefore, it is apparent that they are insensitive to any changes in E and are incapable of recovering the target variables. Since the load is applied downwards at the tip of the beam, u in the Y-direction (vertical direction) is more sensitive to E than u in the X-direction (horizontal direction). This is demonstrated in Figure 2.6(b) by observing that the displacement variables in the Y-direction is consistently sensitive all over the beam. On the other hand, the displacement variables in the X-direction on the neutral axis (sensor ID number: 13, 21, 29, etc.) offer poor results. This is due to the horizontal displacement on the neutral axis being effectively zero as there is no horizontal strain along the neutral axis. As the displacement is measured further away from the root, the recovery rate improves slightly, although there is no strain. Some axial displacement occurs further from the tip although the strain remains zero. This clearly shows the role sensor placement plays but requires the problem to be well understood. As mentioned in the previous section, success rate is defined for properly converged solutions. Figure 2.7 shows the histogram of absolute percentage errors obtained from the sensors u_{13} and u_{39} (see Figure 2.6(b)) which are horizontal displacements at the root on the neutral axis and at the middle of the beam below the neutral axis, respectively. It is evident that the performance of u_{13} is much worse than u_{39} allowing for a more well-posed problem when u_{39} is considered.

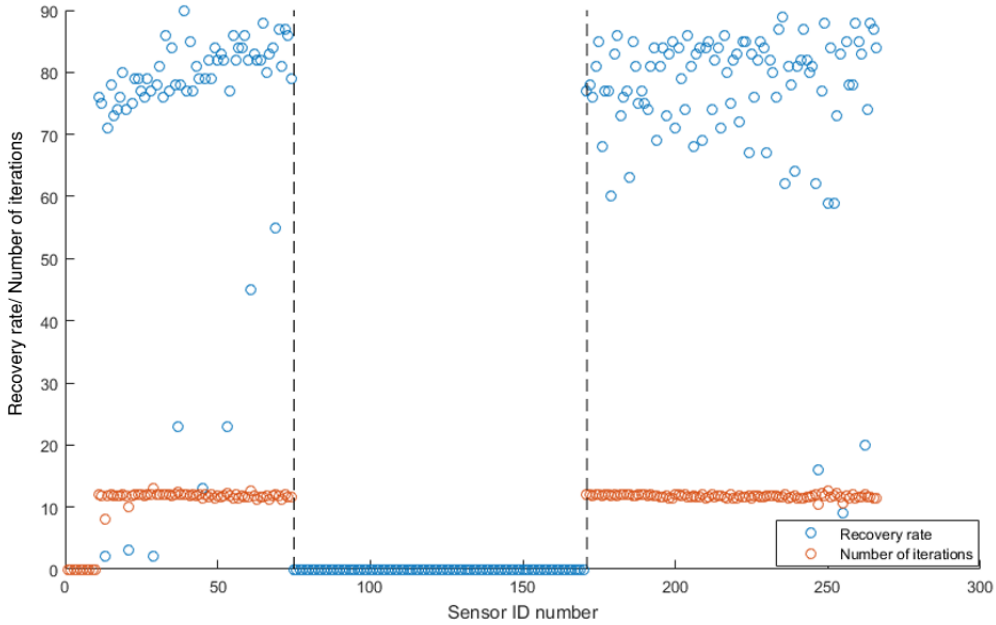
The stress variables, σ_{xx} are shown in Figure 2.8(a). It is evident that all the σ_{xx} sensors did not manage to recover any E values through the inverse problems. This was expected since σ_{xx} is independent of the material as no prescribed displacements are applied to this statically determinate problem. Equilibrium is resolved purely from a macroscopic force balance without the need for a constitutive relation. In essence, the primary stress in bending σ_{xx} is given by

$$\sigma_{xx} = \frac{My}{I}, \tag{2.3}$$

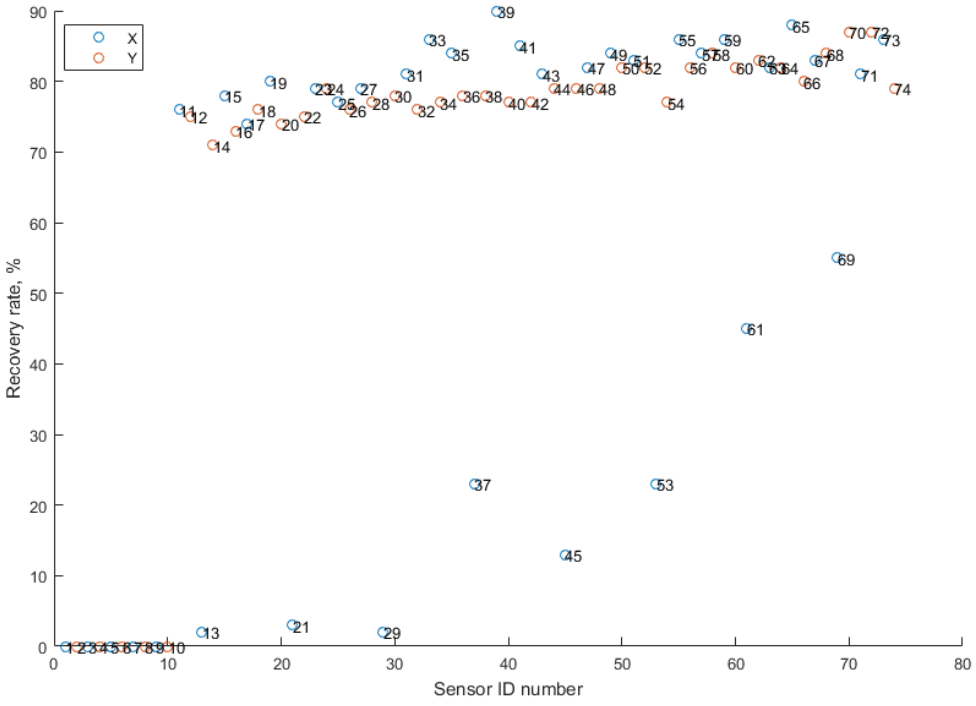
where y, M and I are the perpendicular distance to the neutral axis, the moment around the neutral axis and the moment of inertia. Therefore, the stress remains constant as the only manipulating variable is E. This means that σ sensors are not ideal for the normal cantilever beam inverse problem. Similarly, shear stress, σ_{xy} given by

$$\sigma_{xy} = \frac{VQ}{It}, \tag{2.4}$$

can also be neglected since shear force V is identical along the entire length of the beam. Q,



(a) Recovery rates and numbers of iterations for various sensors



(b) Recovery rates of displacement variables

Figure 2.6: Inverse problem results using Quasi-Newton algorithm; (a) recovery rates and numbers of iterations for various sensors and (b) recovery rates of displacement variables.

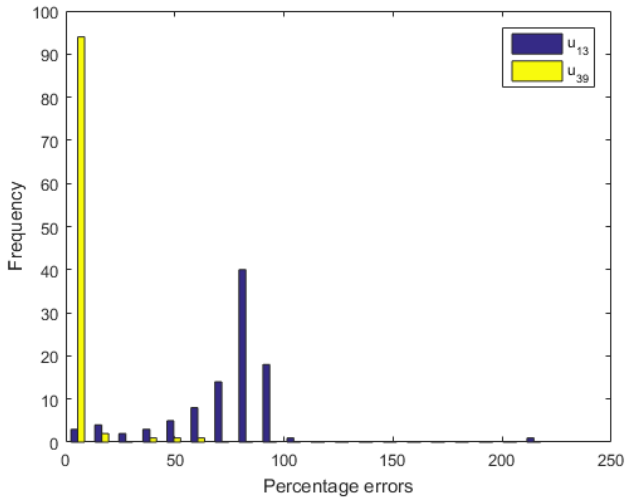


Figure 2.7: Variance of results of a good sensor and a poor sensor.

I and t denote the first moment of area, the moment of inertia and the width of the cross section where σ_{xy} is computed.

The primary strain, ϵ_{xx} is shown in Figure 2.8(b). It is the first derivative of displacement

$$\epsilon_{xx} = \frac{\delta u_x}{\delta x}. \tag{2.5}$$

It depends on E since

$$\epsilon = \frac{E}{\sigma}. \tag{2.6}$$

Strains are inversely proportional to Young's modulus. Shear strain, ϵ_{xy} is expressed as

$$\epsilon_{xy} = \frac{\tau}{2G}, \tag{2.7}$$

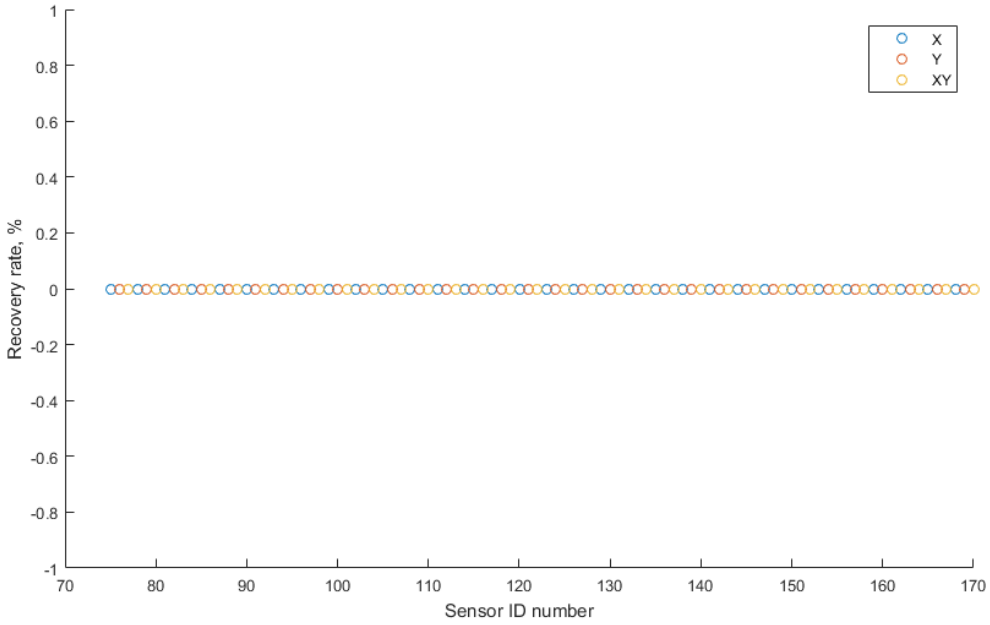
where shear modulus, G is related to Young's modulus, E through the Poisson's ratio, ν as given by

$$E = 2G(1 + \nu). \tag{2.8}$$

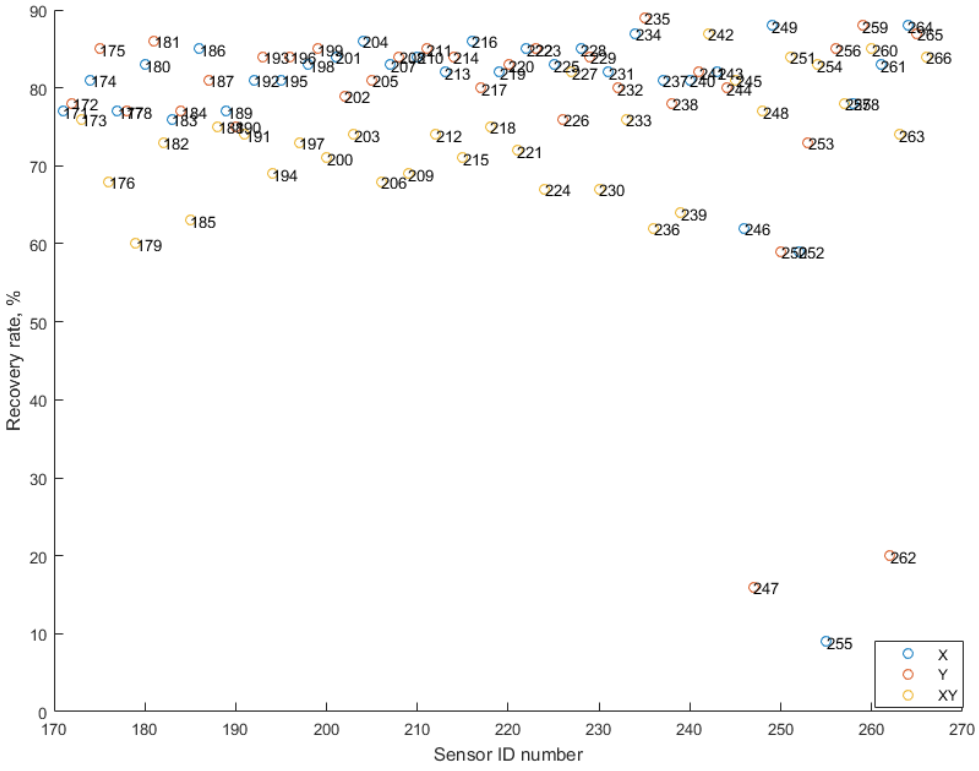
As depicted in Figure 2.8(b), it performs similarly to the displacement variables in characterising E. Both ϵ_{xx} and ϵ_{yy} performed equally well, while ϵ_{xy} is shown to be less robust. Note that the ϵ_{247}, ϵ_{255} and ϵ_{262} perform the worst as depicted in Figure 2.8(b). Their spatial location is depicted in Figure 2.9(a) and they have exceptionally low variance as depicted in Figure 2.9(b). These three strain sensors are all located close to the tip of the beam as depicted in Figure 2.9(a). ϵ_{xx} is a minimum due to the linearly varying nature of the bending moment. Note from Figure 2.9(b) that there is a strain sensor with its variances between ϵ_{255} and ϵ_{262}. However, its recovery rate is much higher than ϵ_{255} and ϵ_{262}. This demonstrates that even though a large variance in measurements is an indication of promising sensors, it is not the only driving factor. Although the variance of the sensor in between ϵ_{255} and ϵ_{262} was lower than the variance of ϵ_{262}, the sensor might have specific information that ϵ_{262} could not explain in terms of ϵ_{yy}.

2.2.2 Inverse problems with Nelder-Mead algorithm

Nelder-Mead [45, 46] is an unconstrained optimisation algorithm that does not use any gradient information and only optimises on function evaluations. Therefore, although Nelder-Mead often requires larger numbers of iterations, it has a higher chance to find the global optimum solutions in highly nonlinear problems compared to any gradient methods. Thus, the numbers of iterations and recovery rates of the sensors using Nelder-Mead algorithm are shown

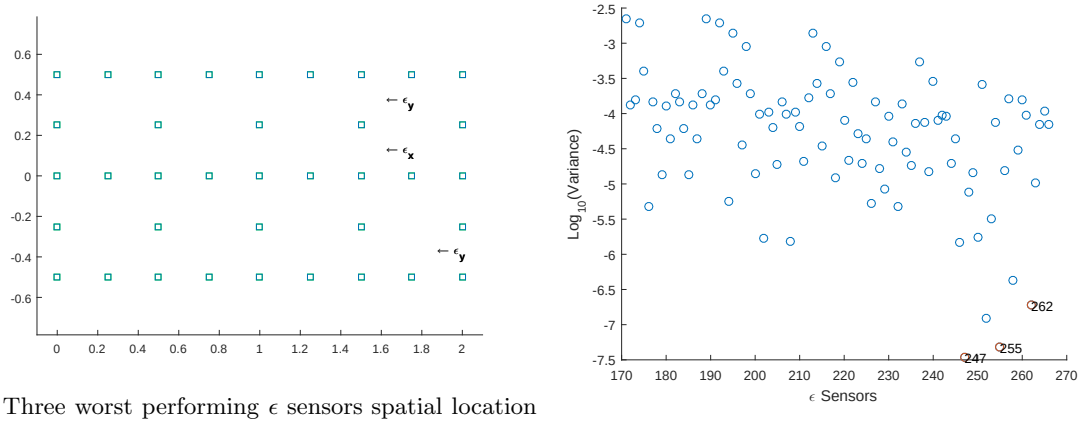


(a) Recovery rates of stress variables



(b) Recovery rates of strain variables

Figure 2.8: Inverse problem results using Quasi-Newton algorithm; (a) the recovery rates of stress variables and (b) strains variables.



(a) Three worst performing ϵ sensors spatial location on the beam

(b) $log_{10}(Variance)$ of ϵ sensors

Figure 2.9: (a) The spatial locations and (b) $log_{10}(Variance)$ of the three worst performing ϵ sensors.

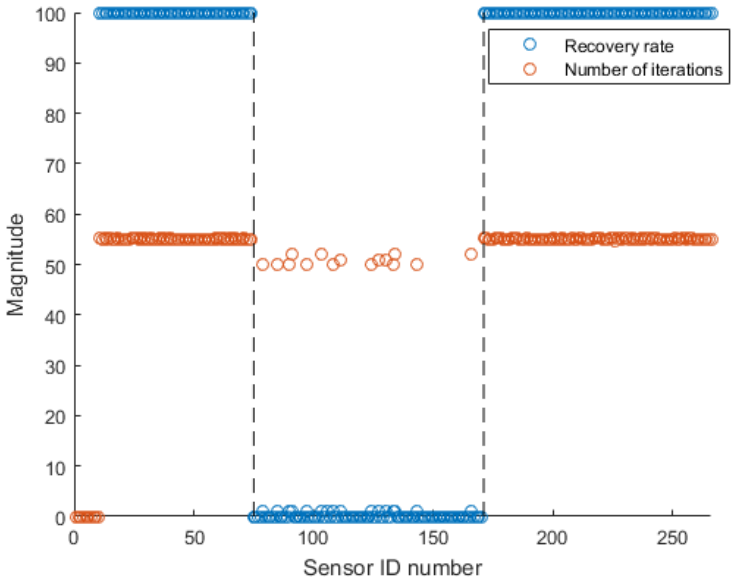


Figure 2.10: Recovery rates and numbers of iterations for various sensors.

in Figure 2.10 and the recovery rates of all the u and ϵ sensors are consistently 100% except for the σ sensors which are not related to E. Since it is difficult to see the difference in recovery rates and number of iterations due to the robustness of the optimisation algorithm, the complexity of the problem needs to be increased.

In order to ramp up the complexity of the beam problem, the number of target variables are increased. Note that the number of sensors should be equal to or more than the number of the target variables (FEM input variables) to accurately recover results. It becomes computationally intractable to search through all possible combinations of sensors when increasing the number of the target variables. Figure 2.11 illustrates how quickly the number of combinations of sensors grows with increase in the numbers of required sensors, k in C_k^n where n is the number of all the sensors. Note that the number of combinations is on a log_{10} scale. It is apparent that the growth relationship is exponential which makes the computation of large numbers of inverse problem error functions intractable for a large number of sensors. Decreasing the total number of sensors n or the number of sensors in each combination k can effectively decrease the number of possible combinations.

The more complex problem with 74 sensors is shown in Figure 2.12. This beam has

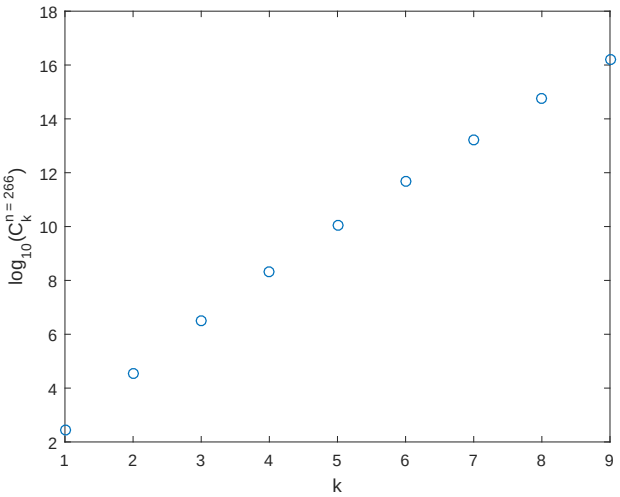


Figure 2.11: Growth in the number of possible combinations with increase in the number of sensors.

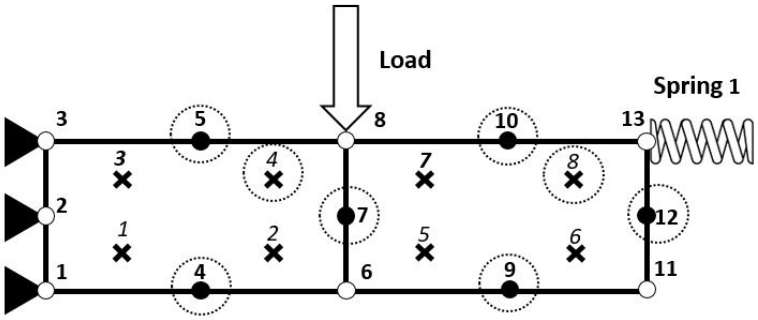


Figure 2.12: Beam bending problem with an extra spring to be identified and the less total number of sensors.

13 nodes and 8 Gauss points which offers 26 displacement, 24 stress and 24 strain sensors. A spring with varying stiffness of $k_1 = 1 - 100N/m$ is attached to the 13th node in the horizontal direction. The range of Young's modulus, E is increased to $10^7 - 10^{11}Pa$. As E increases the effect of the weak spring k_1 becomes trivial since the displacements of the beam are highly dependent on the stiffness of the beam and not much on the spring stiffness. The load is applied to the midpoint of the beam. By including the spring stiffness to be recovered, the inverse problem is now required to recover two variables, namely E and k_1. To keep the problem tractable when solving the combinatorial inverse problems for all the possible combinations with two sensors, the total number of sensors, n were reduced from 266 to 74 by doubling the size of meshes and got further reduced to 24 by only selecting specific nodes. The selected nodes and Gauss points are indicated in dotted circles. Therefore, sensor numbers 1 to 12 measure displacement u, from 13 to 18 measure stress σ and lastly, from 19 to 24 measure strain ϵ. The recovery success rates for this problem using Nelder-Mead are graphically shown in Figure 2.13. Since the success rate is independent of the sensor order, the graph is symmetric. Only the top half of the results are presented. Note that when both sensors are stress sensors to give a sensor pair $< \sigma, \sigma >$, the success rate is zero since the spring stiffness is too low to have any significant influence on the σ sensors.

As expected, when two of the same sensors are chosen, the amount of information is not enough to recover any target variables. Note that when $< \sigma, u/\epsilon >$ sensor pairs are chosen,

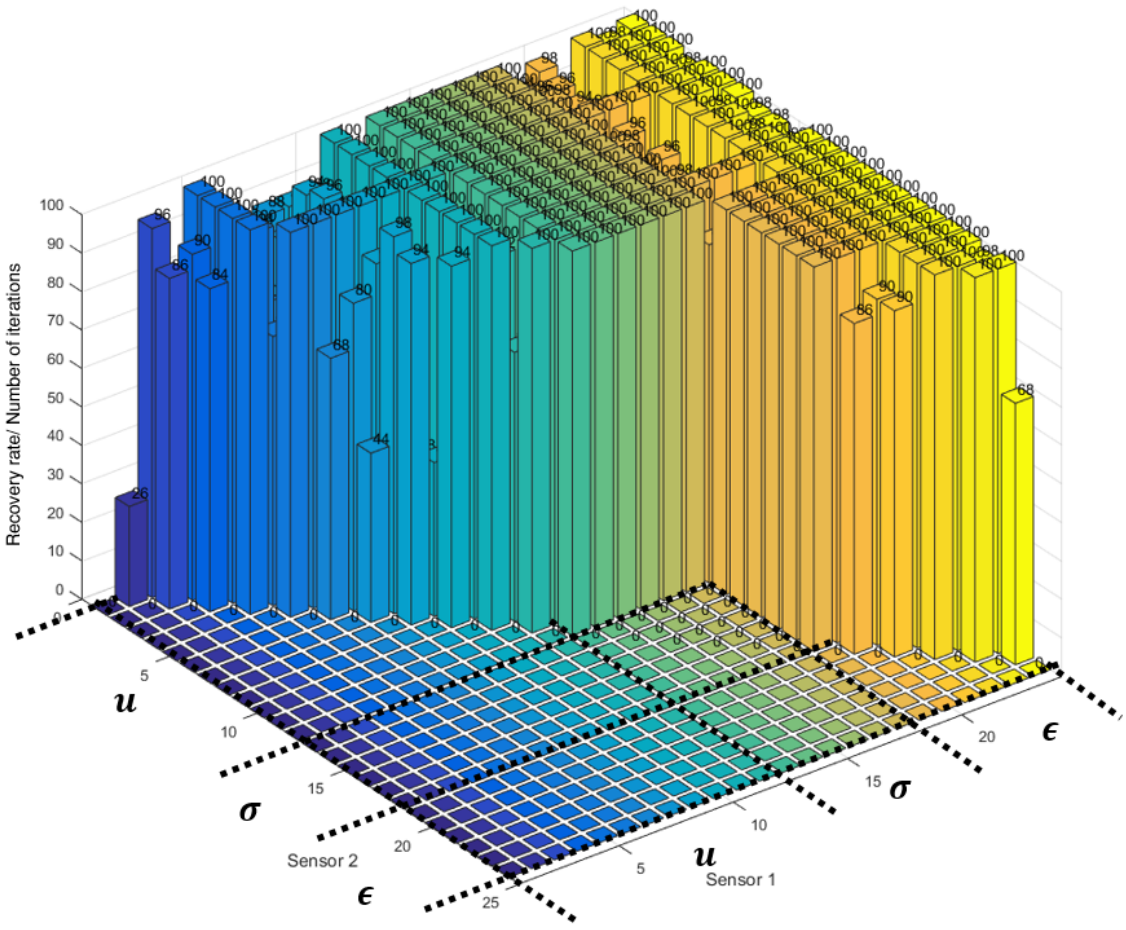


Figure 2.13: Recovery success rates for a more complex problem.

they always perform exceptionally well. However, when $< \sigma, \sigma >$ sensor pairs are chosen, zero recovery percentages are obtained. This is indicative that σ sensors contain more or less identical information. Yet, the single piece of information that σ sensors measure is valuable which complements the information measured by u and ϵ. Thus, two displacement or strain sensors without a stress sensor give poor results. The existence of the spring makes the problem statically indeterminate. The implication is that the stress response is dependent on the stiffness of the beam (i.e. E). Hence, the stress sensor is contributing towards the problem. Note that u measured close to the root of the beam and ϵ measured close to the tip of the beam perform much worse as they are less sensitive to E. For this problem, the Quasi-Newton optimisation approach was unable to recover any target variables. It is recommended to use a zeroth-order optimisation algorithm such as Nelder-Mead for a low dimensional inverse problem such as this.

2.3 Development of mode-based formulation

In the previous sections, the two cantilever beam examples were presented and the characteristics of the inverse problems were investigated. The aim is now to propose a number of proxy objective functions that are effective in replacing the actual objective function to minimise the inverse problem errors. In this section, the development of mode-based formulations is discussed. The two numerical techniques used on which the proposed mode-based techniques rely are partial least squares (PLS) [35, 47] and singular value decomposition (SVD) [21, 20].

SPO requires two types of datasets, namely the measurement variables X (u, σ and ϵ) and the target variables Y (E and k_1) which correspond to the sensors and information to be inferred, respectively. PLS regression (PLSR) is a powerful technique which has integrated features from both principal component regression (PCR) and multi-linear regression [35]. Although PCR and PLSR reduce the number of variables by extracting sets of highly correlated variables (mode) to a set of measurement variables, the difference between PLSR and PCR is that PCR extracts the principal components independently from both sets of target and measurement variables to construct directions, while PLSR extracts the principal components from the covariance matrix of X and Y to construct directions [49]. Both PCR and PLSR deal with the multicollinearity within the X dataset due to mode extraction following principal component analysis. Hence, linearly dependent information is handled appropriately and need be of no concern as we develop sensor placement proxies.

SVD is a well-known numerical method often used in statistics and signal processing. It decomposes a $m \times n$ matrix into a $m \times n$ left eigenvector matrix U, $m \times n$ rectangular diagonal matrix S and a $n \times n$ right eigenvectors matrix V. The vectors generated from SVD are orthogonal to each other. The information along a direction is independent of the other directions. Although multicollinearity is addressed implying independent sensor, the numbers of modes need to be chosen carefully. Too few modes would not extract enough information from the datasets and too many extract noise from the data. Mode-based formulations would require a step for selecting the number of modes that may be computationally demanding. In this section, only mode-based formulations developed from PLS are considered, although both PLS and SVD are considered at a later stage.

2.3.1 Simple cantilever beam

We now revisit the inverse problems of the simple cantilever beam example depicted in Figure 2.1, which we solved using Nelder-Mead to obtain the best possible sensor placement based on the ability of the optimisation algorithm to solve the inverse problem. The aim is to consider proxy objectives that are computationally efficient but still capture essentially the same information as the computational demanding actual objective that requires the inverse problem to be solved. Figure 2.14(a) shows the recovery success rates of the sensors in descending order when solving the inverse problem. Figures 2.14(b)-(f) depict potential proxies, standard deviation, correlation coefficient [48] between X and Y, norm of Y residual

and variance explained in the same order. This allows for convenient comparison between the proxies and the desired performance. Recall that the only FEM input variable for this problem was E and it did not affect σ. Therefore, the first 158 sensors with non-zero success rates are the u and ϵ sensors. The rest are either the u at the fixed nodes or the σ sensors.

It was noticed earlier that standard deviation and variance may be appropriate indicators to decide which sensors are suitable to solve an inverse problem since a sensor whose reading does not vary as the system changes is never a good predictor. Note from Figure 2.14(b) that the promising sensors have relatively large standard deviation. However, the prominent mismatches between the results and the recovery rates depicted in Figure 2.14(a) are noticed which means standard deviation alone is insufficient to be a promising proxy.

Figure 2.14(c) shows the correlation coefficients between the sensors and the FEM input variable, E. The correlation coefficient, r [48] is given by

$$r = \frac{\sum_{i=1}^{N}((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}, \tag{2.9}$$

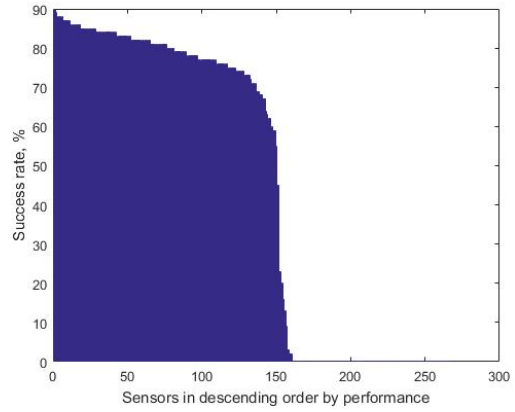
where N is the number of observations and the x and y are the sensor measurement variables and target variables (FEM input variables), respectively. The mean of a variable is denoted by placing a bar above the variable. It is evident from Figures 2.14(a) and (c) that sensors that succeeded to recover the target variables once or multiple times have an equally high correlation. The ten fixed nodes have no correlation to E and lastly, σ has an only slightly higher correlation but it is much lower than the correlation of u and ϵ. Note that since the correlation coefficient can vary from -1 to 1, their absolute values were plotted. The rationale for only considering absolute values of the correlation coefficient is that the variation of the measurement variables with respect to the manipulations of the target variable matters and not whether they are directly or inversely related.

Figure 2.14(d) shows the norm of the regression error $Y_{residuals}$ computed using PLSR [35], specifically the SIMPLS algorithm [47]. Lower error norms indicate better predictions. Only one mode is available for selection, since only one target variable is predicted from a single sensor. SIMPLS maximises the covariance between the measurement and target variables. The σ values which are constant as E varies result in higher errors than the varying u and ϵ sensors. Note that SIMPLS uses Z-score values for both measurement variables and target variables. The scaling of data becomes essential to avoid biased measuring of errors. In fact, there are numerous ways to scale data which emphasize different aspects of the data. A detailed discussion on various scaling method is presented in the next chapter.

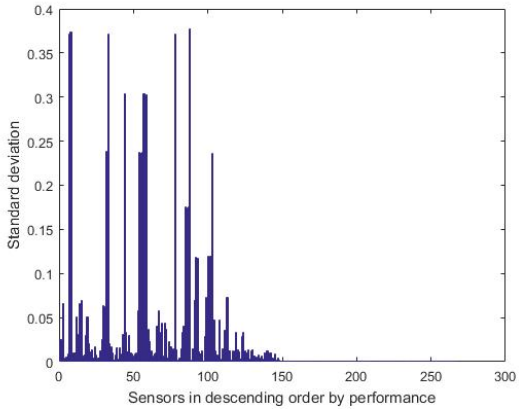
The variance explained in the target variables Y is shown in Figure 2.14(e). It is interesting to note that the results look almost identical to the result of the correlation coefficients. The only difference is that the variance explained in Y distinguishes the σ sensors better from the u and ϵ sensors than what the correlation coefficient achieves. The variance explained in X, the sensor measurements, are shown in Figure 2.14(f). The variance explained in X is expected to give poor performance since this checks how much of the information in X is used when mapping to the Y domain and SIMPLS constructs X scores to decompose both X and Y. Hence, this allows us to access the information captured in the measurement data that all the sensors explain with indistinctly highest possible function value, 1. Note that the variance explained is measured by dividing the amount of information that has already been explained by the total amount of information. As expected, this shows that comparing the variance explained in X is never a good idea for ranking the sensors according to the recovery success rates.

2.3.2 Cantilever beam with a spring

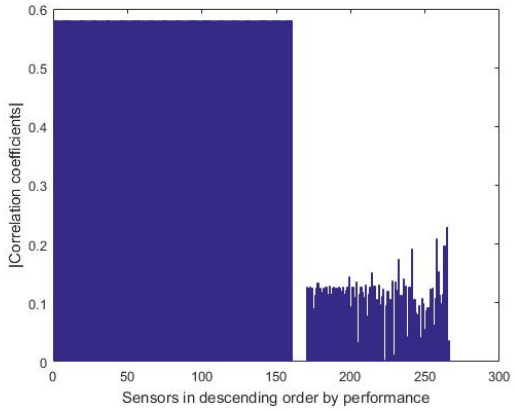
Recall that, in Section 2.2.1, it was demonstrated that the Quasi-Newton method in Matlab version was unable to solve the inverse problems of the cantilever beam with a spring. The cantilever beam with a spring (Figure 2.12) is a more ill-posed problem than the normal



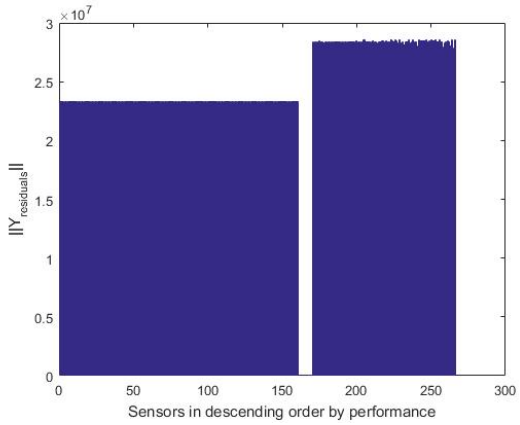
(a) Recovery rates



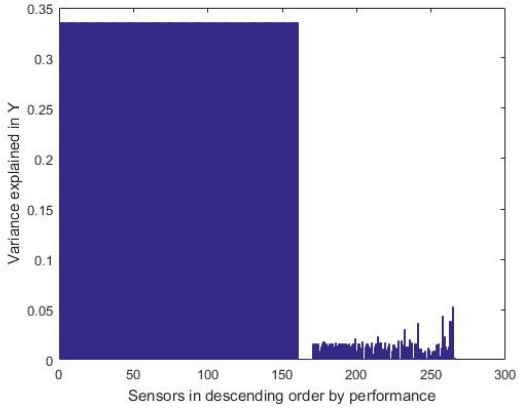
(b) Standard deviation



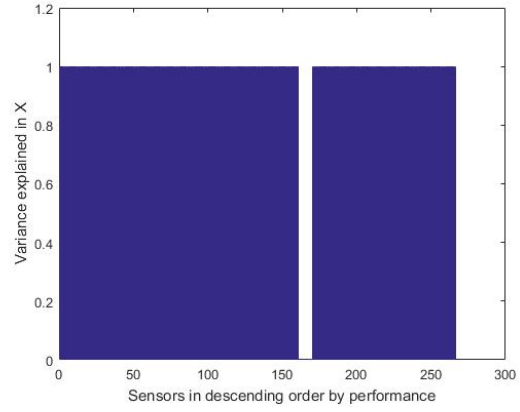
(c) Correlation coefficient



(d) Norm of $Y_{residuals}$



(e) Variance explained in Y



(f) Variance explained in X

Figure 2.14: Proxies of the inverse problems using the simple cantilever beam; (a) recovery rates, (b) standard deviation, (c) correlation coefficient, (d) norm of $Y_{residuals}$, (e) variance explained in Y and (f) in X.

cantilever beam without the spring, which could only be solved with Nelder-Mead. Since we have now two target variables for which we allow at least two sensors to be placed, we have up to two modes that can be selected.

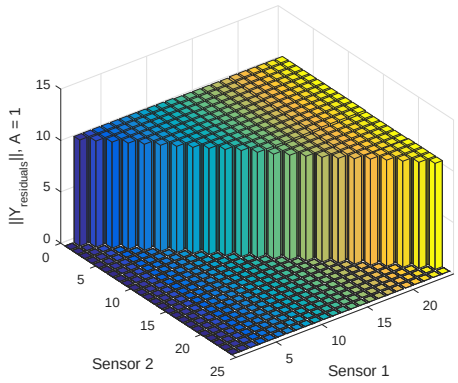
To establish the relevant proxies of the recovery rates (Figure 2.13), a number of numerical analyses were done as shown in Figure 2.15. Figures 2.15(a) and (b) show the norms of regression error, $Y_{residuals}$ from SIMPLS regressions for all possible combinations of the sensors using one or two modes, respectively. It is important to note that lower values of $Y_{residuals}$ imply the better sensor as it is the only proxy that forms part of a minimisation problem. For one mode, there is no significant difference between the sensors, although the σ sensors numbered 13-18 are slightly lower. This is enhanced when two modes are considered. Of concern is that $< \sigma, \sigma >$ which we know are not reliable to solve the inverse problems have the same associated function value as the other sensors that we know were reliable.

Figures 2.15(c) and (d) present the variance explained in the target variables Y using SIMPLS for one and two modes. Note the variance explained is a maximisation problem which means it can be directly compared with the recovery rate graphs considered earlier. As intuitively expected, the magnitude of the variance explained for one mode is generally small compared to two modes. Surprisingly a σ sensor is always preferred over other sensors and surprisingly enough, $< \sigma, \sigma >$ sensor pair explains most of the variables. For two modes, both $< \sigma, u >$ and $< \sigma, \epsilon >$ rate significantly higher than $< \sigma, \sigma >$ sensor pairs, which we know perform poorly on the inverse problem. These results demonstrate that variance explained in Y might be a useful proxy but also highlight the importance of selecting the number of modes appropriately.

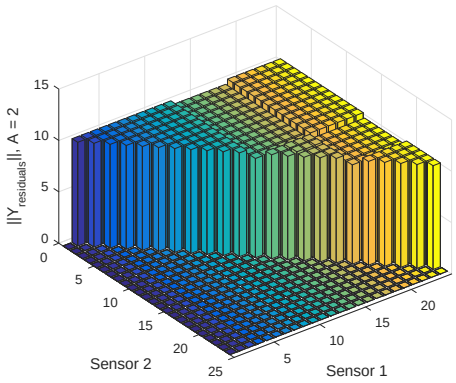
The norms of correlation coefficients between measurement variables and target variables are displayed in Figure 2.15(e). These proxies are far from the actual recovery rates since $< \sigma, \sigma >$ gives the best correlation to the target variables. This clearly shows that correlation between the measurement and target variables is not the right proxy for the inverse problems. This is in particular against intuition that correlation should be a good indicator. Lastly, the pure sum of standard deviation of variables in X_c, the chosen sensors, are shown in Figure 2.15(f). Note that all the previous methods used transformed Z-score measurement but in this case, the untransformed, original variables were used for the analyses. The standard deviations of the σ sensors are higher than the other sensors in particular when $< \sigma, \sigma >$ sensor pairs are considered due to the fact that σ has larger magnitudes. The results show that both correlation coefficient and standard deviation are not suitable proxies when considered in isolation.

It must be emphasized that the difference between the ranges of each target variables is a few order of magnitudes and they are manipulated simultaneously. Note that the best combinations of sensors may vary depending on the chosen sensor ranges (Figure 2.16(a)). The challenge is to find a set of sensors that perform well over the defined ranges. If there is a strong confidence in which ranges the solution lies, the size of the training ranges can be reduced to help the proxies to find the specialised sets of sensors for those narrowed ranges. Figure 2.16(b) illustrates that each selected sensor has its own explanatory domain of variables, E and k, that the inverse problem aims to solve. It is important to note that the aim is to maximise the total coverage by minimising the overlapping information between the sensors. For instance, sensors 1 and 2 will struggle to identify k but should recover E with ease. However, replacing sensor 2 with sensor 3 would allow for both E and k to be recovered.

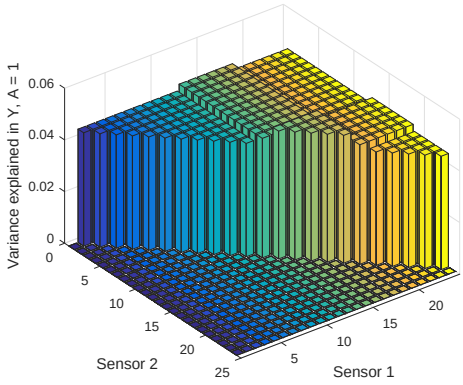
Figure 2.17 illustrates the norms of $Y_{residuals}$ and variance explained in Y and when k is fixed, while Figure 2.18 depicts their response when E is fixed and k varied. Values are fixed around their mid-values in a range to ensure sufficient data to reconstruct the responses. By varying one variable at a time, it is possible to find appropriate sensors for each target variable. Figure 2.17 shows that both the norm of $Y_{residuals}$ and variance explained are equivalent for one mode since all sensor combinations have the same function values. However, for two modes, the $\|Y_{residuals}\|$ predicts worse performance for $< \sigma, u >$ and $< \sigma, \epsilon >$, which is



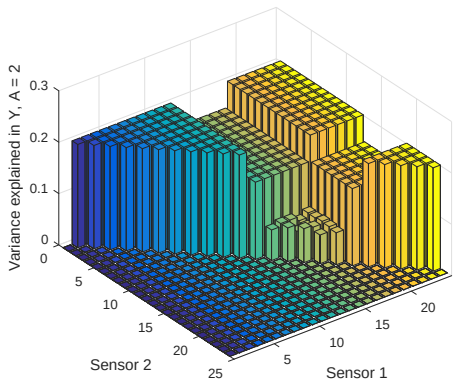
(a) Norm of $Y_{residuals}$ with $A = 1$



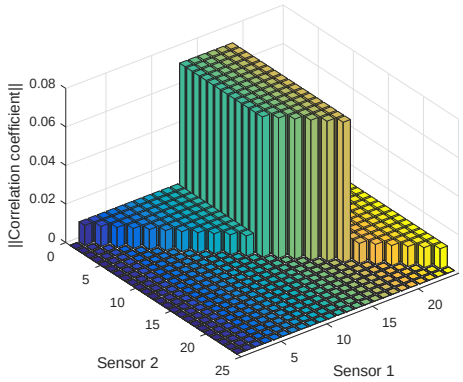
(b) Norm of $Y_{residuals}$ with $A = 2$



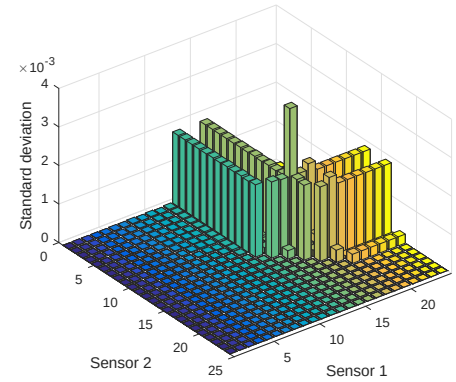
(c) Variance explained in Y with $A = 1$



(d) Variance explained in Y with $A = 2$

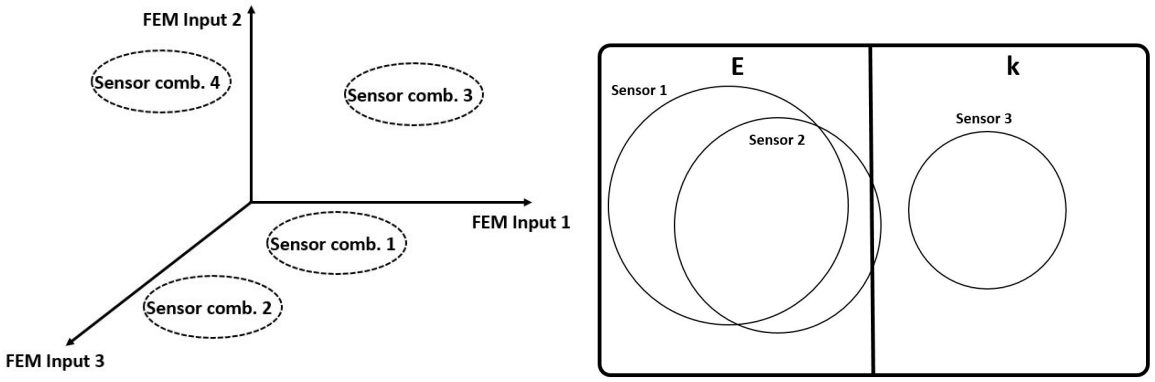


(e) Norm of correlation coefficient



(f) Standard deviation of X

Figure 2.15: Various proxy responses for combinations of sensors; (a) norm of $Y_{residuals}$ with $A = 1$ and (b) $A = 2$, (c) variance explained in Y with $A = 1$ and (d) $A = 2$ (e) norm of correlation coefficient and (f) standard deviation of X.



(a) The best combinations of sensors for different input domains

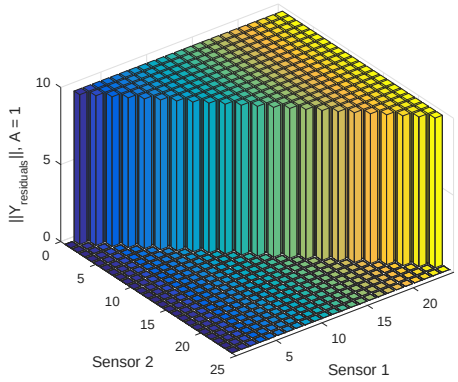
(b) Domains of measurement variables that sensors explain

Figure 2.16: The relationships between the FEM input variables and the sensors; (a) the best combinations of sensors for different input domains and (b) domains of measurement variables that sensors explain.

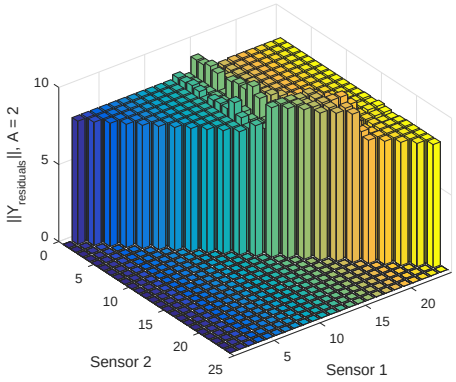
contradictory to the actual performance on the inverse problem recovery. On the other hand, many of the combinations in $< \sigma, u >$ and $< \sigma, \epsilon >$ show higher variance explained in Y (Figure 2.17(d)) which is better correlated with the actual recovery of the inverse problem.

Similarly, the function values of mode-based formulations when only k is manipulated are shown in Figure 2.18. Remember that whereas the parameter E varies over three orders of magnitude, k only varies over two at much lower magnitudes. The variation of k is less than E with $\|Y_{residuals}\|$ significantly lower when the number of modes, $A = 1$ and 2. For $A = 1$, the σ sensors are definitely not preferred. For $A = 2$, the ϵ sensors are less helpful for the recovery of k than the other sensors due to the ϵ measurement values and ranges being much smaller than the other sensors. The variance explained in k is shown to be always 100% for both $A = 1$ and 2. This implies that the small variance of k can be fully explained using any sensor pair. When the two target variables are manipulated at the same time, the sensor optimisation algorithm might initially focus more on a target variable with higher variance and later on lower variance variables should they improve the variance explained. Note that although the two target variables are independent, they might often be internally related to each other in FEM. Hence, the simultaneous manipulation of the two variable is critical to capture the behaviour of the system globally.

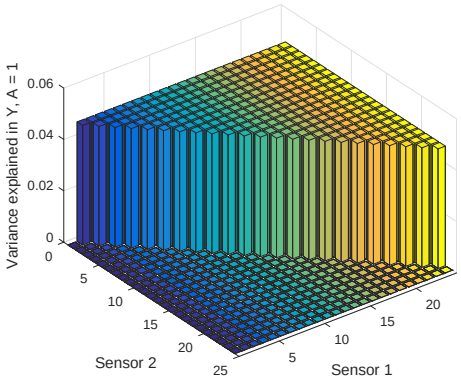
Additional optimisation objective functions and scaling methods will be presented in the next chapter. The investigations up to now were sufficient to declare that the mode-based formulations merit further investigations for potential proxies. However, there are a few disadvantages to using the mode-based formulations that might not have been clear up to now. Since mode-based formulations rely on eigenvectors and eigenvalues, the computational time required to compute these vectors can grow quickly as the number of target variables grows. Another issue is that different combinations of sensors might have different number of optimal modes A. Therefore, the optimal number of sensors need to be established adding to the computational burden. For example, should three sensors be required, up to three modes are possible candidates to be selected. The optimisation approaches need to consider three cases $A = 1$, 2 and 3 to establish the optimal number of modes. This process can be computationally expensive for a large number of target variables. Ultimately, we would like to revisit the fundamental premise on whether it is required to consider mode-based approaches and whether computationally efficient alternatives could be considered. The next section explores alternatives.



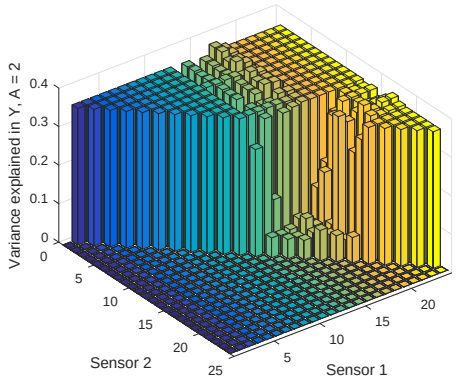
(a) Norm of $Y_{residuals}$ with $A = 1$



(b) Norm of $Y_{residuals}$ with $A = 2$

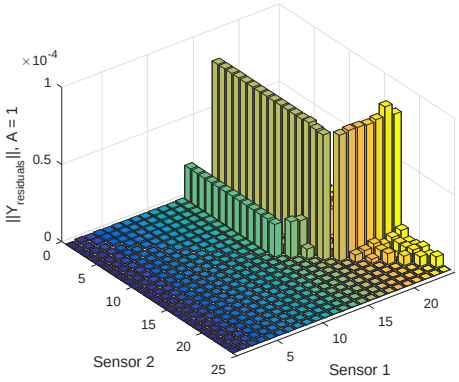


(c) Variance explained in Y with $A = 1$

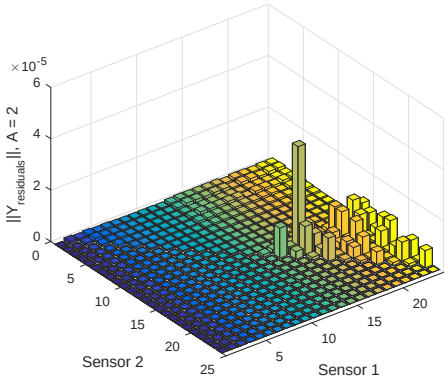


(d) Variance explained in Y with $A = 2$

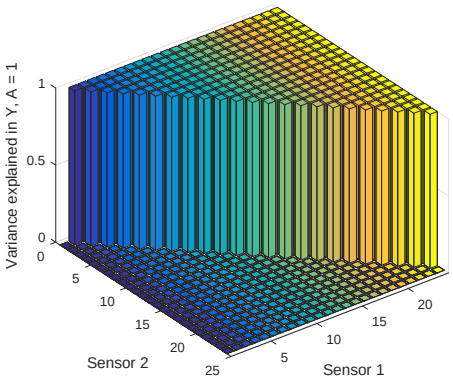
Figure 2.17: Various function values on the combinations of sensors when k is fixed; (a) norm of $Y_{residuals}$ with $A = 1$ and (b) $A = 2$, (c) variance explained in Y with $A = 1$ and (d) $A = 2$.



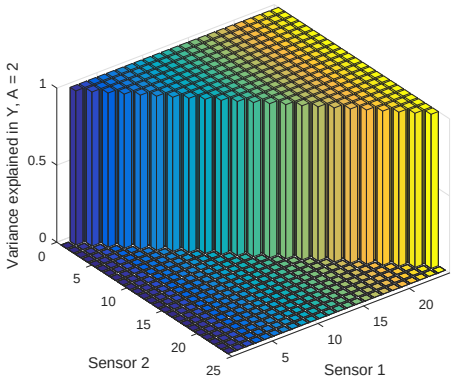
(a) Norm of $Y_{residuals}$ with $A = 1$



(b) Norm of $Y_{residuals}$ with $A = 2$



(c) Variance explained in Y with $A = 1$



(d) Variance explained in Y with $A = 2$

Figure 2.18: Various function values on the combinations of sensors when E is fixed; (a) norm of $Y_{residuals}$ with $A = 1$ and (b) $A = 2$, (c) variance explained in Y with $A = 1$ and (d) $A = 2$.

2.4 Development of mode-free formulations

The mode-based formulations can be robust with the right scaling methods and objective functions. However, it is challenging to reduce the time required for these methods. Here, we aim to reduce the computational cost by considering approaches that do not require modes to be computed. We will refer to these methods as mode-free formulations. The investigations conducted up to this point highlighted a number of central characteristics of inverse problems and the findings are as follow:

1. Informative relationship between the sensors and target variables.
2. Uniqueness of contents in each sensor.
3. High sensitivity of sensors.

In the following sections, these three criteria and the thought processes followed to incorporate and address them in a mode-free approach are discussed in detail.

2.4.1 Relationship between measurement and target variables

It is essential to have strong relationships between the sensors and the target variables. The two sets of variables, X and Y, projected onto each other explain their relationships. By constructing a Z-score scaled covariance matrix between the two datasets, the relationships between each measurement and target variables can be measured relatively. Hence, each row in the covariance matrix X^TY represents how well each measurement variable is correlated with the target variables and the uniqueness of their relationships is represented by uniqueness of the row entries. Hence

$$rank\left(\left(\frac{X_c-\bar{X}_c}{SD(X_c)}\right)^T\left(\frac{Y-\bar{Y}}{SD(Y)}\right)\left(\frac{Y-\bar{Y}}{SD(Y)}\right)^T\left(\frac{X_c-\bar{X}_c}{SD(X_c)}\right)\right) \tag{2.10}$$

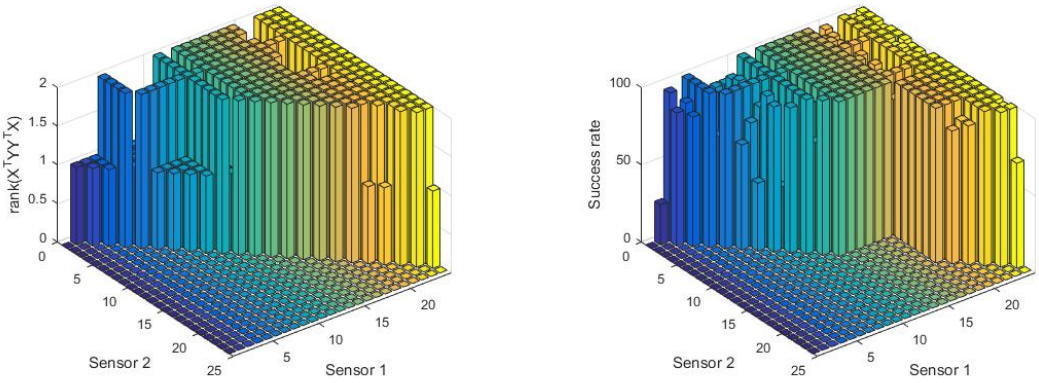
measures the rank of the covariance matrix between the two sets of sensor and target variables and this is an efficient way to ensure uniqueness of information. Note that both the measurement variables X and the target variables Y are scaled using the Z-score method so as to avoid having biased covariance matrices due to the differences in the orders of magnitudes. X_c denotes chosen X variables or sensors. Squaring the covariance matrix effectively polarises the correlation coefficients so that the difference can be emphasized when computing the rank. It is clear from Figure 2.19 that there is significant similarity between the rank of covariance matrices and the success rate of the inverse problems. However, the biggest issue of this measure alone is that the $<\sigma,\sigma>$ sensor pairs also have a maximum rank of 2 (the size of covariance matrix, X^TYY^TX in this case is 2 by 2). Note that there is no difference between the ranks of X^TYY^TX and Y^TXX^TY. Another concern is that the $<\sigma,u>$ and $<\sigma,\epsilon>$ groups are equal to the other sensor pairs that we know performed worse when solving the actual inverse problem in Figure 2.19(a). These two issues will be considered and resolved in the remaining two sections of this chapter.

2.4.2 Uniqueness of contents in each sensor

No matter how promising each one of the selected sensor is, if the sensors contain almost identical information, they do not effectively maximise the explanatory domain for the target variables. Hence, it is imperative that each sensor explains the target variables by reducing repeated information. Towards the aim consider

$$rank\left(\left(\frac{X_c-\bar{X}_c}{SD(X_c)}\right)^T\left(\frac{X_c-\bar{X}_c}{SD(X_c)}\right)\left(\frac{X_c-\bar{X}_c}{SD(X_c)}\right)^T\left(\frac{X_c-\bar{X}_c}{SD(X_c)}\right)\right), \tag{2.11}$$

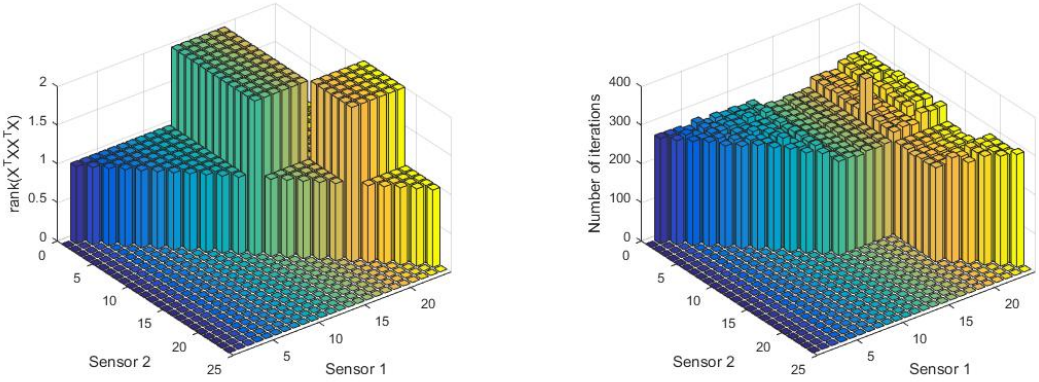
instead of considering the covariance matrix of the sensor variable; we consider the covariance projected onto itself. This amplifies higher rank estimates and diminishes lower rank



(a) Rank of covariance matrix

(b) Success rates from inverse problem

Figure 2.19: Comparison between the (a) ranks of covariance matrix and the (b) inverse problem success rates.



(a) Rank of chosen sensors

(b) Number of iterations from inverse problem

Figure 2.20: Comparison between the (a) ranks of sensors and the (b) inverse problem number of iterations.

estimates. In this way, it allows us to distinguish more clearly between the sensor pairs. It compares purely the collinearity of data since Z-score transformation normalises magnitude differences away. The $< \sigma, u >$ and $< \sigma, \epsilon >$ combinations are the only sensors to now have 100% recovery rates and Figure 2.20(b) shows that they require the least numbers of iterations. They are now consistent with the actual inverse problem and highlighted as the best sensor pairs. Equation 2.11 allows us to assess uniqueness of sensor information.

2.4.3 Sensitivity of sensors

Equation 2.10 distinguishes between sensors that correlates uniquely to the target variables and Equation 2.11 ensures that the correlations within selected sensors describe unique information when compared to the other selected sensors. At this point both objective functions are sufficient to create a proxy as shown in Figure 2.21 by the pointwise product of the two functions shown in Figures 2.19(a) and 2.20(a), respectively. The maximisation of this objective is sufficient to rank the $< \sigma, \epsilon >$ or $< \sigma, u >$ sensors the highest. However, note that the $< \sigma, \sigma >$ sensor pair which is supposed to be the most ineffective combination is ranked equally to some of the other regions that are known to be more effective albeit not the best. We now consider an objective function which can distinguish $< \sigma, \sigma >$ from the rest which was found to be the sensitivity of sensors.

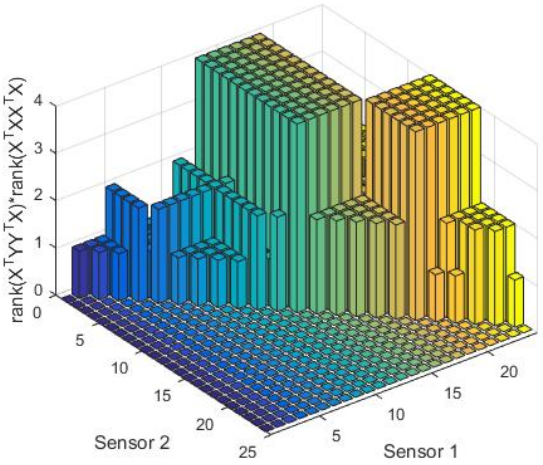


Figure 2.21: Multiplied ranks of both sensor matrix and covariance matrix.

The sensor sensitivity is the variance (or standard deviation) of a sensor

$$\prod_{i=1}^{N_{sensors}} var(x_{c,i}), \tag{2.12}$$

or the relative sensitivity measured using the following mean scaling

$$x_{scaled,i} = \frac{x_i - \bar{x}_i}{\bar{x}_i}. \tag{2.13}$$

In fact, the variances of the σ sensors are higher than those of the ϵ or u. Therefore, this can not be a good measure of sensitivity alone. Although the mean scaling gives a good indication of variation of data relative to its mean magnitudes, not all the σ sensors have relatively large mean values and particularly some are very close to zero which may cause numerical ill-conditioning.

Due to the biased variance measure caused by various magnitude ranges of the sensors, it is not ideal to combine the three objectives (Eqs. 2.10, 2.11 and 2.12). Hence, instead of treating them as unconstrained optimisation, we can solve the problem in an alternative way by only considering the sets of variables that have the maximum possible ranks for functions in Eqs. 2.10 and 2.11. In that case, the first two functions represent two constraint functions and Equation 2.12 becomes the only objective function. This allows for constrained optimisation of the three objectives. In fact, since only the sets of sensors which have the full rank for the two rank equations are considered, it is already guaranteed that all the sensors are unique and that the correlations between the sensors and target variables are also unique. Hence, additional optimisation using Equation 2.12 can only focus on maximising the variances of the sensors which increases the sensor sensitivity and the ratio of the information from the sensors to the measurement noise.

This ends the overview to both mode-based formulations and mode-free formulations. The following three chapters detail the numerical experiments on the mode-based formulations, followed by the numerical studies on mode-free formulations in Chapter 6.

Chapter 3

Mode-Based Formulations: Data Scaling and Objective Functions

This chapter presents six different data scaling methods and their roles. In addition, six objective functions based on SIMPLS and SVD mode extraction methods are introduced. To clearly demonstrate the effects of scaling, a new statically indeterminate beam problem is introduced first.

3.1 Numerical FEM: Experimental setup

It is time to introduce the main experimental setup which is used throughout the rest of the study. A statically indeterminate linear elastic FEM beam bending problem (see Figure 3.1) is considered. The beam consists of two eight-noded finite elements that is 0.5 m high, 2 m long and 0.5 m thick. The elements are integrated using reduced integration (i.e. 2×2 Gauss integration points). The 2D plane strain assumption was used with the Poisson's ratio, ν of 0.3. The Young's modulus, E of the beam and the spring stiffness values for the two springs (spring 1 and 2), k_1 and k_2 are the three target variables in this problem. This problem can be more ill-posed than the previous single spring problem shown in Figure 2.12. The target variables were varied randomly using LHS (Latin hypercube sampling) in various ranges as shown in Table 3.1. The reason for dividing the ranges of each variable into low and high is to allow us to dictate the posedness of the problem as will be discussed later. The left side of the beam is completely fixed and a point load of 1000 N is applied at the 8th node in the negative y-direction. The spring 1 and 2 are attached at the 13th node in the x-direction and 11th node in the y-direction, respectively.

The measurement data contains 74 variables in total: (1) displacements, u: from the 1st to 26th variables, (2) stresses, σ: from the 27th to the 50th variables and (3) strains, ϵ: from the 51st to 74th variables.

1. Displacement, u has 26 variables since there are 13 nodes and each node involves 2 variables: displacements in x-direction, u_x and y-direction, u_y. u_x always come before u_y for every node.

2. Stress, σ has 24 variables since there are 8 Gauss points for the two elements integrated using 2×2 reduced integration rule and each Gauss point involves 3 variables: normal

Range	Young's modulus, E [Pa]	Spring stiffness, k_1 and k_2 [N/m]
Low	10^5 - 10^7	1 - 10^2 and 1 - 10^2
High	10^7 - 10^9	10 - 10^3 and 10^3 - 10^5

Table 3.1: Chosen variables for different correlation coefficient tolerances from the FEM data (high E and high k)

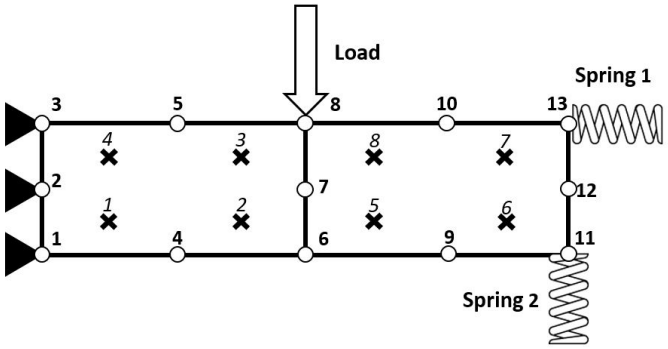


Figure 3.1: Diagram of the FEM bending example.

stress in x-direction, σ_x, y-direction, σ_y and shear stress along the xy-direction (shear stress), σ_{xy}.

3. Strain, ϵ has 24 variables since there are 8 Gauss points for the two elements integrated using 2×2 reduced integration rule and each Gauss point involves 3 variables: normal strains in x-direction, ϵ_x, y-direction, ϵ_y and shear strains along the xy-direction, ϵ_{xy}.

The sensor ID numbers for u, σ and ϵ are all shown in Figure 3.2. The measurements were taken for 100 independent LHS generated samples for varying E, k_1 and k_2. In other words, the sample size was 100. Therefore, the measurement data has a size of 100 by 74 and the target variable data has a size of 100 by 3.

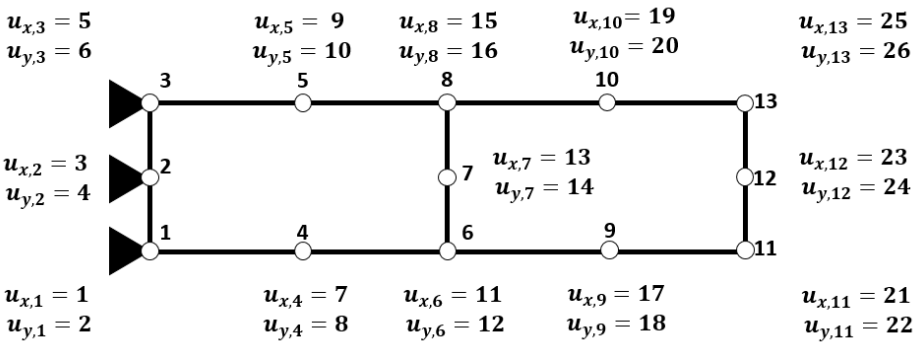
The sample size n is a critical factor for training data since n needs to be large enough to capture significant variation of the data. Although LHS was used to generate E and k, if the observations are too sparse, the modes extracted using SPO technique may not be consistent results in different sensors to be placed at every run. As n increases the resolution of the data gets finer and it allows more modes to be created before it starts to capture noise.

Even though it is necessary to capture most variations in the data, the main objective of SPO is to find the most dominant measurement variables which can recover the target variables most accurately and conduct an accurate mapping. The modes only captured when n is large are considered less important in mode-based formulations. In this study, n=100 was found to be sufficient. However, in mode-based formulations, it is certainly necessary to increase the number of n as the number of sensor variables increase due to increase in the number of elements since more sensors require better resolution to distinguish between them.

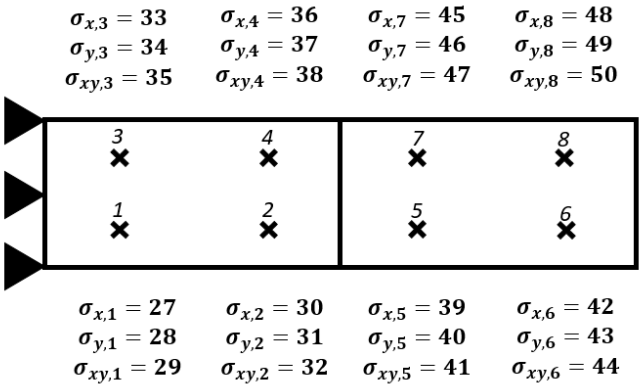
3.2 Data Scaling

Data scaling is an essential key to successful SPO since SPO is indeed a type of sensitivity analyses. This means that it requires normalisation and centering of data, especially when σ is orders higher than u and ϵ. It needs to be emphasized that there is no fixed way to scale data but it depends on the purposes and objective functions used. It is necessary to find scalings to emphasize responsiveness between the measurement and target variables. The scaling methods tested are as follows:

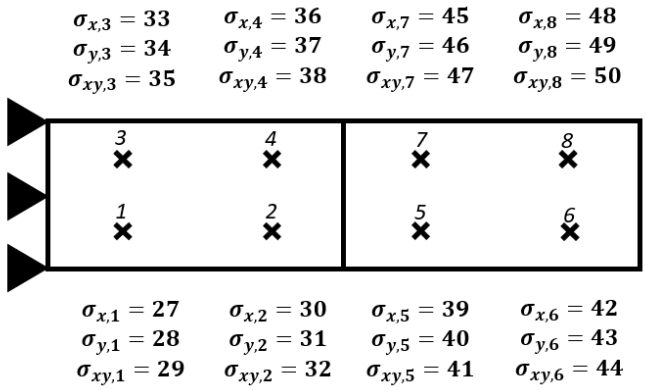
1. Component mean centering.
2. Component mean centering and norm normalisation.
3. Component mean centering and mean normalisation.
4. Component mean centering and SD (standard deviation) normalisation.
5. Vector mean centering and mean normalisation.



(a) Sensor ID numbers for u, displacements

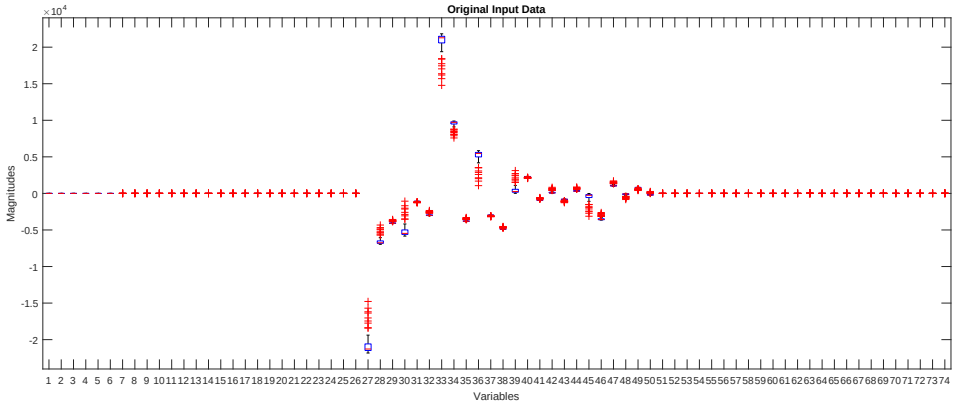


(b) Sensor ID numbers for σ, stresses

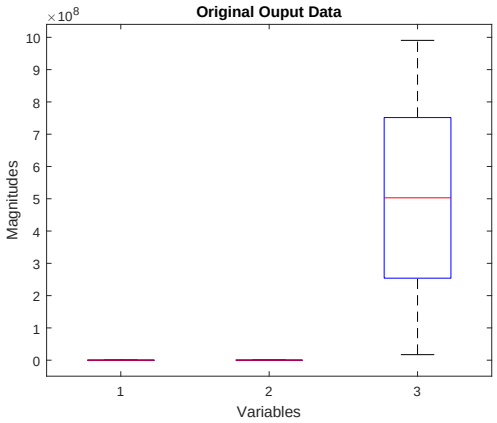


(c) Sensor ID numbers for ϵ, strains

Figure 3.2: Sensor ID numbers for the two spring problem for (a) displacements, (b) stresses and (c) strains.



(a) Original measurement data (FEM output data)



(b) Original target data (FEM input data)

Figure 3.3: Unscaled original (a) measurement and (b) target data.

6. Vector mean centering and SD (standard deviation) normalisation.

The scaling methods are largely divided into two groups: component and vector scaling methods. Component scaling scales and transforms each variable independently. Vector scaling scales each of the sensor types with the same value. Different scalings emphasize different aspects of a data set. The aim is to find the scaling that works best for a certain algorithm. The data used to demonstrate scaling is presented in Figure 3.3. The data were created by choosing both E and k to be in their high ranges (see Table 3.1). We will refer to KHEH for stiffness (K) high (H) and Young's modulus (E) high (H). In Figure 3.3(a), the untransformed data for the KHEH case (k: high and E: high ranges) shows that the magnitudes of u (sensor ID numbers: 1-26) and ϵ (sensor ID numbers: 51-74) are order smaller when compared to σ (27-50). Similarly, the target variables are also biased towards the large E (target variable ID number: 3) in Figure 3.3(b). From here, the various scaling methods applied to the original FEM data and their effects are investigated.

3.2.1 Scaling method S1: Component mean centering

The first scaling method is called component mean centering and is denoted as S1 (scaling 1). Component mean centering is the minimalistic work for scaling data since statistics naturally assumes any data to be centred at zero. This method can be expressed as follows:

$$X_{scaled} = X_{component} - \bar{X}_{component}; \quad Y_{scaled} = Y_{component} - \bar{Y}_{component} \tag{3.1}$$

The scaled measurement and output data are shown in Figures 3.4(a) and (c). As expected the transformed data look just like the original data but centred. The ranges of σ are still

relatively larger than those of u and ϵ. It is noticed that the magnitude axis is dominated by σ in measurement data and E in target data. Therefore, normalisation between them are still required.

3.2.2 Scaling method S2: Component mean centering and norm normalisation

The second scaling method is called component mean centering and norm normalisation and is denoted as S2 (scaling 2). Each variable (column) gets mean-centred and divided by its norm. The scaling method can be expressed mathematically as follows:

$$X_{scaled} = \frac{X_{component} - \bar{X}_{component}}{\|X_{component}\|}; \quad Y_{scaled} = \frac{Y_{component} - \bar{Y}_{component}}{\|Y_{component}\|} \tag{3.2}$$

Note that the norm is computed before the centering of the data not after it. The normalised data using S2 is shown in Figures 3.4(b) and (d). It is noticed that S2 suppresses σ and enhances both u and ϵ. Note that the scaled output data have almost identical ranges of variables.

3.2.3 Scaling method S3: Component mean centering and mean normalisation

The third scaling method is called component mean centering and mean normalisation and is denoted as S3 (scaling 3). Each variable gets mean-centred and divided by the mean values computed before centering (the mean values after centering are always zero). S3 is capable of investigating how large variations of variables compare to the magnitudes of the variables. S3 emphasizes relative sensitivity. The scaling method can be expressed as follows:

$$X_{scaled} = \frac{X_{component} - \bar{X}_{component}}{|\bar{X}_{component}|}; \quad Y_{scaled} = \frac{Y_{component} - \bar{Y}_{component}}{|\bar{Y}_{component}|} \tag{3.3}$$

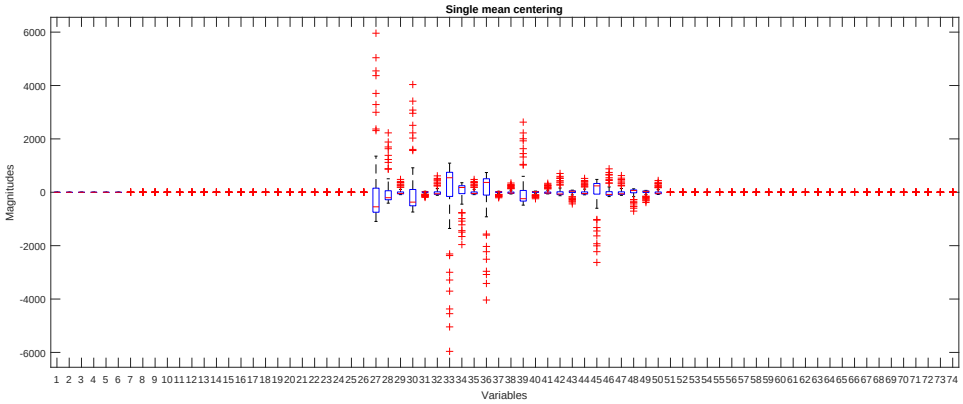
It is important to note that the mean values for the normalisation are computed between the centering of the data and taking absolute values of the data. Therefore, it is the mean of the absolute values, not absolute values of mean. The main advantage for this is that it prevents the explosive growth of a variable which has a mean of almost zero. If the mean value is still close to zero after taking absolute values of the variable, it is clearly known that the variable is not worth considering due to its minute sensitivity. The scaled measurement and target data are shown in Figures 3.5(a) and (c). Notice that the scaled variables are showing the relative sensitivities of each variable. Notice that the target variables are scaled so that they have almost identical ranges.

3.2.4 Scaling method S4: Component mean centering and SD (standard deviation) normalisation

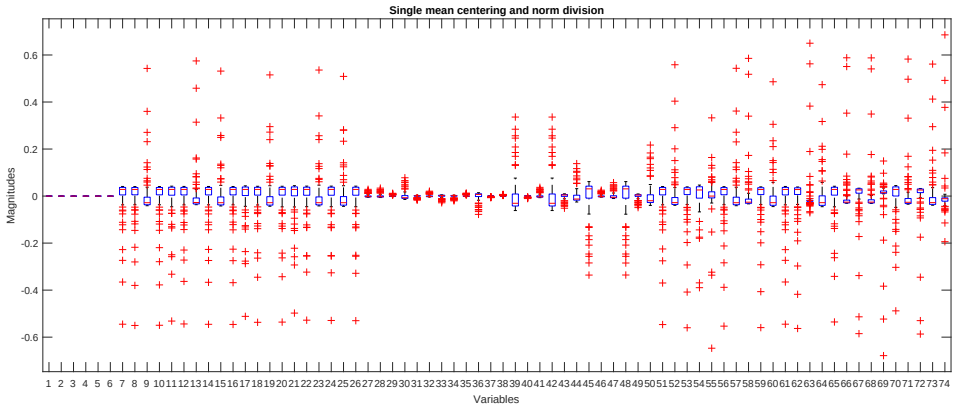
The fourth scaling method is called component mean centering and SD (standard deviation) normalisation and is denoted as S4 (scaling 4). The other well-known name of this is Z-score which is widely used in statistics for normalisation purposes as presented earlier. This scales all variables to have a mean of 0 and SD of 1. S4 is mathematically expressed as follows:

$$X_{scaled} = \frac{X_{component} - \bar{X}_{component}}{SD_{X,component}}; \quad Y_{scaled} = \frac{Y_{component} - \bar{Y}_{component}}{SD_{Y,component}} \tag{3.4}$$

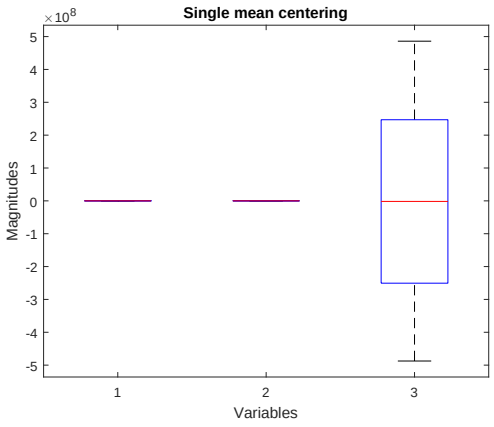
The scaled measurement and target variables are shown in Figures 3.5(b) and (d). It is observed that it rescales all variables so that they are in similar ranges. Each σ variable numbered between 26 and 50 presented in the original measurement data (see Figure 3.3(a)) seemed to contain unique information due to various means and ranges of their values. Notice



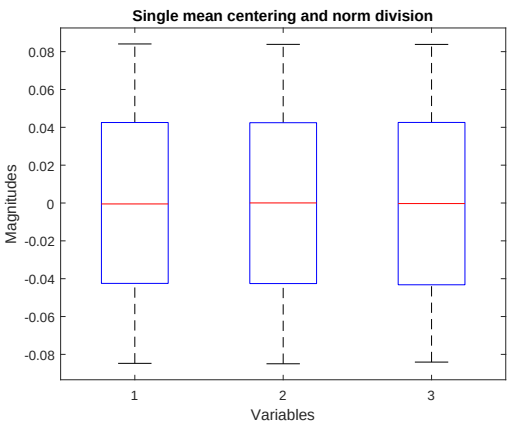
(a) Component mean centering for the measurement data



(b) Component mean centering and norm normalisation for the measurement data



(c) Component mean centering for the target data



(d) Component mean centering and norm normalisation for the target data

Figure 3.4: (a) Component mean centering S1 and (b) component mean centering and norm normalisation S2 scaling for the measurement data and (c) S1 and (d) S2 for the target data.

that the collinear coefficient presented in Eq. 2.9 includes Z-score transformation and the coefficients between those σ variables turned out to be either 1 or -1. This explains why the scaled σ variables look almost identical to each other in Figure 3.5(b). This explains why having two σ sensors is not a good choice for inverse problems. Of course, the collinearity between the sensors was possible since the coarse mesh model with only two elements is expected to have large model errors. The studies about the model errors will be presented in Chapter 5.

In SPO, uniqueness of sensors are considered to be important and checking collinearity using Z-score helps avoid gathering repeated information. Consider this example: Figure 3.6 shows two variables looking different due to the different means and variances, being transformed using Z-score. It also keeps the covariance matrices between the two datasets unbiased by adjusting the orders of magnitudes of the variables in the datasets and this is why Z-score has been used popularly in PLS regression. Remember that the objective of SIMPLS is to maximise covariance between measurement and target data. Therefore, it is essential to consider unbiased covariance matrices.

3.2.5 Scaling method S5: Vector mean centering and mean normalisation

The fifth scaling method is called vector mean centering and mean normalisation and is denoted as S5 (scaling 5). The data first gets divided into their types (i.e. σ, u, ϵ, E, k_1, k_2). The mean for both centering and normalisation is computed from each group. The mean is the average of all the variables in a certain group irrespective of spatial location. The scaling can be expressed as follows:

$$X_{scaled} = \frac{X_{group} - \bar{X}_{group}}{|\bar{X}_{group}|}; \quad Y_{scaled} = \frac{Y_{group} - \bar{Y}_{group}}{|\bar{Y}_{group}|} \tag{3.5}$$

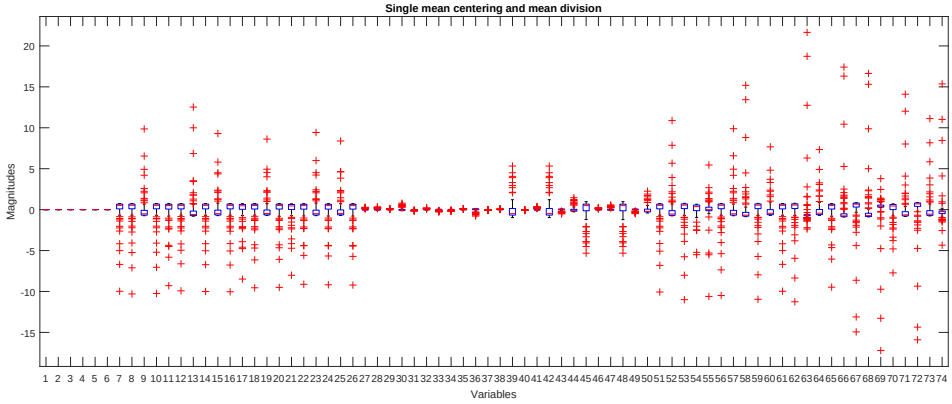
Similar to the component mean normalisation, the absolute values are firstly computed before being averaged to compute the means. The scaled measurement data X (Figure 3.7(a)) and target data Y (Figure 3.7(c)) show that the differences of magnitudes between the groups have been reduced. The scaled measurement data clearly show that the small u and ϵ groups have been stretched and the large σ group has been compressed. It is important to note that the variables in a group get divided by the same value. Therefore, the relative difference between variables in a group remains unchanged. For instance, larger σ values remain greater than smaller σ values. However, this can become problematic when the variation within a group becomes too large. The scaled target variables Y show different shapes from the previous scaling methods since the 1st and 2nd variables are both k and they share the same mean for both centering and normalisation. For this reason, the means of the variables are not zeros after scaling.

3.2.6 Scaling method S6: Vector mean centering and SD (standard deviation) normalisation

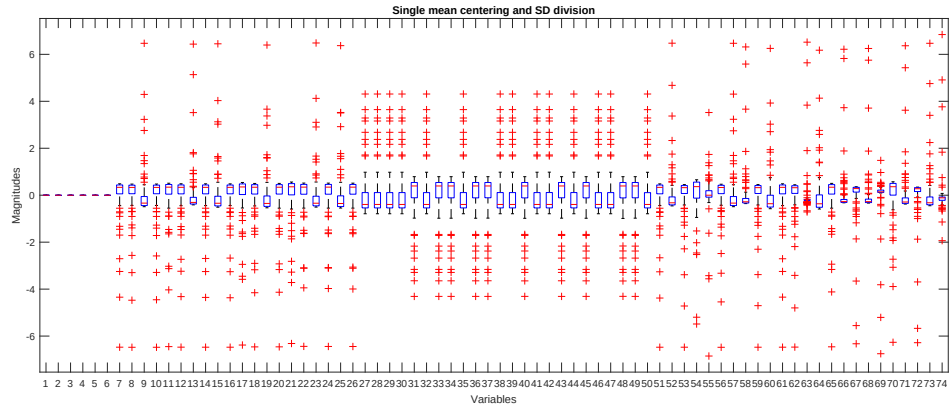
The last scaling method is called vector mean centering and SD normalisation and is denoted as S6 (scaling 6). S6 works similar to S5, instead of dividing each group by the group mean, we normalise by the group SD's. The scaling method can be expressed as follows:

$$X_{scaled} = \frac{X_{group} - \bar{X}_{group}}{SD_{X,group}}; \quad Y_{scaled} = \frac{Y_{group} - \bar{Y}_{group}}{SD_{Y,group}}. \tag{3.6}$$

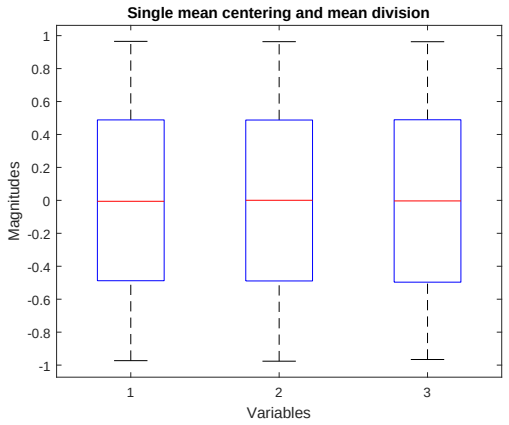
The scaled measurement and target data are shown in Figures 3.7(b) and (d). It is noticed that they look almost identical to the results of S6 but the ranges of the values are smaller. The two stiffness variables k_1 and k_2 (target variable ID numbers: 1 and 2) do not have zero means since they get subtracted by the mean of the two variables. This applies to the three groups of variables in the measurement data X. It is expected that S5 and S6 give similar performance.



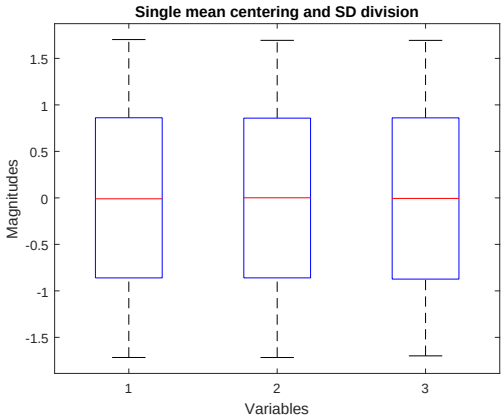
(a) Component mean centering and mean normalisation



(b) Component mean centering and SD normalisation

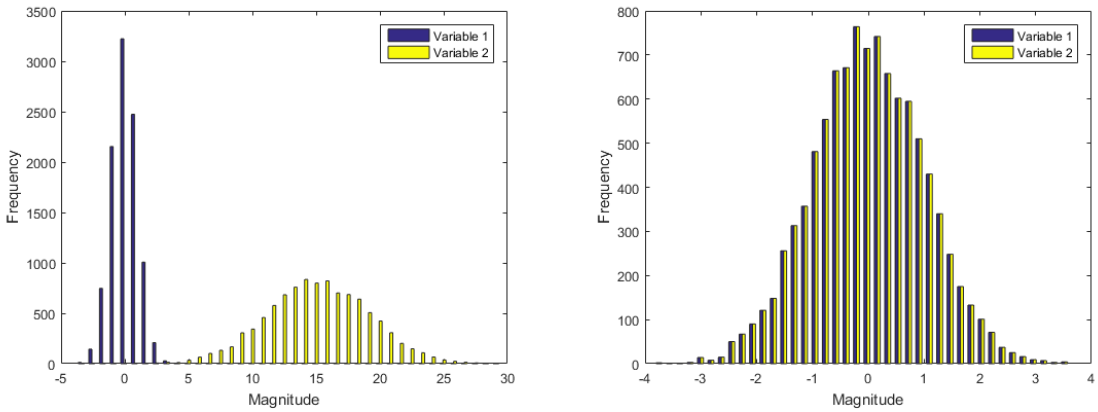


(c) Component mean centering and mean normalisation



(d) Component mean centering and SD normalisation

Figure 3.5: (a) Component mean centering and mean normalisation S3 and (b) component mean centering and SD normalisation S4 scaling for the measurement data and (c) S3 and (d) S4 for the target data.



(a) Before Z-score

(b) After Z-score

Figure 3.6: Effects of Z-score scaling.

3.3 Proxy Objective Functions

As discussed, scaling emphasizes important aspects of the datasets. The role of an objective function is to combine this information into a single scalar to rank the various solutions. The scaling and objectives are related in that a scaling that works well on one objective may be problematic on another. The actual objective function for sensor pair placement in an inverse problem is, of course, to optimise the inverse problem for a given sensor pair which is computationally intractable. The aim here is to establish proxy objective functions that capture the essence of the actual objective function but in a computationally tractable way. We consider six proxy objective functions based on PLSR (SIMPLS Algorithm) or PCA for the mode-based formulations.

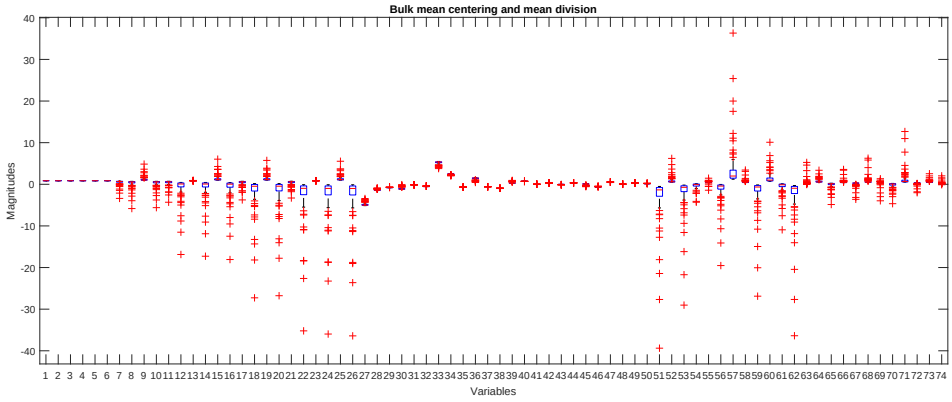
The main aim of SIMPLS is to maximise the covariance between the measurement and target data by decomposing the covariance of two datasets. Note that the original SIMPLS [47] is considered in this study. The sets of measurement and target variables get decomposed into scores and loadings which are related to participation coefficients, eigenvalues and eigenvectors. The eigendecomposition of SVD applied to a matrix such as covariance matrix allows for quantifying and comparing the amount of information included in the matrix. Therefore, various means of rearranging the outputs of SVD (U, S and V defined in Section 2.3) could be used as the potential proxy objectives of the mode-based formulations. The first two objective functions are based on PLSR, while the remaining four are based on SVD. The number of modes, A is very much dependent on variable types, scaling methods and also on the decomposition techniques.

3.3.1 Function F1: Minimisation of Y-residuals

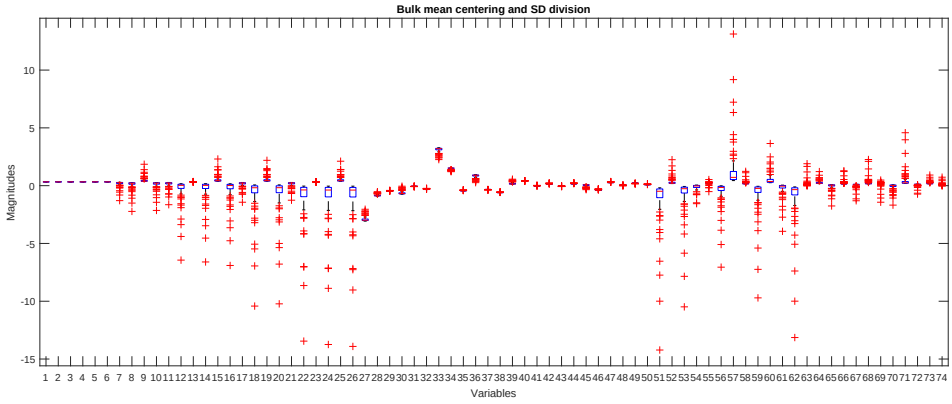
The first objective function is to minimise the norm of Y-residuals ($\|Y_{residuals}\|$) which is minimising the regression error and is denoted as F1 (function 1). The regression coefficient, β_{pls} is expressed as RQ' where R is the measurement X block factor weights and Q is the target Y block factor loadings [47]. The vector of Y-residuals is simply given by

$$Y_{residuals} = Y - X\beta_{pls} = Y - X(RQ'). \tag{3.7}$$

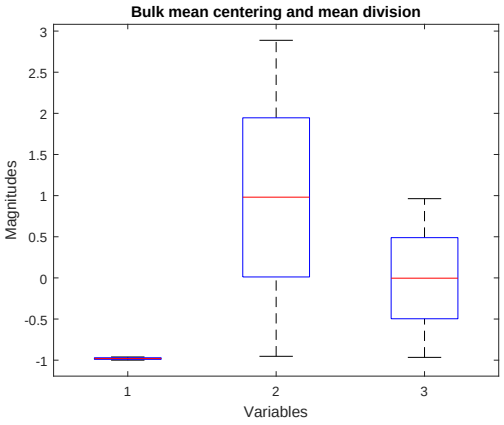
SIMPLS regresses the measurement data on the target data by constructing linear combinations of the measurement variables. This effectively linearises the inverse problem over an entire domain, which is a significant simplification to the actual non-linear inverse response over the domain. It is critical to remember that the core purpose of it is not simply to minimise the norm of regression errors but to construct modes (components or latent variables [47]) that contain the most sensitive measurements X to the targets Y for inverse problems.



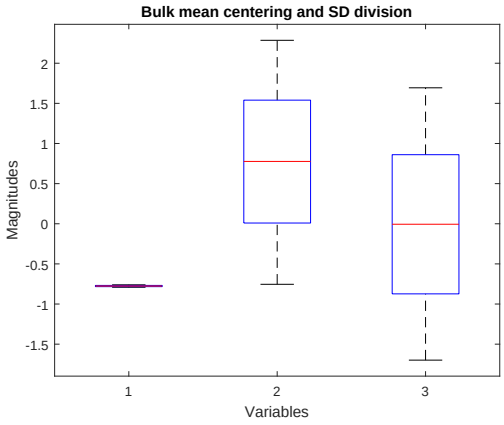
(a) Vector mean centering and mean normalisation



(b) Vector mean centering and SD normalisation



(c) Vector mean centering and mean normalisation



(d) Vector mean centering and SD normalisation

Figure 3.7: (a) Vector mean centering and mean normalisation S5 and (b) vector mean centering and SD normalisation S6 scaling for the measurement data and (c) S5 and (d) S6 for the target data.

Therefore, by strictly performing only linear regressions, PLS can concentrate on assembling modes that capture the essence of the observed response.

3.3.2 Function F2: Maximisation of Total Variance Explained in Y

The second objective function is to maximise the percentage of variance explained for the Y variables, $varY$ and is denoted as F2 (function 2). The expression for $varY$ for each mode is given by

$$varY_a = \frac{\sum_{i=1}^{n} |Q_{i,a}|^2}{\sum_{i=1}^{n}\sum_{j=1}^{m} |Y_{i,j}|^2}; \quad a = 1, ..., A, \tag{3.8}$$

where n is the 100 LHS FEM observations. Here, the m and a denote the number of target variables and the mode index, respectively. It is important to acknowledge that the output loading matrix Q contains not only directions (eigenvectors) but also magnitudes (eigenvalues). This objective function is related to the minimisation of the $Y_{residuals}$ since the regressions are based on the directions that maximise the covariance between the sensor measurement and inference domains. $Y_{residuals}$ is the result of a linear mapping from the sensor measurement to inference domain. Hence, F2 assesses the quality of the variables, whereas F1 assesses the consequence of mapping the variables to the target domain. Consequently, there is a considerable difference between F1 and F2 in that F2 is susceptible in selecting the same information in different descriptions that have high variance. For instance, when finding the volume of a cylinder, both the diameter and radius may be identified should it have high variance although it is essentially the same information. F1, on the other hand, is less susceptible to this potential issue. The susceptibility of F2 to select identical information in different descriptions is largely dictated by the number of modes that are chosen. An increase in number of modes tends to increase the susceptibility to select identical information.

The correlation coefficients allow us to investigate the linear correlation amongst the target variables, which is given by

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}, \tag{3.9}$$

where the bar and s denote the mean and variance, respectively. The r values can only be between 1 and -1 which indicate strong positive and negative linear correlations, respectively. An r of 0 implies no correlation. Figure 3.8. depicts eliminated variables when $|r| < 10^{\alpha}$, where α is the tolerance on the X-axis. The measurement space removes all but one variable when α is close to 0, while large negative α's result in 7 eliminated variables that include the six fixed displacements. Note linear correlation coefficients do not identify correlated non-linear relationships such as x and x^2, that is the same information in a linear and non-linear description.

3.3.3 Function F3: Maximisation of Total Singular Values

SVD naturally requires much less computational time than PLS. Therefore, SVD is desired for SPO applications, in particular when industrial scale problems are to be considered. Function 3 aims to maximise the covariance information from the covariance matrix, $X^T Y$

$$\arg\max_x (\sum S_{xy}), \tag{3.10}$$

with S_{xy} the positive singular value diagonal matrix of covariance matrix, $X^T Y$. The higher the correlation between X and Y the larger the singular value.

3.3.4 Function F4: Maximisation of Normalised Total Singular Values

F4 is merely F3 normalised by the norm of the singular value diagonal matrix

$$\arg\max_x \left(\frac{\sum S_{xy}}{\|S_{xy}\|} \right). \tag{3.11}$$

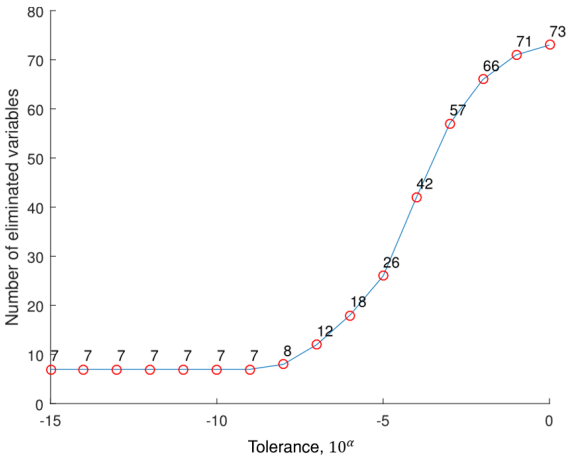


Figure 3.8: Filtering measurement variables using correlation coefficients.

As the nominator tries to maximise the information contained in the covariance matrix, the denominator tries to minimise the norm of the covariance matrix which effectively minimises size difference among singular values. This forces the information of each mode to explain the same amount of the Y data.

3.3.5 Function F5: Maximisation of Singular Values of the Regression Coefficients

Function 5 (F5) has the similar form as F4 except the fact that the diagonal matrix does not come from a covariance matrix but from $\beta = X^{-1}Y$. Since the size of X is not square in most cases, the inverse denotes the pseudo-inverse. Since, $X\beta = Y$, the β regression coefficients β can simply be seen as the variation of Y as a function of X (i.e. the gradient of the regression model). The singular values of the regression coefficients are merely,

$$\arg\max_{x}\left(\frac{\sum S_\beta}{\|S_\beta\|}\right). \tag{3.12}$$

This approach is susceptible to multicollinearity but the main goal of this algorithm is to attempt to maximise the gradient sensitivity between X and Y.

3.3.6 Function F6: Maximisation of Singular Values of the Regression Coefficients with Direction Scale

Lastly, consider F5 scaled exponentially using the matrix norm of the left U_x and right U_{xy} eigenvector matrices of X^TX and X^TY as follows:

$$\arg\max_{x}\left(\frac{\sum S_\beta}{\|S_\beta\|}\right)^{(\|U_x^T U_{xy}\|)}. \tag{3.13}$$

The correlation between the two eigenvectors is computed and normalised. The higher the exponent the better the correlation between variables that explain X and more useful towards explaining Y.

3.4 Optimisation method

A number of data scaling methods and objective functions have been discussed. In order to evaluate their ability to optimally place sensors, an appropriate optimisation strategy is required. Firstly, sensor placement optimisation is a binary optimisation problem as a sensor is placed or not. The three optimisation strategies considered in this study are:

1. Exhaustive combinatorial method: Exhaustive combinatorial method is the best method to find the global minimum/maximum as all possible sensor combinations are evaluated. It requires an enormous number of function evaluations for large dimensional problems that quickly become computationally intractable due to the exponential growth of potential combinations. This method is only considered for a small number of sensors. The advantage is that the global minimum of the problem is guaranteed to be found [36].

2. BGA (Binary genetic algorithm): Binary genetic algorithm is a binary function value based heuristic optimisation algorithm which imitates the process of natural selection. Since the method is inspired by evolution, it uses biological techniques such as inheritance, mutation, selection and crossover. This technique can play an important role when a large number of sensors are involved. Its disadvantage would be that it cannot guarantee that the global maximum/minimum will be found [38, 50, 51].

3. Addition method: Addition method is one of the fastest optimisation methods since it selects sensors one at a time until the prescribed number of sensors are collected. The largest disadvantage of this method is that it might start off by choosing a less important sensor since it does not know a sensor performs with other sensors.

Chapter 4

Mode-Based Formulations: Reduced Sensor Optimisation

For the evaluation of proxy objective functions and dataset scalings, it is necessary to know the optimal solutions. In this chapter, the available number of sensors are reduced from the original 74 sensors to 24 sensors as shown in Figure 4.1(a). All possible combinations of sensors were examined and each combination has three measurement variables (sensors). Only three variables were used since that is the minimum requirement for recovering the exact solution of the three target variables (k_1, k_2 and E). The optimal solutions were obtained for three cases and each case consists of different ranges for k (both k_1 and k_2) and E: (1) high k and low E (KHEL), (2) high k and high E (KHEH) and (3) low k and high E (KLEH) as are specified in Table 3.1.

1. KHEL: $k_1 = 10$ - 10^3 N/m, $k_2 = 10^3$ - 10^5 N/m and $E = 10^5$ - 10^7 Pa
2. KHEH: $k_1 = 10$ - 10^3 N/m, $k_2 = 10^3$ - 10^5 N/m and $E = 10^7$ - 10^9 Pa
3. KLEH: $k_1 = 1$ - 10^2 N/m, $k_2 = 10$ - 10^3 N/m and $E = 10^7$ - 10^9 Pa

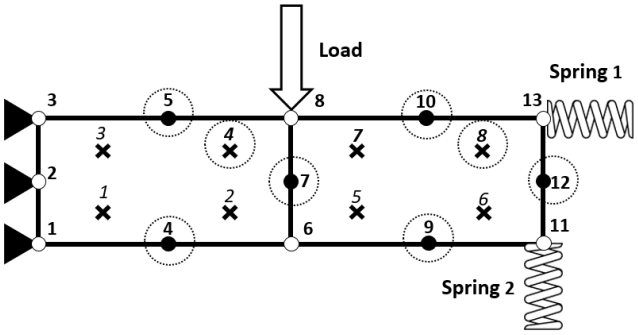
Although the three cases consider the same experiment, the optimal set of sensors differ and the well or ill-posedness of the problems differ. The global minima for the optimal sensors are tabulated in Table 4.1. The optimal sensors have the highest accuracy and the lowest number of iterations for each case and are graphically presented in Figures 4.1(b) - (d). The inverse problems were solved on the ten randomly generated test datasets identically each for every sensor combination of three sensors ($N_{variables} = 3$) by minimising the sum of error squared

$$\mathbb{E} = \sum_{i=1}^{N_{variables}} \left(\frac{(X_{i,actual} - \bar{X}_i) - (X_i - \bar{X}_i)}{SD_i}\right)^2 = \sum_{i=1}^{N_{variables}} \left(\frac{X_{i,actual} - X_i}{SD_i}\right)^2, \tag{4.1}$$

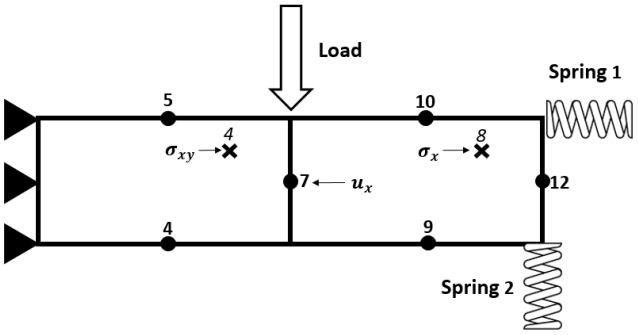
as in the previous chapter. The minimisation of $\mathbb{E}$ was done using the Nelder-Mead method with a maximum number of function evaluations and iterations of 10000. The initial guesses were always fixed to the logarithmic means of the chosen target variable ranges (i.e. 10^6 when its range is $10^5 - 10^7$) for every ten datasets tested on each combination. The logarithmic means were chosen since the ranges are often built around the most confident assumptions

Cases	KHEL	KHEH	KLEH
Variables	13, 38, 48	38, 48, 62	48, 50, 74
Accuracy	10/10	10/10	10/10
Iterations	229.8	238.2	206.6

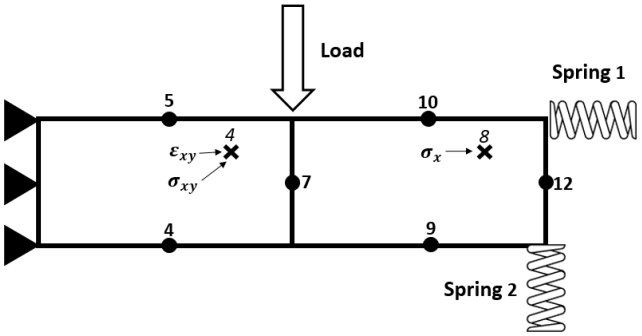
Table 4.1: Optimal solutions



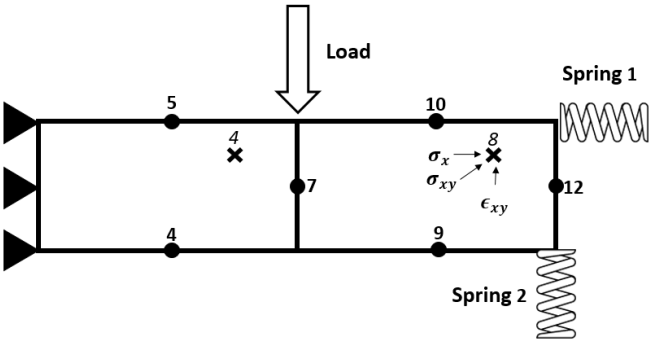
(a) Reduced sensor options



(b) High k and low E: KHEL

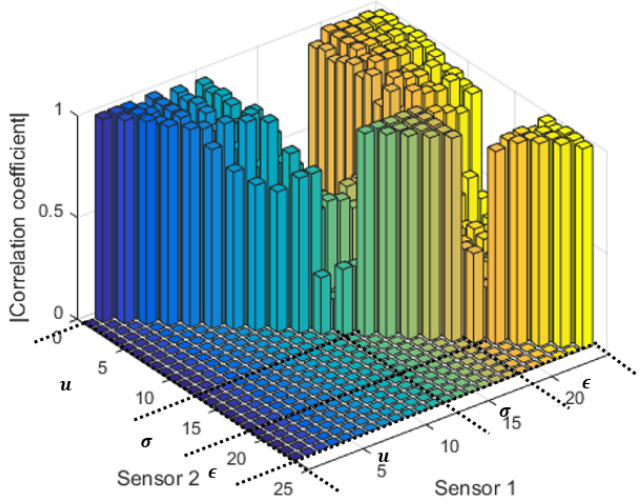


(c) High k and high E: KHEH

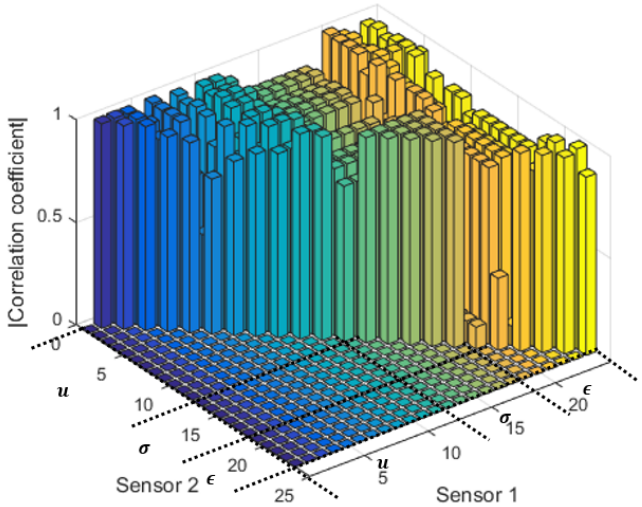


(d) Low k and high E: KLEH

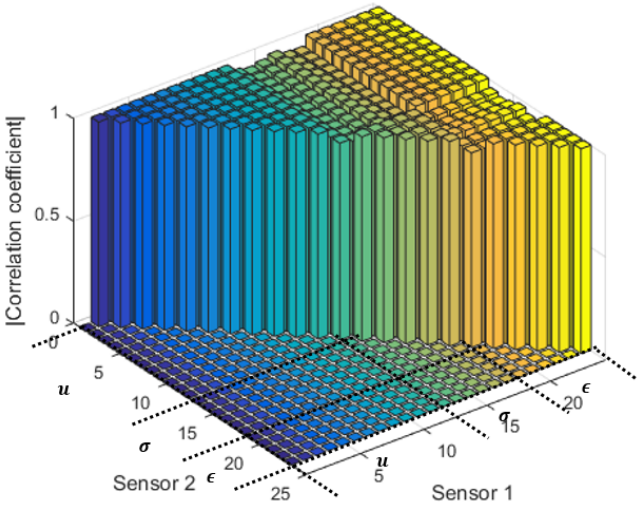
Figure 4.1: Graphical presentation of the beam with single spring for (a) the reduced sensor options and optimally positioned sensors for (b) KHEL, (c) KHEH and (d) KLEH.



(a) High k and low E: KHEL



(b) High k and high E: KHEH



(c) Low k and high E: KLEH

Figure 4.2: Absolute correlation coefficients between sensors for (a) KHEL, (b) KHEH and (c) KLEH.

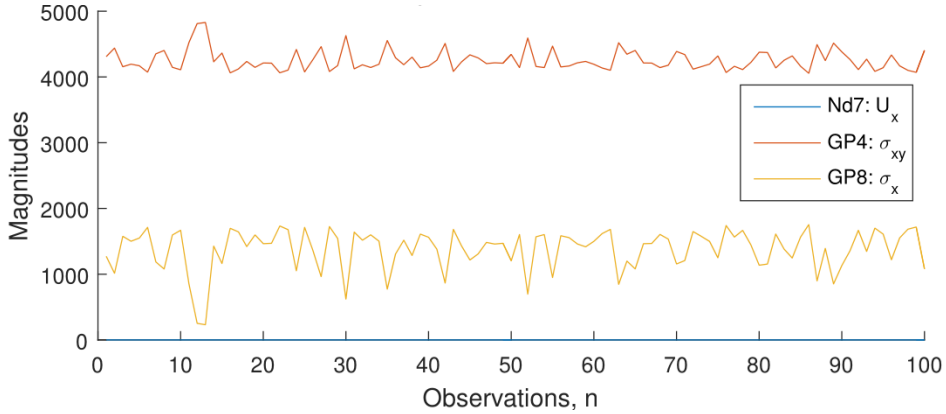
of the target variables. Solving only ten test datasets in each combination together with the reduced number of available sensors allows the inverse problems computationally more tractable. The Nelder-Mead was chosen as the optimisation algorithm for the inverse problem as the objective is likely multi-modal, while the low problem dimensionality is well suited for the gradient-free Nelder-Mead approach.

The optimal sensors are placed where they are able to predict both E and k well. For an insensitive target variable, the sensors are placed to focus more on the insensitive target variables than better solving sensitive target variables. This is evident from Figures 4.1(b) - (d). The KHEL case (Figure 4.1(b)) has its sensors on the two Gauss points for measuring σ_{xy} close to the root and σ_x close to the tip of the beam. Due to the relatively high flexibility of the beam, measuring u_x at the middle of the beam is meaningful. As E is increased as in the KHEH case (Figure 4.1(c)), the beam becomes stiffer and measuring ϵ_{xy} at the root becomes more meaningful than measuring u_x at the middle of the beam. As the ranges of the two springs become less while E increases, the beam became stiffer such as in the KLEH case (Figure 4.1(d)). Since the two springs became highly insensitive to the sensor measurements, the sensors are brought closer to the two springs. Recall that k_1 has the smallest ranges. Therefore, σ_x close to k_1 seems to be important as it is present in all solutions.

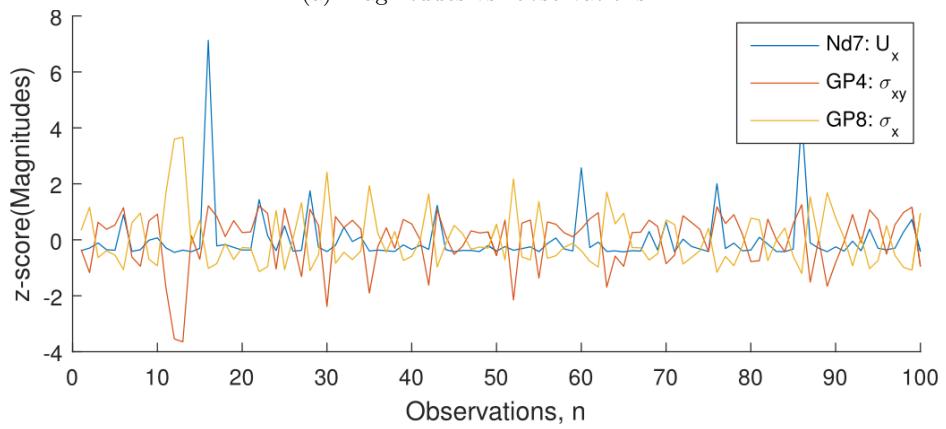
As intuitively expected, the KLEH case is the most ill-posed due to the low beam stiffness relative to the high spring stiffness while the KHEL case is the most well-posed as the opposite case. This also means the similarities between the measurements are the largest for the KLEH case and the least for the KLEH case. The absolute correlation coefficients between two sensors for each case are shown in Figure 4.2. This clearly illustrates that KLEH is the most correlated, while the KHEL is the least correlated. Therefore, numerous combinations of the sensors can be used to solve the inverse problems for the KHEL case, while only a few combinations may be useful to solve the KLEH case.

Figures 4.3 - 4.5 show the magnitudes of the optimal sensors for each case along the hundred observations. Consider the KHEL case shown in Figure 4.3 which is the mixture of u and σ. It is noticed that there are large magnitude differences between each measurement. When they get normalised using Z-score, they become similarly weighted and not biased towards a type with larger magnitudes. After taking the absolute of the second figure as shown in the third figure, it is shown that the σ_{xy} at GP (Gauss point) 4 are almost identical to the σ_x at GP 8. The differences between the information contained in the σ sensors are extremely small and they are highly correlated with each other. The correlation coefficient between the two sensors is close to 0.99 just under 1. The result shows that the inverse problems are ill-posed as we will explore later when considering stochastic noise as it aims to infer sensible information from very small difference. The optimal sensors for the KHEH case are shown in Figure 4.4. After they were transformed using Z-score, it is again noticed that the two optimal sensors, σ_{xy} sensor at GP 4 and σ_x at GP 8 look almost identical and their signs are different. Due to the increase in the stiffness of the beam, it pushed the springs less and created less distinctive σ measurements compared to the two σ sensors in the KHEL case. The optimal sensors for the KLEH case are shown in Figure 4.5. Recall that all the sensors were placed on GP 8. Therefore, the extracted patterns (Z-score transformation) of all the three sensors σ_x, σ_{xy}, ϵ_{xy} are closely related to each other, yet, none of them are perfectly identical to each other. From these behaviours of the optimal sensors, a few observations about the inverse problems can be noticed:

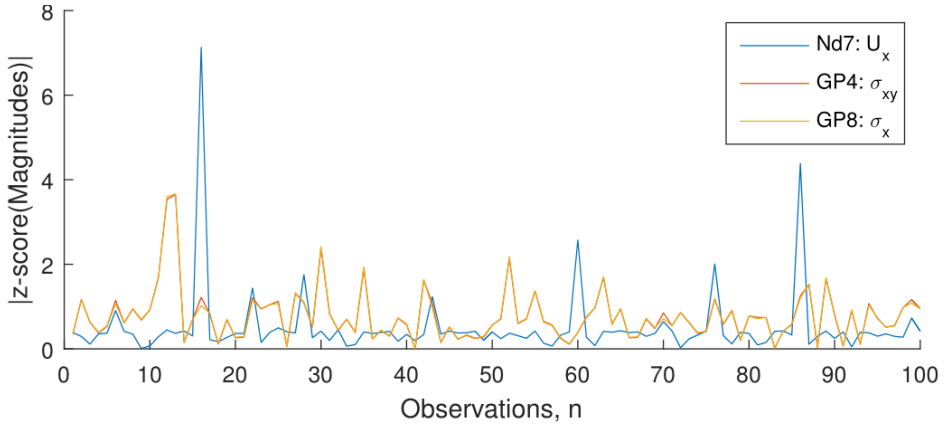
- A sensor is not chosen more than once. Although there is a sensor which might be the best, re-selection of the same sensor is pointless.
- The optimal sensors can have either strong positive or negative correlations but not perfect correlations.
- The more ill-posed the problem, the higher the correlation between the optimal sensors in the Z-score domain.



(a) Magnitudes vs. observations

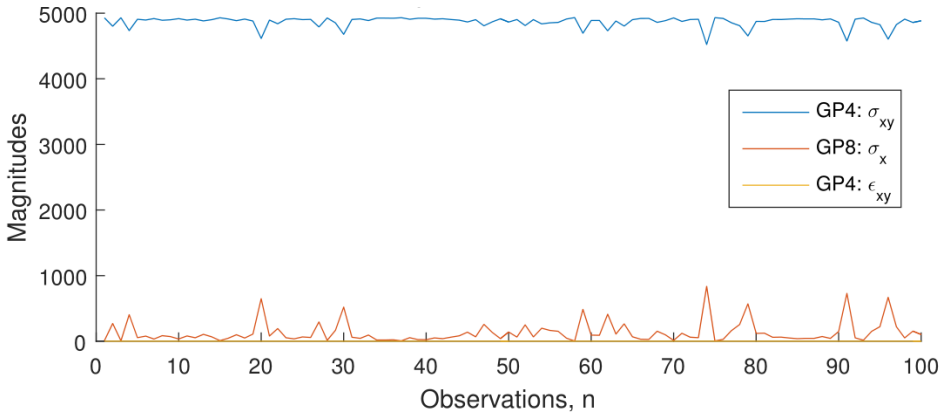


(b) Z-score(magnitudes) vs. observations

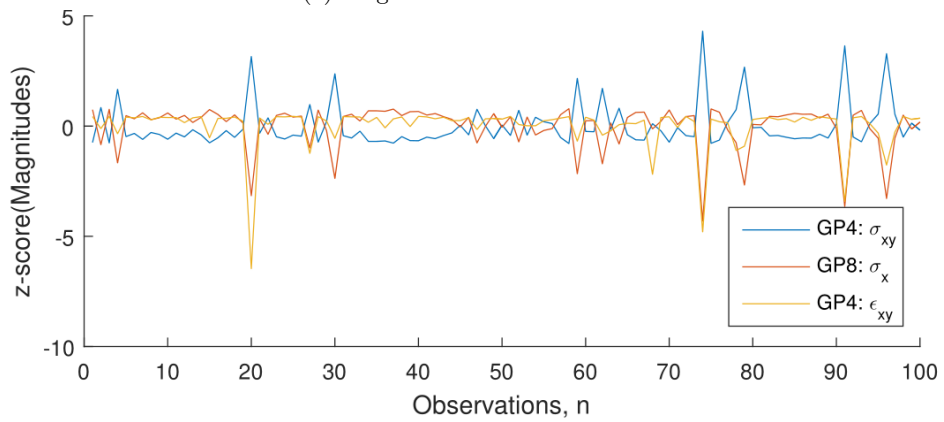


(c) |Z-score(magnitudes)| vs. observations

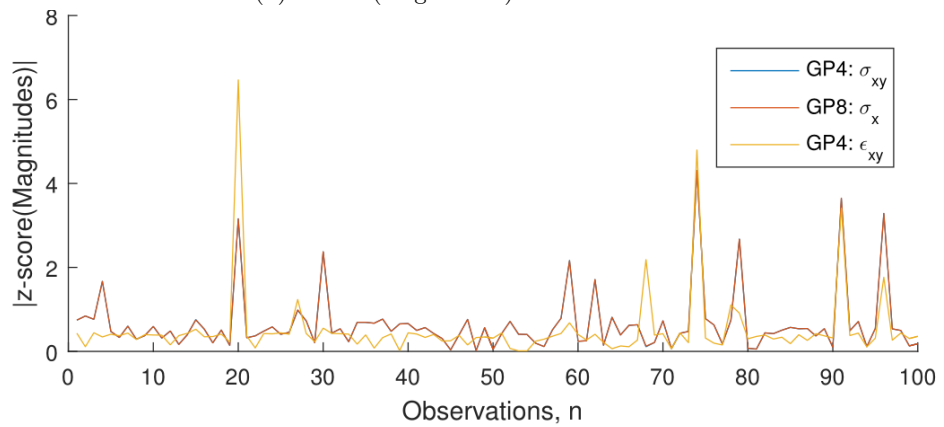
Figure 4.3: Optimal sensors for KHEL; (a) magnitudes vs. observations, (b) Z-score(magnitudes) vs. observations and (c) |Z-score(magnitudes)| vs. observations.



(a) Magnitudes vs. observations

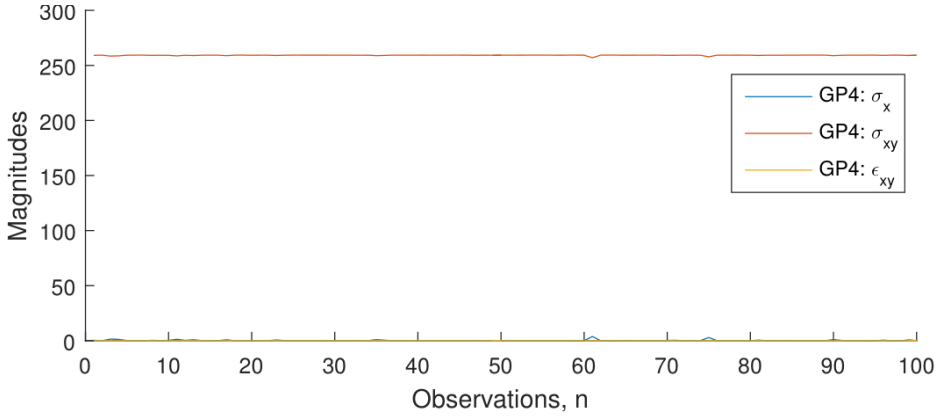


(b) Z-score(magnitudes) vs. observations

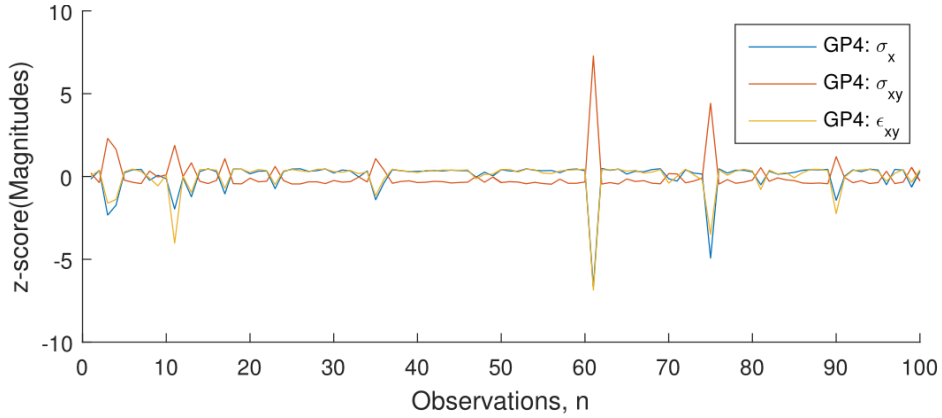


(c) |Z-score(magnitudes)| vs. observations

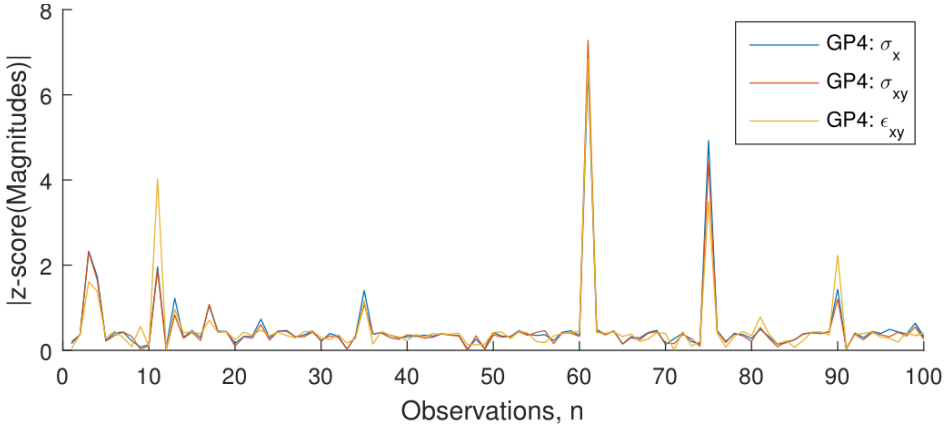
Figure 4.4: Optimal sensors for KHEH; (a) magnitudes vs. observations, (b) Z-score(magnitudes) vs. observations and (c) |Z-score(magnitudes)| vs. observations.



(a) Magnitudes vs. observations



(b) Z-score(magnitudes) vs. observations



(c) |Z-score(magnitudes)| vs. observations

Figure 4.5: Optimal sensors for KLEH; (a) magnitudes vs. observations, (b) Z-score(magnitudes) vs. observations and (c) |Z-score(magnitudes)| vs. observations.

Objective functions	
F1	$\arg\min_x(\|Y_{residuals}\|)$
F2	$\arg\max_x(varY)$
F3	$\arg\max_x(\sum S_{xy})$
F4	$\arg\max_x\left(\frac{\sum S_{xy}}{\|S_{xy}\|}\right)$
F5	$\arg\max_x\left(\frac{\sum S_{\beta}}{\|S_{\beta}\|}\right)$
F6	$\arg\max_x\left(\frac{\sum S_{xy}}{\|S_{xy}\|}\right)^{(\|U_x^T U_{xy}\|)}$
Scaling methods	
S1	Component mean centering
S2	Component mean centering and norm normalisation
S3	Component mean centering and mean normalisation
S4	Component mean centering and SD normalisation
S5	Vector mean centering and mean normalisation
S6	Vector mean centering and SD normalisation

Table 4.2: Objective functions and scaling methods

We now consider the various scaling of variables when evaluating a number of the proxy objective function to optimize the sensor placement. All 36 possible combinations of the six dataset scaling methods and the six objective functions were performed for all three inverse problem results which are tabulated in Tables 4.3-4.5. Due to the small total number of available sensors, the exhaustive combinatorial method was computationally tractable. Hence, it was implemented as the optimisation method for the sensor placement for each proxy combination. Each table presents the ID numbers of the chosen sensors (Sen.1,2 and 3), accuracies out of ten and number of iterations when 1-3 modes are considered. The ten identical inverse problems tested previously to find the optimal solutions were again tested on the chosen sensors of each proxy in order to directly compare the performance of the proxies to the optimal solutions for each case by solving the ten inverse problems using Nelder-Mead method with a maximum number of function evaluations and iterations of 10000. The initial guesses were again the logarithmic means of the chosen ranges. In the table, the accuracies are denoted by 'acc.' and the number of iterations are denoted by 'iter.'. Note that the functions and scaling methods are numbered as presented in Table 4.2.

It is important to emphasize that the objective of this observation is to find out which combinations of objective functions and scaling methods perform successfully compared to the global optimum presented in Table 4.1. The sensors obtained for modes A from 1 to 3 were tested using the inverse problems for every proxy combination and the best result among the three solutions measured by comparing the accuracies out of ten was chosen to represent the performance of the proxy combination. In case the sensors obtained from different modes have the same accuracy, the sensors with lower number of iterations were chosen as the best. Note that F4, F5 and F6 were only tested for two modes $A = 2$ and 3 since $A = 1$ would give identical function values for all combinations of variables in the optimisation. Choosing the number of modes, A in advance is challenging since it is difficult to know in advance which A would perform the best. Note that the prior selection of modes using over-fitting of SIMPLS regression has been attempted and the largest mode before over-fitting occurs was selected initially. Yet, it turned out not to be a promising indicator for the selection of A for the mode-based formulations in the study as it often did not match the performance of the various modes. The NaN values in the table simply mean that its accuracy is zero.

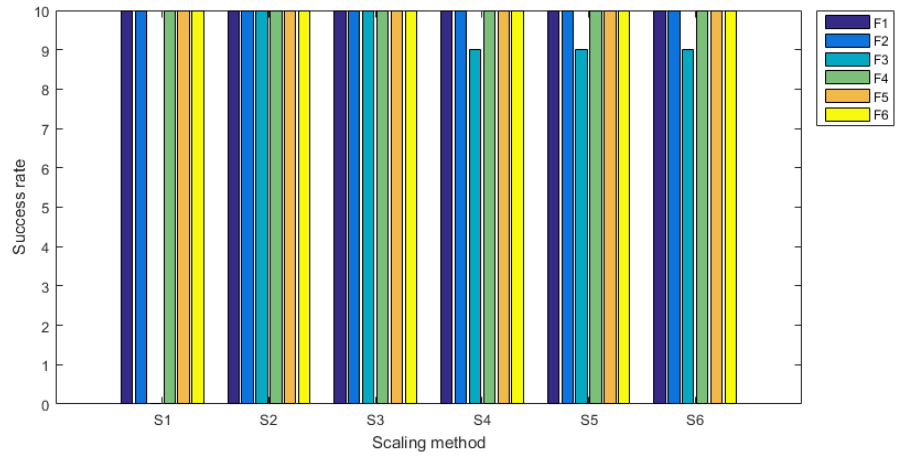
	Func.	F1			F2			F3			F4		F5		F6	
Scal.	Mode	A=1	A=2	A=3	A=1	A=2	A=3	A=1	A=2	A=3	A=2	A=3	A=2	A=3	A=2	A=3
S1	Sen.1	14	8	24	14	8	24	36	36	36	17	17	23	23	17	17
	Sen.2	18	24	36	18	24	36	48	48	48	19	19	48	48	19	19
	Sen.3	62	37	38	62	37	38	50	50	50	24	24	60	60	24	24
	Acc.	9	**10**	**10**	9	**10**	**10**	0	0	0	**10**	**10**	**10**	**10**	**10**	**10**
	Iter.	310.6	**273.1**	**278.5**	310.6	**273.1**	**278.5**	NaN	NaN	NaN	**299.3**	**299.3**	**254.3**	**254.3**	**299.3**	**299.3**
S2	Sen.1	23	24	13	37	20	37	19	18	24	8	8	10	20	17	24
	Sen.2	49	37	37	38	24	72	36	24	49	24	24	13	24	24	36
	Sen.3	50	48	74	50	49	73	49	49	72	36	36	48	49	37	62
	Acc.	**10**	**10**	**10**	0	9	**10**	**10**	**10**	**10**	**10**	**10**	**10**	9	**10**	9
	Iter.	**283.1**	**256.5**	**269.5**	NaN	286.6	**293**	**275.8**	**280.4**	**273.1**	**294.5**	**294.5**	**268.1**	286.6	**282.4**	287.1
S3	Sen.1	20	24	13	37	20	37	17	23	17	19	9	9	20	18	23
	Sen.2	37	37	37	38	24	72	60	49	23	20	24	10	24	20	24
	Sen.3	49	48	74	50	49	73	74	60	49	24	49	48	49	49	49
	Acc.	**10**	**10**	**10**	0	9	**10**	**10**	**10**	**10**	**10**	**10**	**10**	9	9	**10**
	Iter.	**267.1**	**256.5**	**269.5**	NaN	286.6	**293**	**283**	**256.5**	**251.7**	**307.7**	**271.5**	**286.3**	286.6	296.6	**262.1**
S4	Sen.1	14	24	13	37	20	37	37	20	20	19	9	20	23	19	23
	Sen.2	48	36	37	49	24	72	49	37	37	24	24	23	48	24	24
	Sen.3	50	48	74	50	37	73	50	50	50	60	37	36	60	60	48
	Acc.	**10**	**10**	**10**	0	9	**10**	0	9	9	9	**10**	**10**	**10**	9	**10**
	Iter.	**249.4**	**258.3**	**269.5**	NaN	277.6	**293**	NaN	263.1	263.1	312.1	**283.7**	**258.8**	**254.3**	312.1	**269.2**
S5	Sen.1	7	14	19	9	24	24	14	18	18	7	7	13	13	7	7
	Sen.2	36	36	48	36	37	36	18	24	38	10	10	24	24	8	10
	Sen.3	61	62	62	61	38	38	62	62	62	48	60	73	73	48	60
	Acc.	**10**	**10**	**10**	8	**10**	**10**	9	9	5	9	**10**	**10**	**10**	**10**	**10**
	Iter.	**288.5**	**307.3**	**249.5**	284	**269.6**	**278.5**	310.7	317.3	287.6	288.4	**303.7**	**273.5**	**273.5**	**303.4**	**303.7**
S6	Sen.1	50	24	24	9	9	24	14	14	18	10	10	36	7	7	10
	Sen.2	72	36	36	36	24	36	18	38	38	19	19	37	23	48	19
	Sen.3	73	74	38	50	36	38	62	62	62	48	48	74	36	60	48
	Acc.	**10**	**10**	**10**	**10**	**10**	**10**	9	8	5	**10**	**10**	**10**	**10**	**10**	**10**
	Iter.	**299.5**	**292.8**	**278.5**	**252.8**	**304**	**278.5**	310.7	312.4	287.6	**255**	**255**	**245.8**	**252.9**	**270.8**	**255**

Table 4.3: Reduced sensor option: KHEL

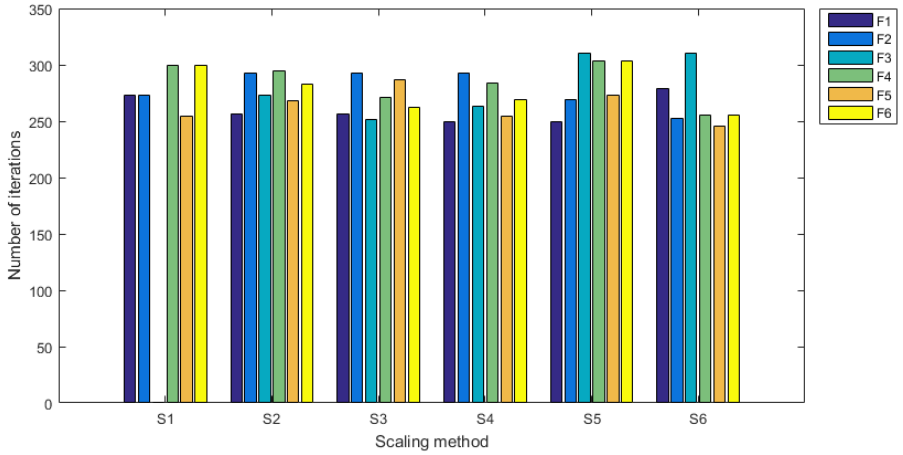
The higher the accuracy and the lower the number of iterations to solve the inverse problem, the better the proxy is ranked. The results from Tables 4.3-4.5 are supplemented by the bar plots from Figures 4.6-4.8. For each case, three bar plots are presented; namely, success rate, number of iterations and similarity to the known optimum solution from Table 4.1. Success rate is defined as the number of the inverse problems which its recovered target variable has percentage error of less than 1% to the generated actual target variable. The similarity of the global optimum solution is defined in percentage as the percentage error of the number of iterations required for the proxies to that for the optimal solution.

Table 4.3 and Figure 4.6 show the results of the KHEL case. Recall the optimal sensors for the KHEL case are 13, 38 and 48 as shown in Figure 4.1(b). These sensors are also presented in Table 4.3. However, we are not aiming to find proxies that exactly pick optimal solutions but rather proxies that perform well when solving the inverse problem. The difficulties of the three cases from the least to most challenging are KHEL, KHEH and KLEH. The difficulty measured by computing the averages of the accuracies obtained by testing random sensors on the test datasets over 100 times are 92%, 73% and 32%. Figure 4.6 shows that since the KHEL case is relatively well-posed, all the proxy combinations show high recovery success rates and high similarity to the global optimum solution except the F1S3 case which has zero success rates for all the modes tested. Note that the sensor variable ID number 24 and 60 appear frequently and they turn out to be U_y at the tip and ϵ_x at the root of the beam which may be helpful in recovering k_2 and E. Since the compliance of the beam for this case is relatively high, U_y is also more significant and sensitive information. In fact, for the same reason, most of the sensors became more responsive to the manipulations which made the accuracy of randomly chosen variables over 90%. The combinations of the objective functions and the scaling methods which offered 100% accuracies as well as over 90% similarities to the optimal solution are as follows: F1S4, F1S5, F3S3 and F5S6. Note that although this was an easy problem, the numbers of iterations of only four combinations were actually close to the optimal number of iterations.

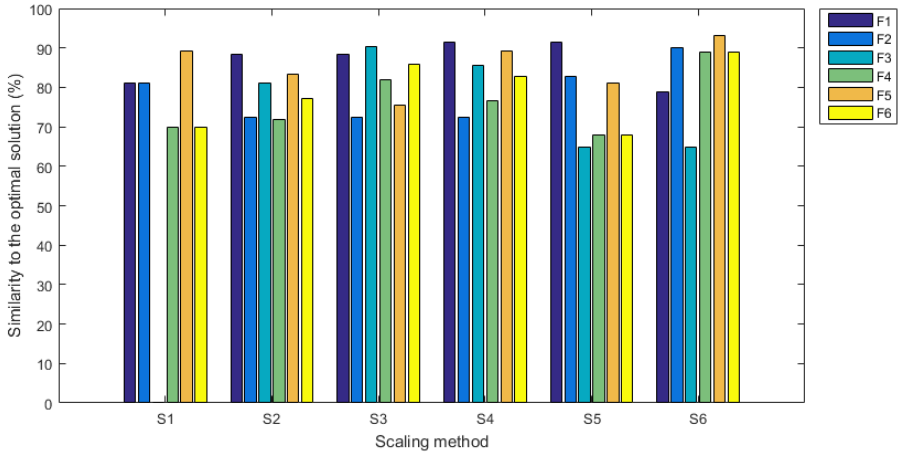
Consider Tables 4.4 which are the results of the KHEH case. Recall the optimal sensors for the KHEH case are 38, 48 and 62 as shown in Figure 4.1(c). These sensors are also



(a) Success rate



(b) Number of iterations



(c) Similarity to the optimal solution

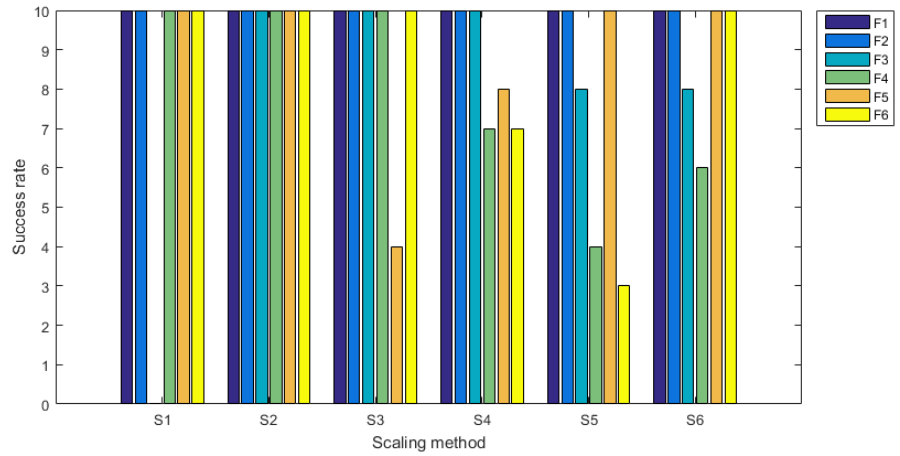
Figure 4.6: Performance of the objective functions and scaling methods on the KHEL case; (a) success rate, (b) number of iterations and (c) similarity to the optimal solution.

	Func.	F1			F2			F3			F4		F5		F6	
Scal.	Mode	A=1	A=2	A=3	A=1	A=2	A=3	A=1	A=2	A=3	A=2	A=3	A=2	A=3	A=2	A=3
S1	Sen.1	13	37	9	13	37	9	36	36	36	13	13	18	18	13	13
	Sen.2	23	73	23	23	73	23	48	48	48	23	23	37	37	23	23
	Sen.3	60	74	48	60	74	48	50	50	50	74	74	49	49	74	74
	Acc.	**10**	8	**10**	**10**	8	**10**	0	0	0	**10**	**10**	**10**	**10**	**10**	**10**
	Iter.	**297.3**	281.6	**253.6**	**297.3**	281.6	**253.6**	NaN	NaN	NaN	**307**	**307**	**265.9**	**265.9**	**307**	**307**
S2	Sen.1	37	37	49	48	37	17	17	17	23	37	48	13	13	36	50
	Sen.2	50	38	50	50	38	49	19	48	48	48	60	48	48	60	60
	Sen.3	60	49	60	60	49	50	60	60	60	60	74	74	74	74	74
	Acc.	6	0	**10**	**10**	0	9	6	7	**10**	**10**	**10**	**10**	**10**	**10**	8
	Iter.	272	NaN	**270.4**	**257.3**	NaN	273	345	313.4	**251.1**	**257.4**	**287.4**	**263.2**	**263.2**	**282.5**	296.8
S3	Sen.1	36	37	49	48	37	17	13	13	13	37	48	48	48	37	48
	Sen.2	50	38	50	50	38	49	62	48	48	48	60	60	60	60	60
	Sen.3	60	49	60	60	49	50	72	72	72	60	74	72	72	74	74
	Acc.	**10**	0	**10**	**10**	0	9	**10**	**10**	**10**	**10**	**10**	4	4	7	**10**
	Iter.	**270.9**	NaN	**270.4**	**257.3**	NaN	273	**291.5**	**264.7**	**264.7**	**257.4**	**287.4**	313.7	313.7	292.8	**287.4**
S4	Sen.1	17	36	49	36	37	17	36	37	37	37	37	37	37	37	37
	Sen.2	48	38	50	48	38	49	38	50	48	60	60	73	73	60	60
	Sen.3	60	49	60	60	49	50	48	60	60	74	74	74	74	74	74
	Acc.	7	0	**10**	**10**	0	9	0	6	**10**	7	7	8	8	7	7
	Iter.	313.4	NaN	**270.4**	**257.1**	NaN	273	NaN	272	**257.4**	292.8	292.8	281.6	281.6	292.8	292.8
S5	Sen.1	37	23	13	37	23	9	18	24	24	8	8	9	9	7	7
	Sen.2	49	37	37	49	36	23	24	36	36	10	10	37	37	8	8
	Sen.3	50	50	50	50	48	37	62	62	62	61	61	60	60	14	14
	Acc.	0	8	**10**	0	**10**	**10**	4	8	8	4	4	**10**	**10**	3	3
	Iter.	NaN	291.1	**278.4**	NaN	**246.5**	**253.4**	321.5	301.7	301.7	331	331	**287.9**	**287.9**	327.3	327.3
S6	Sen.1	7	36	9	23	23	9	18	24	24	7	7	14	14	10	10
	Sen.2	24	60	23	37	48	23	24	36	36	17	17	48	48	23	23
	Sen.3	72	74	37	48	60	37	62	62	62	61	61	74	74	61	61
	Acc.	**10**	**10**	**10**	**10**	**10**	**10**	4	8	8	6	6	**10**	**10**	**10**	**10**
	Iter.	**335.6**	**282.5**	**253.4**	**249.4**	**251.1**	**253.4**	321.5	301.7	301.7	326.5	326.5	**270.6**	**270.6**	**322.1**	**322.1**

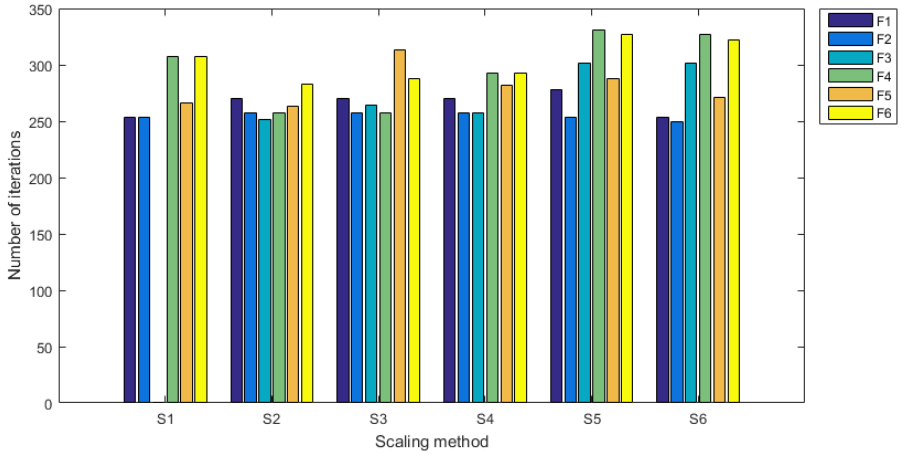
Table 4.4: Reduced sensor option: KHEH

presented in Table 4.4. By observing the success rates, it is noticed that the KHEH case is definitely more challenging than the previous case. The results of the various combinations are depicted on bar graphs shown in Figure 4.7. Note that F1 and F2 performed with the highest accuracies of all the scaling methods (see Figure 4.7(a)). These two proxy objective functions are the least sensitive to variable scaling of the data sets. The SVD based proxy objective functions (F3-F6) show noticeable changes in their performances for the various dataset scalings. Maximisation of singular values (F3) using S1 on the datasets gives poor results for all possible modes. The maximisation of the contents of the covariance matrix is not ideal without scaling since the sensor selection can be biased towards the measurements with larger orders of magnitudes such as σ sensor. The component norm normalisation (S2) led to success of all the objective functions. The objective functions and scaling methods that offer 100% accuracy as well as over 90% similarities to the optimal solution are F1S1, F1S6, all F2's, F3S2, F3S4, F4S2 and F4S3.

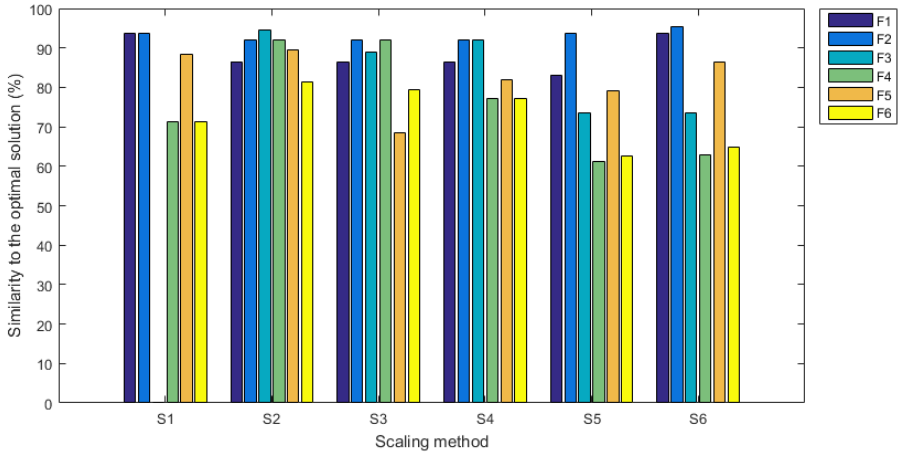
Lastly, consider Table 4.5 and Figure 4.8 which are the results of the KLEH case. It is observed that this is the most challenging case of all; the combinations hardly have a 10/10 accuracy result. The optimal sensors for this case were 48, 50 and 74 and they are easily found in the favourable combinations. The combinations which offer 100% accuracies and over 90% similarities to the optimal solution are F3S4, F4S1, F4S4, F4S5, F4S6, F6S1, F6S4, F6S5 and F6S6. It is interesting to note that none of the good combinations appeared commonly for all three cases. It should be made clear that the success rates are by far more important than the similarities of the numbers of iterations. Therefore, the success rates have been prioritised and any combinations with average success rates of at least $29/3 \approx 9.667$ are considered. These include F1S1, all F2's, F3S4, F4S2, F6S2 and F6S6. Note that all F5 combinations which aimed to maximise the magnitude of the regression coefficients failed as F2 is insensitive to the scaling methods. This implies there is a strong correlation between maximising the variance explained in Y in the SIMPLS output and the ability to solve an inverse problem. When the number of iterations is also considered F3S4 performed the best with over 80% similarity to the global optimum solution. A number of promising proxy candidates emerged from the 36 combinations considered for the beam problem. However,



(a) Success rate



(b) Number of iterations



(c) Similarity to the optimal solution

Figure 4.7: Performance of the objective functions and scaling methods on the KHEH case; (a) success rate, (b) number of iterations and (c) similarity to the optimal solution.

	Func.	F1			F2			F3			F4		F5		F6	
Scal.	Mode	A=1	A=2	A=3	A=1	A=2	A=3	A=1	A=2	A=3	A=2	A=3	A=2	A=3	A=2	A=3
S1	Sen.1	13	10	23	13	10	23	36	36	36	48	48	13	13	48	48
	Sen.2	23	23	48	23	23	48	48	48	48	49	49	23	23	49	49
	Sen.3	74	73	72	74	73	72	50	50	50	50	50	72	72	50	50
	Acc.	2	1	9	2	1	9	0	0	0	0	0	2	2	0	0
	Iter.	358	369	264.2	358	369	264.2	NaN	NaN	NaN	NaN	NaN	361.5	361.5	NaN	NaN
S2	Sen.1	37	10	14	37	10	23	48	48	23	23	23	7	7	23	23
	Sen.2	38	36	37	38	36	37	60	60	48	38	48	37	37	38	48
	Sen.3	50	73	61	48	73	72	74	74	74	48	50	61	61	48	50
	Acc.	0	3	1	0	3	**10**	7	7	8	**10**	**10**	1	1	**10**	**10**
	Iter.	NaN	264.6	335	NaN	264.6	**266.5**	274.4	274.4	270.8	**211.1**	**207.6**	317	317	**211.1**	**207.6**
S3	Sen.1	37	8	14	37	8	23	13	13	48	23	48	23	23	13	48
	Sen.2	38	14	37	38	14	37	48	48	72	48	72	72	72	23	72
	Sen.3	50	49	61	48	49	72	72	72	73	74	74	73	73	48	74
	Acc.	0	0	1	0	0	**10**	7	7	1	8	5	2	2	7	5
	Iter.	NaN	NaN	335	NaN	NaN	**266.5**	281.1	281.1	370	270.8	262	364	364	285.8	262
S4	Sen.1	37	36	14	36	38	23	36	36	37	13	37	23	23	13	37
	Sen.2	49	38	37	38	49	37	38	48	48	23	48	72	72	23	48
	Sen.3	50	49	61	48	50	72	48	74	74	48	74	73	73	48	74
	Acc.	0	0	1	0	0	**10**	0	**10**	**10**	7	**10**	2	2	7	**10**
	Iter.	NaN	NaN	335	NaN	NaN	**266.5**	NaN	**216.9**	**216.4**	285.8	**216.4**	364	364	285.8	**216.4**
S5	Sen.1	37	13	14	37	17	23	18	18	18	23	23	36	36	23	23
	Sen.2	49	50	37	38	18	37	24	24	24	36	36	48	48	36	36
	Sen.3	50	74	61	48	36	72	62	62	62	50	50	50	50	50	50
	Acc.	0	6	1	0	0	**10**	0	0	0	**10**	**10**	0	0	**10**	**10**
	Iter.	NaN	274.6	335	NaN	NaN	**266.5**	NaN	NaN	NaN	**224.6**	**224.6**	NaN	NaN	**224.6**	**224.6**
S6	Sen.1	37	18	14	37	10	23	18	18	18	23	23	36	36	23	23
	Sen.2	49	20	37	38	48	37	24	24	24	37	37	48	48	37	37
	Sen.3	50	49	61	48	73	72	62	62	62	49	49	50	50	49	49
	Acc.	0	0	1	0	3	**10**	0	0	0	**10**	**10**	0	0	**10**	**10**
	Iter.	NaN	NaN	335	NaN	293.6	**266.5**	NaN	NaN	NaN	**225.7**	**225.7**	NaN	NaN	**225.7**	**225.7**

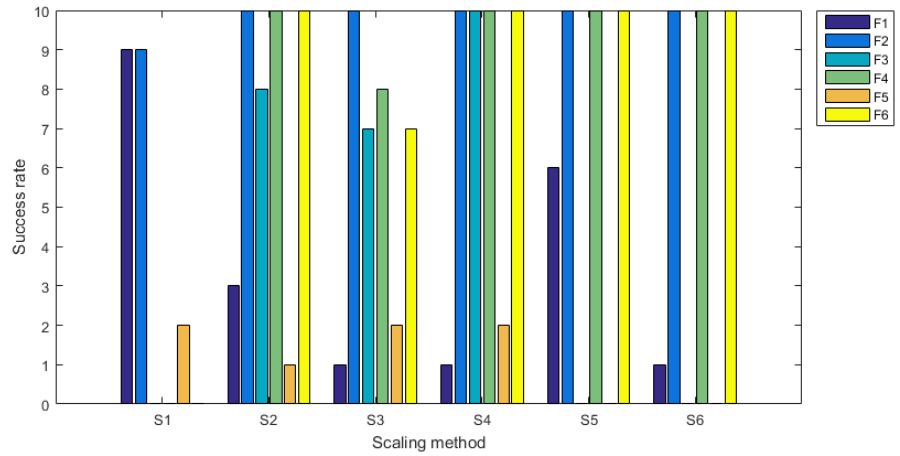
Table 4.5: Reduced sensor option: KLEH

it is necessary to conduct additional studies using a larger number of problems to ensure that general and not problem specific conclusions are drawn. This is however, not within the scope of this study.

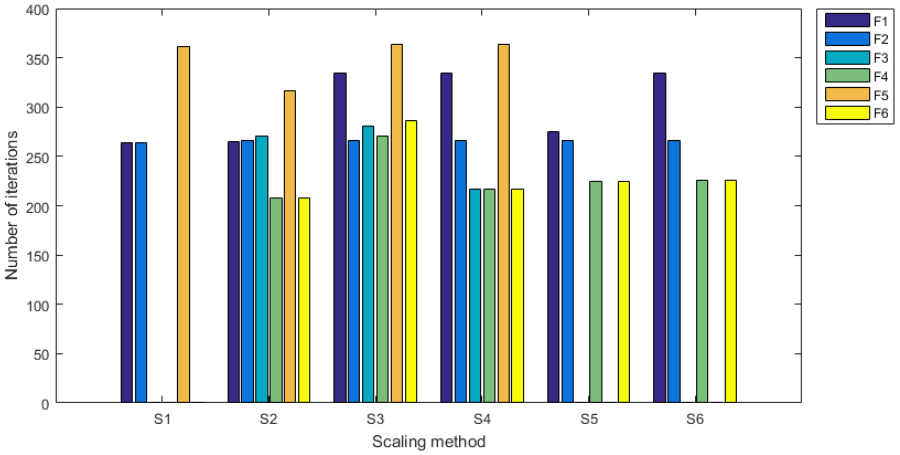
There are three important observations to be made from the observed sensor ID numbers listed in Tables 4.3-4.5. First of all and most importantly, the only difference between the cases is the ranges of E and k. However, the variable selections become noticeably distinctive. When E becomes significantly larger than the k values (KLEH), the deformation of the beam due to the point load becomes mainly dependent on the E values and not so much on k values. This means the objective functions would attempt to find sensors which will be able to recover the insensitive k as accurately as possible by choosing the most responsive variables to the small variations of the k. Since the importance of recovering E and k is not biased, it will be noticed later that sensors tend to be placed to recover less sensitive measurement variables.

Secondly, regardless of scaling methods, objective functions and A, some sensors occur more often than others. For instance, consider the following values in Tables 4.4: 23 (u in X-direction at the tip), 48 (σ in X-direction at the tip) and 60 (ϵ in X-direction at the root) and the presence of the first two and the last sensors seem appropriate for predicting k_1 and E, respectively. Numerous times the placed sensors are not obvious, but if they are shown to perform well then the benefits of SPO are evident. Note that since this is an optimisation problem, it is known that each variable explains what the objective functions ask for and it tries to keep the information from each variable separate and unique.

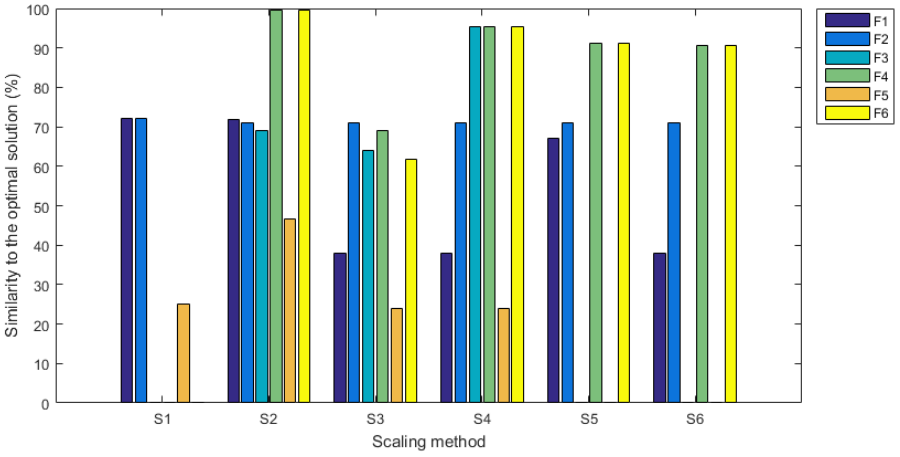
Lastly, the same sensor pairs are found using different scaling and proxy objective functions. The results for $F1$ and $F2$ are often identical. In particular, the solutions of mean centering (S1) for A from 1 to 3 are identical. Note that the chosen variables of $F2$ for $A = 3$ for the KLEH case (see Table 4.5) are almost all identical regardless of the chosen dataset scaling. This demonstrates the importance of the dataset scaling and that effective scaling depends on the problem characteristics at hand.



(a) Success rate



(b) Number of iterations



(c) Similarity to the optimal solution

Figure 4.8: Performance of the objective functions and scaling methods on the KLEH case; (a) success rate, (b) number of iterations and (c) similarity to the optimal solution.

Chapter 5

Mode-Based Formulations: Model Error Investigation

The aim of this chapter is to investigate how the formulations react to models with distinct model errors. The model errors were introduced by changing the sizes of meshes: (1) coarse mesh (1 by 2 has been considered in the previous chapters), (2) medium mesh (2 by 4) and (3) fine mesh (4 by 8) (see Figure 5.1). Changing the mesh sizes allows for the model error to be varied as well as the number of sensors to be increased spatially, should we want to keep placing them at the integration points of the finite elements. Although the stress, strain and displacement can be evaluated at any spatial location, the developed software can efficiently compute it at the integration points and nodes. Hence, for this study we restrict ourselves at placing sensors at the integration points and nodes. The numbers of possible sensor options for coarse, medium and fine meshes are 74, 266 and 1010, respectively. The sensor options in this chapter for the coarse model have all 74 sensors available. The aim is to investigate on whether the sensors converge spatially as the mesh is refined. However, this is only possible if the spatial resolution of the stress, strain and displacement fields starts to converge (i.e. model error has decreased enough to allow for the spatial resolution of sensitive domains to be resolved accurately enough for sensors to converge spatially). In addition, a denser distribution of sensors requires a better resolution of the areas of interest to warrant convergence amongst sensors to be placed as more integration points are available to place sensors.

The various mode-based formulations were tested on the same three cases (KHEH, KHEL and KLEH) as in the previous chapter on 100 problems for coarse mesh with 74 sensors to select from. The results of the KHEH, KHEL and KLEH cases are shown in Tables 5.1-5.3. The accuracies (success rates) in the tables are obtained by counting the inverse problems which its percentage errors of the recovered variables being less than 1%. Therefore, the maximum accuracy is 100 since there are 100 test problems. The average accuracy for randomly chosen variables for the KHEL, KHEH and KLEH is 89%, 71% and 32%, respectively. As mentioned in the previous chapter, although there are two criteria for scoring the proxies, the accuracy results which drive the robustness of the methods are considered to be more critical than the numbers of iterations. Since the reduced sensor models in the previous chapter are identical to the coarse mesh models in this chapter and the only difference is the number of sensor options, if the methods worked in the previous chapter with the reduced sensor options, it should definitely work as well with the coarse model in this chapter. The same six objective functions and the scaling methods were tested as presented in Table 4.2. Notice that the same phenomenon as discussed in Section 4 is also observed in Tables 5.1-5.3 with a larger number of available sensors and test problems. The proxy functions and dataset scaling pairs which resulted in accuracies of at least 90% in both the reduced sensors options and the full sensor options (coarse mesh) are as follows:

- F1S1: $\arg\min_x(\|Y_{residuals}\|)$ with component mean centering.

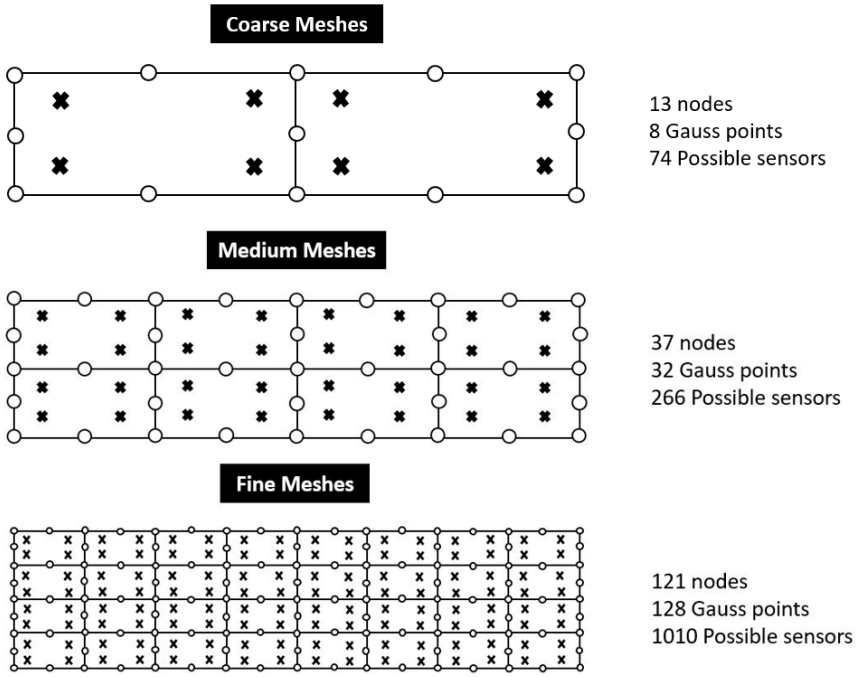


Figure 5.1: Coarse, medium and fine meshes.

	Func.	F1			F2			F3			F4		F5		F6	
Scal.	Mode	A=1	A=2	A=3	A=1	A=2	A=3	A=1	A=2	A=3	A=2	A=3	A=2	A=3	A=2	A=3
S1	Sen.1	23	11	22	23	11	22	27	27	27	17	17	23	23	17	17
	Sen.2	65	23	35	65	23	35	30	30	30	22	22	53	53	22	22
	Sen.3	73	37	41	73	37	41	33	33	33	25	25	73	73	25	25
	Acc.	99	99	73	99	99	73	0	0	0	**100**	**100**	87	87	**100**	**100**
	Iter.	254.17	247.91	321.67	254.17	247.91	321.67	NaN	NaN	NaN	**286.25**	**286.25**	365.33	365.33	**286.25**	**286.25**
S2	Sen.1	23	22	23	31	11	21	30	15	22	21	22	13	26	24	11
	Sen.2	49	37	25	37	31	37	36	49	49	22	41	47	49	25	23
	Sen.3	50	48	37	50	37	60	49	65	72	48	54	59	65	31	30
	Acc.	98	**100**	**100**	0	93	97	0	98	**100**	**100**	83	**100**	98	99	**100**
	Iter.	292.15	**252.93**	**257.2**	NaN	307.61	305.92	NaN	271.86	**274.09**	**264.96**	294.48	**267.46**	271.09	270.61	**247.63**
S3	Sen.1	18	22	23	31	11	21	60	49	23	15	9	13	15	36	22
	Sen.2	41	37	25	37	31	37	64	54	25	19	26	27	22	41	23
	Sen.3	49	48	37	50	37	60	70	64	49	26	49	52	49	65	49
	Acc.	**100**	**100**	**100**	0	93	97	98	99	**100**	**100**	98	**100**	90	99	99
	Iter.	**260.22**	**252.93**	**257.2**	NaN	307.61	305.92	323.11	286.93	**257.52**	**287.01**	276.57	**268.09**	295.68	262.2	260.88
S4	Sen.1	12	22	23	31	11	21	31	11	11	19	9	23	11	19	11
	Sen.2	32	40	25	37	31	37	37	31	31	22	26	48	18	22	48
	Sen.3	34	48	37	50	37	60	50	37	37	54	37	65	48	54	61
	Acc.	**100**	**100**	**100**	0	93	97	0	93	93	99	99	**100**	**100**	99	**100**
	Iter.	**265.66**	**260.96**	**257.2**	NaN	307.61	305.92	NaN	307.61	307.61	280.16	270.44	**251.44**	**272.38**	280.16	**261.12**
S5	Sen.1	11	48	11	11	11	26	27	27	27	7	7	16	11	7	22
	Sen.2	45	64	42	28	31	35	33	33	33	10	22	17	36	8	25
	Sen.3	73	68	45	34	37	41	62	62	62	31	36	46	46	31	36
	Acc.	99	**100**	0	**100**	93	71	**100**	**100**	**100**	97	97	99	**100**	97	93
	Iter.	281.77	**270.67**	NaN	**272.88**	307.61	301.19	**272.67**	**272.67**	**272.67**	300.14	293.15	276.59	**261.4**	291.19	297.05
S6	Sen.1	32	11	24	47	11	24	27	27	27	7	10	24	7	8	10
	Sen.2	68	31	35	50	31	35	33	33	33	31	19	35	36	28	22
	Sen.3	73	37	41	65	37	41	34	62	62	60	48	70	40	64	29
	Acc.	**100**	93	69	**100**	93	69	0	**100**	**100**	99	**100**	98	99	99	99
	Iter.	**279.62**	307.61	294.2	**261.29**	307.61	294.2	NaN	**272.67**	**272.67**	263.43	**257.41**	279.97	256.14	282.44	277.61

Table 5.1: Chosen variables for high k's & low E

	Func.	F1			F2			F3			F4		F5		F6	
Scal.	Mode	A=1	A=2	A=3	A=1	A=2	A=3	A=1	A=2	A=3	A=2	A=3	A=2	A=3	A=2	A=3
S1	Sen.1	23	37	9	23	37	9	27	27	27	13	13	18	18	13	13
	Sen.2	54	65	23	54	65	23	33	33	33	23	23	43	43	23	23
	Sen.3	64	69	48	64	69	48	36	36	36	55	55	47	47	55	55
	Acc.	97	92	**100**	97	92	**100**	0	0	0	**100**	**100**	95	95	**100**	**100**
	Iter.	307.78	282.68	**259.75**	307.78	282.68	**259.75**	NaN	NaN	NaN	**289.51**	**289.51**	275.8	275.8	**289.51**	**289.51**
S2	Sen.1	44	31	30	50	31	17	42	39	23	30	44	10	10	37	39
	Sen.2	50	37	35	54	37	30	45	45	39	36	55	48	48	44	54
	Sen.3	54	50	60	55	50	35	48	54	54	55	69	63	63	55	55
	Acc.	99	0	10	83	0	16	1	98	**100**	**100**	97	**100**	**100**	96	73
	Iter.	266.39	NaN	282.9	296.39	NaN	276.87	216	269.32	**259.79**	**266.98**	291.59	**278.88**	**278.88**	253.74	299.05
S3	Sen.1	48	31	30	44	31	17	66	39	39	37	37	28	39	37	39
	Sen.2	54	37	35	50	37	30	68	66	63	50	50	66	63	50	54
	Sen.3	64	50	60	54	50	35	72	68	66	55	55	68	71	55	55
	Acc.	98	0	10	99	0	16	**100**	52	92	75	75	52	30	75	73
	Iter.	288.39	NaN	282.9	266.39	NaN	276.88	**304.44**	303.8	290.24	273.24	273.24	298.67	304.83	273.24	299.05
S4	Sen.1	48	30	30	48	28	17	54	31	37	37	37	37	37	55	37
	Sen.2	54	35	35	54	30	30	60	37	54	54	54	56	56	66	54
	Sen.3	64	41	60	55	35	35	64	54	64	55	55	69	69	74	55
	Acc.	98	0	10	98	0	16	21	56	25	79	79	99	99	93	79
	Iter.	288.39	NaN	282.9	287.34	NaN	276.87	333	286.26	305.52	291.93	291.93	277.75	277.75	290.01	291.93
S5	Sen.1	31	23	13	31	23	9	27	27	27	48	50	8	45	48	50
	Sen.2	37	27	31	37	27	23	33	33	33	67	67	62	70	67	67
	Sen.3	50	48	50	50	48	37	57	57	57	70	70	68	74	70	70
	Acc.	0	**100**	81	0	**100**	**100**	**100**	**100**	**100**	98	91	58	99	98	91
	Iter.	NaN	**261.02**	282.14	NaN	**261.02**	**260.97**	**280.65**	**280.65**	**280.65**	282.79	287.31	312.72	286.96	282.79	287.31
S6	Sen.1	21	19	9	33	40	9	27	26	26	42	10	48	48	42	50
	Sen.2	26	27	23	36	48	23	33	27	27	65	39	65	65	65	69
	Sen.3	43	54	37	54	64	37	34	33	33	72	54	74	74	72	70
	Acc.	60	71	**100**	93	99	**100**	0	**100**	**100**	99	79	**100**	**100**	99	99
	Iter.	302.66	296.92	**260.97**	280.47	261.49	**260.97**	NaN	**273.2**	**273.2**	282.45	298.31	**274.7**	**274.7**	282.45	290.63

Table 5.2: Chosen variables for high k's & high E

	Func.	F1			F2			F3			F4		F5		F6	
Scal.	Mode	A=1	A=2	A=3	A=1	A=2	A=3	A=1	A=2	A=3	A=2	A=3	A=2	A=3	A=2	A=3
S1	Sen.1	13	13	23	13	13	23	27	27	27	43	43	51	51	43	43
	Sen.2	23	23	48	23	23	48	33	33	33	48	48	61	61	48	48
	Sen.3	63	58	71	63	58	71	36	36	36	50	50	73	73	50	50
	Acc.	35	20	90	35	20	90	0	0	0	0	0	1	1	0	0
	Iter.	353.25	357.75	269.25	353.25	357.75	269.25	NaN	NaN	NaN	NaN	NaN	246	246	NaN	NaN
S2	Sen.1	31	50	14	37	10	23	42	42	39	23	39	7	7	23	39
	Sen.2	37	67	37	38	36	37	46	48	48	38	48	37	37	38	48
	Sen.3	50	71	61	48	73	72	48	63	63	48	63	61	61	48	63
	Acc.	0	12	7	0	32	**100**	1	**100**	**100**	**100**	**100**	6	6	**100**	**100**
	Iter.	NaN	284.33	310.14	NaN	281.78	**261.31**	186	**212.05**	**211.17**	**206.91**	**211.17**	298	298	**206.91**	**211.17**
S3	Sen.1	31	21	14	37	7	23	39	39	39	23	23	23	23	13	39
	Sen.2	37	41	37	38	35	37	42	42	48	48	39	56	56	23	48
	Sen.3	50	60	61	48	57	72	45	66	66	63	48	71	71	48	63
	Acc.	0	5	7	0	2	**100**	0	**100**	**100**	97	**100**	31	31	88	**100**
	Iter.	NaN	312.2	310.14	NaN	313	**261.31**	NaN	**221.85**	**209.33**	263.82	**210.23**	348.12	348.12	274.56	**211.17**
S4	Sen.1	31	30	14	40	28	23	40	40	37	13	37	23	23	23	37
	Sen.2	37	35	37	42	30	37	42	48	48	23	48	56	56	48	48
	Sen.3	50	41	61	48	35	72	48	63	63	48	63	71	71	66	63
	Acc.	0	0	7	1	0	**100**	1	**100**	**100**	88	**100**	31	31	86	**100**
	Iter.	NaN	NaN	310.14	160	NaN	**261.31**	160	**211.72**	**204.61**	274.56	**204.61**	348.12	348.12	274.73	**204.61**
S5	Sen.1	31	23	14	31	48	23	24	24	24	23	23	32	32	23	23
	Sen.2	37	49	37	40	51	37	26	26	26	28	28	33	33	28	28
	Sen.3	50	66	67	48	61	72	51	51	51	39	39	49	49	39	39
	Acc.	0	98	7	0	5	**100**	0	0	0	**100**	**100**	0	0	**100**	**100**
	Iter.	NaN	263.89	310.14	NaN	313.2	**261.31**	NaN	NaN	NaN	**235.82**	**235.82**	NaN	NaN	**235.82**	**235.82**
S6	Sen.1	31	8	14	31	38	23	24	24	24	23	23	32	32	23	23
	Sen.2	37	42	37	40	58	37	26	26	26	35	29	33	33	46	40
	Sen.3	50	73	61	48	67	72	51	51	51	48	39	49	49	49	48
	Acc.	0	37	7	0	9	**100**	0	0	0	**100**	**100**	0	0	**100**	**100**
	Iter.	NaN	276.97	310.14	NaN	282.66	**261.31**	NaN	NaN	NaN	**209.05**	**224.06**	NaN	NaN	**230.98**	**212.53**

Table 5.3: Chosen variables for low k's & high E

- F2S1: $\arg\max_x(varY)$ with component mean centering.
- F2S3: $\arg\max_x(varY)$ with component mean centering and mean normalisation.
- F2S4: $\arg\max_x(varY)$ with component mean centering and SD normalisation.
- F2S5: $\arg\max_x(varY)$ with vector mean centering and mean normalisation.
- F2S6: $\arg\max_x(varY)$ with vector mean centering and SD normalisation.
- F3S2: $\arg\max_x(\sum S_{xy})$ with component mean centering and norm normalisation.
- F4S2: $\arg\max_x\left(\frac{\sum S_{xy}}{\|S_{xy}\|}\right)$ with component mean centering and norm normalisation.
- F6S2: $\arg\max_x\left(\frac{\sum S_{xy}}{\|S_{xy}\|}\right)^{(\|U_x^T U_{xy}\|)}$ with component mean centering and norm normalisation.
- F6S6: $\arg\max_x\left(\frac{\sum S_{xy}}{\|S_{xy}\|}\right)^{(\|U_x^T U_{xy}\|)}$ with vector mean centering and SD normalisation.

Note that F3S4 which performed best for the reduced sensor case is not listed since the smaller number of test inverse problems in the previous chapter was not sufficient to prove the robustness of the proxies. It is interesting to note that the most robust objective function so far is the F2 which is maximisation of variance explained in the target data using PLS and this conclusion can be made since it is less sensitive to the scaling methods. The only scaling method which does not work well with F2 is S2 which works well with F3, F4 and F6. Note that the sensors obtained for both F1S1 and F2S1 always are equivalent.

Exhaustive combinatorial approach was implemented to search for the optimal sensors for medium meshes using the ten identified methods and it turned out that F3S2, F4S2, F6S2 and F6S2 failed to recover E and k's with all three cases with higher success rates than randomly chosen sensors. This means that only the two objective functions using SIMPLS (F1 and F2) manage to find useful sensors. While the F1 only performs well with S1: component mean centering, F2 seems to work well with almost any scaling methods. Tables 5.4 - 5.6 show the selected sensors and results of the two objective functions. Notice that the quality of the selected sensors is highly dependent on the number of modes. It must be strongly emphasized that although mesh size reduction definitely increases the opportunities for choosing better sensors, it also allows for a better resolution of the spatial stress, strain and displacement fields. Therefore, the results of the medium meshes are not necessarily better than the coarse meshes but definitely more relevant. The proxies should perform better on average than randomly chosen sensors. Consequently, only the two objective functions are considered for further analyses.

The F2S3 results in Table 5.4 show that when only a single mode is used the accuracy is zero but as the number of modes, A increases, the accuracy gets higher and the number of iterations tend to decrease. However, for the more challenging problems in Tables 5.5 and 5.6, we note that up to two modes perform the best overall, with three modes performing well as the exception. As the modes increase and the performance breaks down it can happen that an additional mode improves the performance as for example with F1S1 in Table 5.6. This indicates that the second mode poorly approximated the inverse problem as F1 is based on $Y_{residuals}$, while the third mode introduced more relevant information. The accuracies (robustness) of the methods in order of magnitude beginning with the highest are as follows: F2S3 and F2S6 are tied at 99.67%, F2S5 with 98.67%, F1S1 and F2S1 with 98% and lastly,

	Func.	F1			F2		
Scal.	Mode	A=1	A=2	A=3	A=1	A=2	A=3
S1	Sen.1	21	29	58	21	29	58
	Sen.2	29	47	166	29	47	166
	Sen.3	264	167	266	264	167	266
	Acc.	97	**100**	**100**	97	**100**	**100**
	Iter.	261.55	**253.86**	**248.22**	261.55	**253.86**	**248.22**
S3	Sen.1				111	68	13
	Sen.2				129	153	70
	Sen.3				168	238	163
	Acc.				0	98	**100**
	Iter.				NaN	271.97	**242.46**
S4	Sen.1				13	13	13
	Sen.2				135	35	70
	Sen.3				170	166	163
	Acc.				**100**	**100**	**100**
	Iter.				**248.39**	**234.25**	**242.46**
S5	Sen.1				72	68	13
	Sen.2				96	80	70
	Sen.3				117	238	90
	Acc.				97	97	**100**
	Iter.				284.99	266.66	**249.87**
S6	Sen.1				138	68	72
	Sen.2				180	82	166
	Sen.3				264	264	265
	Acc.				**100**	**100**	98
	Iter.				**265.85**	**251.45**	242.48

Table 5.4: Medium mesh - High k's & Low E

F2S3 with 93.67%. When considering the number of iterations, F2S3 is slightly better than F2S6.

Note it has been shown that all the six methods provide successfully recovered target variables over 90% of the time. An additional test to quantify the proxy performance is to see how each proxy evaluates the other proxy solutions that we know perform well. The proxy objective function comparisons are presented in Figure 5.2 when the optimal sensor locations obtained by five of the six sensors are evaluated using the proxy function of the sixth sensor. The percentage errors that indicate the percentage difference between the two proxies were computed as follows:

$$PE = 100(\frac{FV - FV_{min}}{FV_{max} - FV_{min}}), \tag{5.1}$$

where FV, FV_{min} and FV_{max} denote the current function value of one proxy and the minimum and maximum possible function values of another proxy, respectively. Figure 5.2(b) depicts the percentage error when F2S1, F2S3, F2S4, F2S5 and F2S6 are evaluated with their optimal sensor locations and how they differ when the same sensor locations are evaluated using F1S1. A percentage error of 100% implies that sets of sensors deemed optimal by one proxy function scaling combination are deemed the worst sensor pair by another proxy function scaling combination. A percentage error of 0% implies that both proxy function scaling combination deem the same sensor pair optimal.

Remember that for mean centering dataset scaling S1, the sensors obtained from F1 and F2 are identical (i.e. 0% percentage error). It is noticed that some proxy function dataset scaling combinations evaluate valuable sensor combinations as poor. For the KHEH case shown in Figure 5.2(b), F2S3 and F2S6 evaluate optimal sensors from F2S4 and F2S5 as almost their worst. The KHEL case (Figure 5.2(a)) shows that most proxy function dataset scaling combinations evaluate the others' optimal solution mostly within 10% percentile. Only F2S4 and F2S5 evaluate the optimal solutions from F1S1 and F2S1 as inferior. Similarly for the KLEH case (Figure 5.2(c)), the proxy function dataset scaling combinations F2S5 and F2S6 evaluate the optimal solution from F2S4 as inferior. Since it is already known that

	Func.	F1			F2		
Scal.	Mode	A=1	A=2	A=3	A=1	A=2	A=3
S1	Sen.1	72	45	127	72	45	127
	Sen.2	74	145	134	74	145	134
	Sen.3	145	205	265	145	205	265
	Acc.	34	**100**	1	34	**100**	1
	Iter.	340.2	**250**	189	340.2	**250**	189
S3	Sen.1				99	86	136
	Sen.2				149	133	142
	Sen.3				199	162	218
	Acc.				99	0	0
	Iter.				260.4	NaN	NaN
S4	Sen.1				166	86	136
	Sen.2				198	133	142
	Sen.3				228	162	218
	Acc.				95	0	0
	Iter.				275.4	NaN	NaN
S5	Sen.1				123	13	166
	Sen.2				151	166	199
	Sen.3				233	202	266
	Acc.				94	94	96
	Iter.				271.7	276.2	265.7
S6	Sen.1				81	132	166
	Sen.2				144	220	199
	Sen.3				198	243	266
	Acc.				99	98	96
	Iter.				261.6	273.9	265.7

Table 5.5: Medium mesh - High k's & High E

	Func.	F1			F2		
Scal.	Mode	A=1	A=2	A=3	A=1	A=2	A=3
S1	Sen.1	13	51	57	13	51	57
	Sen.2	247	71	138	247	71	138
	Sen.3	261	210	264	261	210	264
	Acc.	58	2	94	58	2	94
	Iter.	314.5	386.5	262.14	314.5	386.5	262.14
S3	Sen.1				145	166	17
	Sen.2				166	168	25
	Sen.3				247	266	168
	Acc.				**100**	**100**	1
	Iter.				**191.12**	**191.78**	531
S4	Sen.1				166	145	17
	Sen.2				247	247	25
	Sen.3				261	261	168
	Acc.				9	86	1
	Iter.				265.33	258.43	531
S5	Sen.1				166	160	17
	Sen.2				169	166	25
	Sen.3				247	247	168
	Acc.				**100**	**100**	1
	Iter.				**195.44**	**191.82**	531
S6	Sen.1				166	160	17
	Sen.2				169	166	25
	Sen.3				247	247	168
	Acc.				**100**	**100**	1
	Iter.				**195.44**	**191.82**	531

Table 5.6: Medium mesh - Low k's & High E

Mesh size	Acc.	Iter.	Mesh size	Acc.	Iter.
Coarse	100	259.75			
Coarse to Medium	100	270.21	Medium to coarse	100	280.13
Medium	100	249.96			
Medium to Fine	93	264.04	Fine to medium	82	347.79
Fine	100	316.45			

Table 5.7: Success rates and numbers of iterations for F1S1 and F2S1

the optimal sensor sets chosen from all the proxies performed well, should a proxy function dataset scaling combination consider the optimal sensors of other proxy function dataset scaling combinations to be within its low percentage error range, one could assume that it is better correlated with the inverse problem. To quantify the general alignment with the inverse problem, the percentage errors from all three cases were summed up and the average percentage errors produced as shown in Figure 5.3. The only proxy function dataset scaling combination that generalises poorly is F2S6. F2S6 is the best performing sensor with 99.67% accuracy. F2S6 generalises poorly for the inverse problem under consideration, which indicates some potential strong biases to preferred solution. For the remainder of the study we only consider the proxy function dataset scaling combinations that generalise well for the inverse problem which are the remaining five combinations F1S1, F2S1, F2S3, F2S4 and F2S5.

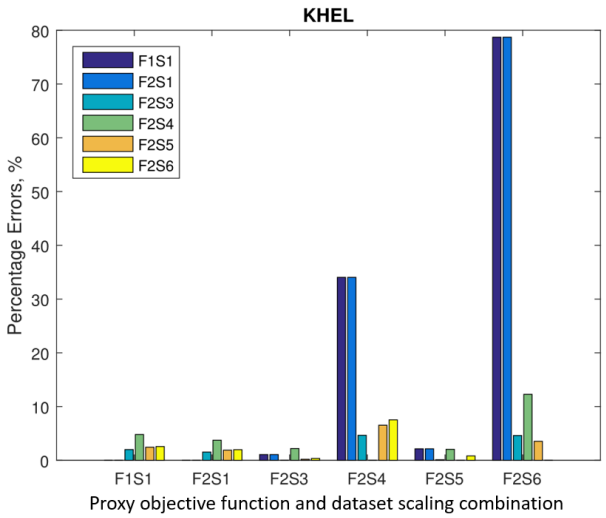
5.1 Model error test

As the meshes are refined the discretisation error decreases, where the discretisation error is indicative of model error in this study. Each model with an associated mesh (i.e. coarse, medium or fine, is considered a different problem). We investigate to what extent the optimal sensors from one model perform well on another model. For small model errors between models, it is expected that the sensors converge spatially. In addition for small model error differences, it is expected that the finer mesh model that has more sensors to choose from, to perform better than the limited selection from a coarser model. The five robust methods selected from the previous section (F1S1, F2S1, F2S3, F2S4 and F2S5) were examined on the various models by applying the optimal sensors found in one model to another model. Since the FEM displacement, stress and strain fields are spatially fully described using interpolation, obtaining a displacement, stress or strain measurement at any spatial location on the structure is possible.

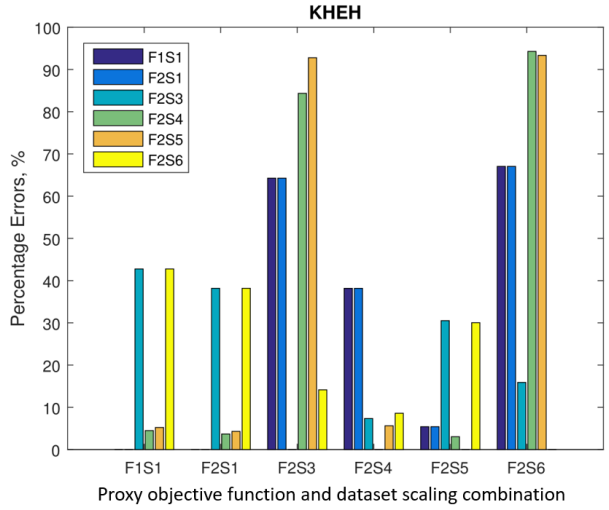
The number of the possible sensors in the fine mesh beam is 1010 which results in 171×10^6 possible combinations as compared to 64824 and 3.1×10^6 combinations in the coarse and medium mesh models, respectively. To keep the problem computationally tractable, the BGA (binary genetic algorithm) was used instead of the combinatorial approach for the fine mesh model. This also means it is not fully guaranteed that the optimal solution for each proxy function dataset scaling combination is obtained for the fine mesh. Hence, the BGA was run ten times for each case and the best results among them were used. Note that each run of the BGA was set to have 100 iterations, 30000 populations, the selection rate of 30% and the extremely high mutation rate of 80%, which may be suitable for multi-modal problems. This can be considered as an exhaustive BGA approach due to the large numbers of iterations and populations. The results from the coarse, medium and fine meshes for all the combinations of the robust methods are presented as well as the results from interchanging the sensors between the models. For this study, only the KHEH case was considered.

5.1.1 F1S1 and F2S1

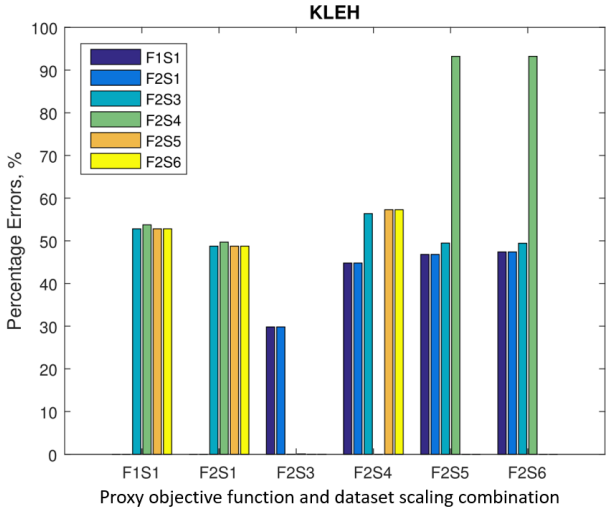
Recall that F1S1 and F2S1 give the same results, although they have two distinct objective functions as shown in Table 5.7. We, therefore, only elaborate on F1S1 which is directly



(a) The percentage error for KHEL



(b) The percentage error for KHEH



(c) The percentage error for KLEH

Figure 5.2: The percentage error between the optimal proxy values of five proxy function dataset scaling combinations and the optimal proxy value when the optimal designs are evaluated by the sixth proxy function dataset scaling combination for (a) KHEL, (b) KHEH, and (c) KLEH.

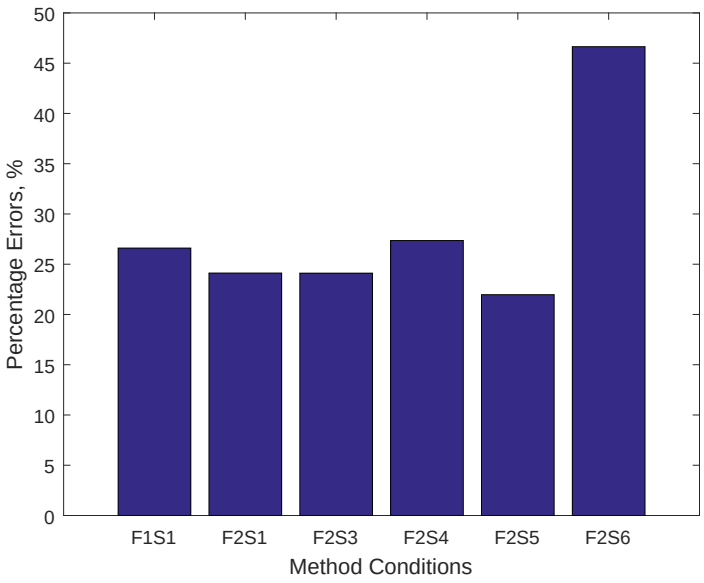


Figure 5.3: Average percentage errors.

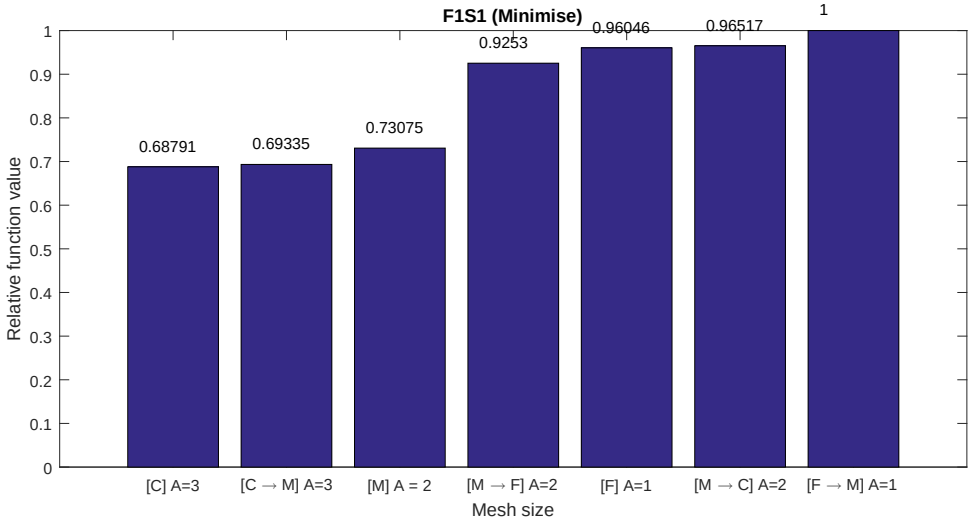


Figure 5.4: Function values of F1S1 (Minimisation problem).

applicable to F2S1. The success rates and the numbers of iterations for the sensors chosen in the distinct mesh sizes are shown in Table 5.7. When the sensors found from one mesh size were applied to another mesh size, it is indicated using the word "to". It is critical to note again that as the mesh gets finer, more sensors are available. At the same time, the model error is reduced. Thus, if the model errors have not converged yet, the optimal sensor placement may vary depending on whether the sensitivity of the information has converged or not. It is anticipated that the optimal sensors from a coarse mesh perform worse than a finer mesh due to the limited options for sensors. From Table 5.7, the numbers of iterations increase and the success rates decrease when optimal sensors from the coarse mesh are applied to the medium mesh and from the medium relocated to the fine mesh. On the other hand, when the optimal sensors from the finer meshes are applied to a coarser mesh, an improvement is only expected if the model errors have converged. Both sets of optimal sensors from the medium to coarse mesh and from the fine to medium mesh show that again the success rates decreased and the numbers of iterations increased. It is indicative that there is a significant development in the complexity of the fine mesh model.

Figure 5.4 shows the function values of the sensor combinations for the different mesh models (For F2S1, consider Figure 5.5. The only difference between the two figures is that F2S1 is a maximisation problem. Hence, it shows exactly the opposite behaviour from that of F1S1). Note that they are all scaled by one single value so that the largest value becomes 1. It is critical to note that each model has its optimal numbers of modes, A based on the performances. Therefore, A used for one model might not be the same as the A used for the other models as F1S1 clearly shows. Each model has its own different optimal A which are 3, 2 and 1 for coarse, medium and fine, respectively. This means that when a set of sensors from one model is transferred to the other model, it is not possible to compare the function value of the transferred sensors with that of the target model. For instance, one can not compare the function value of "from medium to fine" [M→F] with that of the fine model due to the difference in A. The reason is that the explanatory information of the measurement is different when the number of modes A varies. However, it is still meaningful to compare the function values of the transferred set with the models from which they were transferred as the number of modes A stay the same. The model error, however, changes. The function values increase which indicates an anticipated worse performance. To clarify, the function value at the coarse mesh model is lower than the "from coarse to medium"[C→M]. In addition, the "from medium to fine" and "from fine to medium" function values are significantly higher than the function values from which they were transferred, indicative in the decrease of the success rate. A complexity to consider when information is transferred between meshes is that the sensor unfortunately do not overlap and it could be that the course mesh has a great location whereas the other meshes are slightly off.

5.1.2 F2S3

F2S3 uses the mean scaling method and is a maximisation objective function. The results of F2S3 are shown in Table 5.8 and Figure 5.6. The success rates for both the coarse mesh and the "coarse to medium" mesh are identical but the number of iterations for the latter is higher than expected. However, note that the function value of the latter is also higher as depicted in Figure 5.6. This implies that the transferred sensors could explain the variance in Y better according to the function value and also implies that the resolution of the proxy is limited since the function values ranking can not always match the ranks of the number of iterations required by the optimal sensor sets. This can also be caused by the difference in the model errors of the two meshes. Fortunately, A used for both medium and fine meshes are the same. Thus, the direction of comparison in the function values of the medium and fine meshes is possible. Note that the fine mesh performs better than the "medium to fine" mesh and also has a higher function value. Consider the results of the medium mesh and the "fine to medium" mesh. Although the function value of "fine to medium" is slightly higher than that of the medium mesh, it has a lower success rate than the medium mesh. This

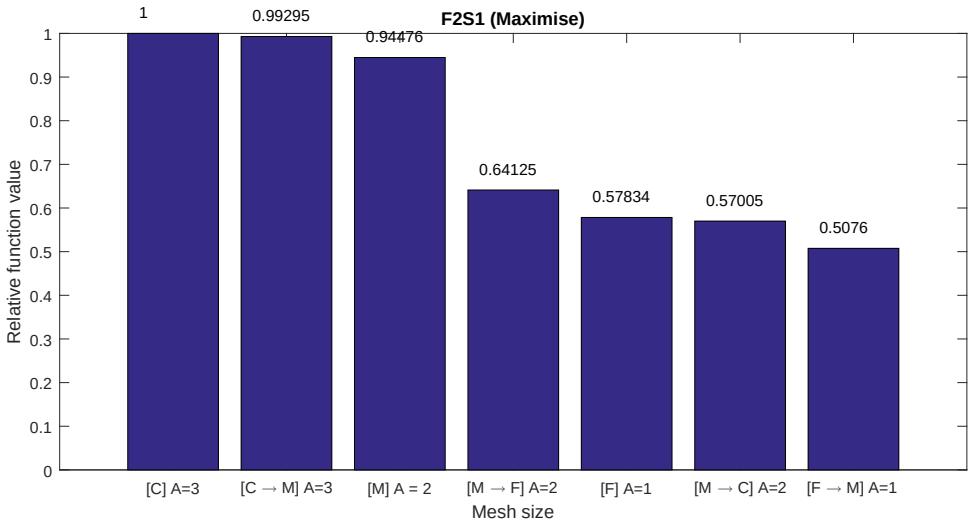


Figure 5.5: Function values of F2S1 (Maximisation problem).

Mesh size	Acc.	Iter.	Mesh size	Acc.	Iter.
Coarse	99	266.39			
Coarse to Medium	99	275.25	Medium to coarse	99	255.51
Medium	99	260.35			
Medium to Fine	65	266.98	Fine to medium	87	263.89
Fine	85	267.16			

Table 5.8: Success rates and numbers of iterations for F2S3

clearly shows the limitation of the resolution of the proxy in that optimal sensor combination rated higher by the proxy actually performs worse on the inverse problem. The proxies of the inverse problems can not be expected to behave perfectly the same as the inverse problem errors. However, it has been shown that the proxy function dataset scaling combinations under investigation are robust in finding suitable sensors that are highly correlated with the ability to solve the inverse problem.

5.1.3 F2S4

F2S4 uses the Z-score scaling. The "coarse to medium" sensors do not perform better than those from the medium mesh. However, it is noticed that the "medium to fine" mesh performs significantly better than the fine mesh, $A_{medium} = A_{fine} = A_{medium \to fine} = 1$. Comparing function values of the "medium to fine" and fine meshes is sensible. Figure 5.7 shows that the "medium to fine" mesh actually has a lower function value than the fine mesh, although the "medium to fine" mesh perform worse than the fine mesh when solving the inverse problem. This highlights again the lack of detailed correlation between the proxy and the inverse problem. Fortunately, the errors are less than 10% in the success rates.

Mesh size	Acc.	Iter.	Mesh size	Acc.	Iter.
Coarse	98	287.34			
Coarse to Medium	64	303.79	Medium to coarse	81	284.33
Medium	95	275.42			
Medium to Fine	100	249.96	Fine to medium	100	260.02
Fine	93	314.01			

Table 5.9: Success rates and numbers of iterations for F2S4

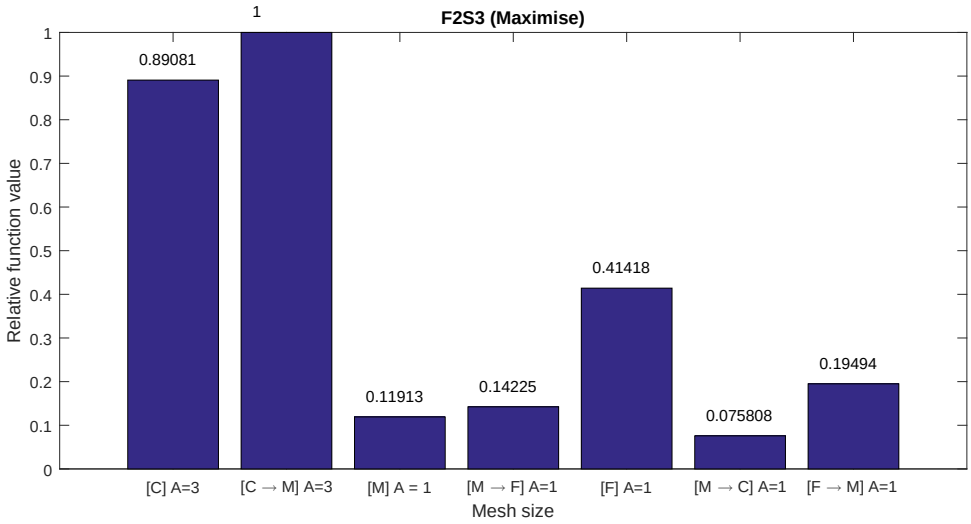


Figure 5.6: Function values of F2S3 (Maximisation problem).

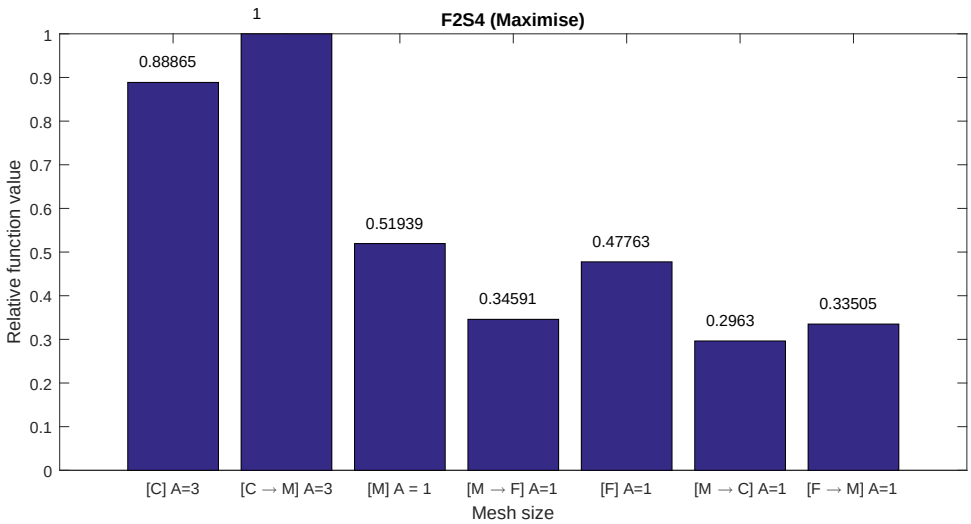


Figure 5.7: Function values of F2S4 (Maximisation problem).

Mesh size	Acc.	Iter.	Mesh size	Acc.	Iter.
Coarse	100	260.97			
Coarse to Medium	99	272.09	Medium to coarse	89	286.31
Medium	96	265.73			
Medium to Fine	100	250.08	Fine to medium	100	222.58
Fine	99	241.23			

Table 5.10: Success rates and numbers of iterations for F2S5

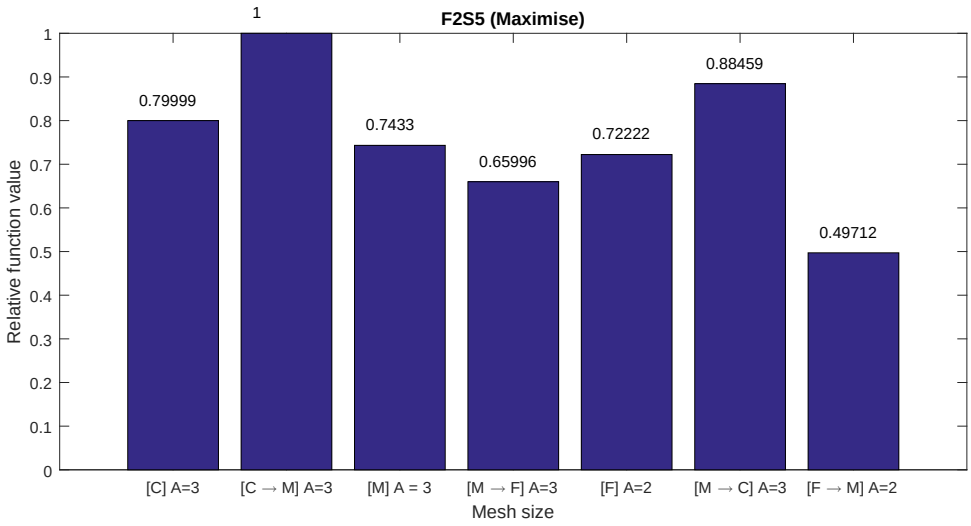


Figure 5.8: Function values of F2S5 (Maximisation problem).

5.1.4 F2S5

F2S5 uses the vector mean scaling method for the maximisation of variance explained. Figure 5.8 shows that this method generally uses a larger number of modes than the other methods and Table 5.10 shows that the success rates are also relatively higher. It is interesting to note that the function value of "coarse to medium" mesh is higher than that of the medium mesh. This is due to the medium mesh that is unable to search through the Gauss points which appeared only in the coarse mesh. It is noticed that the "coarse to medium" mesh has better accuracy than the medium mesh as well. On the other hand, the "medium to coarse" mesh has a greater function value and lower accuracy.

The five mode-based proxy function dataset scaling combinations have been investigated in the presence of model errors. Although they were confirmed to be robust in terms of finding the responsive sensors for the inverse problems, as expected, they are not highly correlated with the inverse problems in details such as the number of iterations and small variations in the success rates. The phenomena commonly observed from all the proxies are that as the mesh sizes of the models decrease, the less modes A are required. As the complexity of a model increases by mesh refinement, the model may contain irrelevant detailed data for the recovery of the optimal sensors. This is why A's decrease as the meshes are refined. Note that S1 is simply just the mean centering and S5 consists of the mean centering and vector mean normalisation which do not change the information of the sensors within the measurement groups. Therefore, the optimal A for F1S1, F2S2 and F2S5 are slightly higher than A chosen for F2S3 and F2S4.

5.2 Different numbers of sensors

The numbers of sensors for the beam problems have been fixed to three sensors. We now consider the consequence of using more or less sensors on the ability to recover the inverse problem. To answer this question, two and four sensors were tested on the coarse mesh beam

in order to be able to implement the combinatorial approach. Again, the KHEH case was considered for this study. Each method was used to find two, three and four sensors and their performances compared.

5.2.1 F1S1 and F2S1

The results of F1S1 for different numbers of sensors are shown in Table 5.11a. It is not surprising to observe that the two sensors do not perform well at all since there are three target variables to be recovered. The number of modes A is limited by the number of sensors. It is noticed that two sensors fail to recover the target variables and successful recovery only starts from three sensors onwards. Only from the three variables which match the number of independent target variables, F1S1 started to show over 90% success rates. Increasing the number of sensors to four has a negligible effect on the results. It is noticed that the same sensors are reselected when the modes A are increased. For instance, the sensor ID number 54 and 64 appear in all the sensor groups with $A = 1$. Note that sensors are also repeated between the three and four sensors result using the same modes, (i.e. $A = 1$), while at other times a sensor is replaced by two other sensors (i.e. for $A = 2$ sensor 65 is replaced by 68 and 70). The only difference between the results of F1S1 and F2S1 (Table 5.11b) is the results of using the full modes when choosing four sensors. Although the two functions produced two distinct sets of sensors due to the numerical errors for implementing SIMPLS using Matlab's `plsregress` built-in function for the full modes, the results of F2S1 and F1S1 are poor due to the over-fitting.

5.2.2 F2S3

The results for F2S3 which use the mean scaling are shown in Table 5.11c. Only from three sensors onwards are target variables successfully recovered. The only improvement of significance that is observed from three to four sensors is when $A = 3$ improves from 16% to 88%. Again, the four sensors with the full modes perform poorly due to over-fitting.

5.2.3 F2S4 and F2S5

The results for F2S4 and F2S5 are shown in Tables 5.11d and 5.11e. As expected, the two sensor sets did not manage to recover any of the unknown variables. The performance between the three and four sensors is effectively unchanged.

The studies showed that having an extra sensor does not deteriorate the performance and may either significantly or slightly improve performance. However, once the optimal number of sensors are obtained adding new sensors has effectively no effect. Significant improvement is achieved when the number of the sensors gets lower than the number of the target variables. Although the five methods experience certain limitations in the resolution as the proxies of the inverse problems, they were proven to be robust in selecting effective sensors for experiments.

5.3 Sensor convergence tests

A big part of this study is to see whether the sensors converge spatially or not. Hence, the following three types of sensor convergence for the mode-based formulations are investigated and discussed.

1. Sensor convergence for changing modes.
2. Sensor convergence for BGA for the fine mesh problem.
3. Sensor convergence for the model errors (coarse, medium and fine meshes).

In addition, the sensors selected for the different number of sensors are also graphically presented and discussed.

2 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	54	48		
Sen. 2	64	55		
Acc.	**0**	**0**		
Iter.	**NaN**	**NaN**		
3 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	23	37	9	
Sen. 2	54	65	23	
Sen. 3	64	69	48	
Acc.	**97**	**92**	**100**	
Iter.	**307.8**	**282.7**	**259.8**	
4 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	13	37	9	62
Sen. 2	23	68	23	67
Sen. 3	54	69	42	68
Sen. 4	64	70	45	70
Acc.	**99**	**93**	**100**	**53**
Iter.	**297.2**	**286.5**	**260.2**	**311.1**

a F1S1

2 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	54	48		
Sen. 2	64	55		
Acc.	**0**	**0**		
Iter.	**NaN**	**NaN**		
3 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	23	37	9	
Sen. 2	54	65	23	
Sen. 3	64	69	48	
Acc.	**97**	**92**	**100**	
Iter.	**307.8**	**282.7**	**259.8**	
4 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	13	37	9	14
Sen. 2	23	68	23	16
Sen. 3	54	69	42	51
Sen. 4	64	70	45	52
Acc.	**99**	**93**	**100**	**56**
Iter.	**297.2**	**286.5**	**260.2**	**346.8**

b F2S1

2 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	50	30		
Sen. 2	54	35		
Acc.	**0**	**0**		
Iter.	**NaN**	**NaN**		
3 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	44	31	17	
Sen. 2	50	37	30	
Sen. 3	54	50	35	
Acc.	**99**	**0**	**16**	
Iter.	**266.4**	**NaN**	**276.9**	
4 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	44	28	17	7
Sen. 2	50	30	38	8
Sen. 3	54	35	39	9
Sen. 4	55	38	41	10
Acc.	**98**	**0**	**88**	**46**
Iter.	**263.6**	**NaN**	**276.9**	**344.5**

c F2S3

2 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	48	30		
Sen. 2	54	35		
Acc.	**0**	**0**		
Iter.	**NaN**	**NaN**		
3 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	48	28	17	
Sen. 2	54	30	30	
Sen. 3	55	35	35	
Acc.	**98**	**0**	**16**	
Iter.	**287.3**	**NaN**	**276.9**	
4 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	48	34	17	10
Sen. 2	54	38	28	12
Sen. 3	55	39	30	14
Sen. 4	64	41	35	16
Acc.	**98**	**0**	**19**	**11**
Iter.	**284**	**NaN**	**278.2**	**329.6**

d F2S4

2 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	31	23		
Sen. 2	37	37		
Acc.	**0**	**0**		
Iter.	**NaN**	**NaN**		
3 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	31	23	9	
Sen. 2	37	27	23	
Sen. 3	50	48	37	
Acc.	**0**	**100**	**100**	
Iter.	**NaN**	**261**	**261**	
4 Sensors				
Mode	A = 1	A = 2	A = 3	A = 4
Sen. 1	31	23	23	11
Sen. 2	37	27	30	23
Sen. 3	43	45	42	35
Sen. 4	50	48	65	41
Acc.	**0**	**100**	**100**	**100**
Iter.	**NaN**	**259.2**	**257.9**	**260.1**

e F2S5

Table 5.11: Objective function value comparisons

5.3.1 Sensor convergence test for changing modes

Identifying appropriate numbers of mode is essential in the mode-based formulations. This test aims to investigate whether there is any sign of convergence as the number of mode varies. As a representative, the sensors selected using F1S1 which is one of the best performing proxies for the KHEH case of the coarse mesh model presented in Table 5.2 are considered and the spatial distributions of the sensors for $A = 1$, 2 and 3 are presented in Figures 5.9(a)-(c), respectively. It is noticed from Table 5.2 that these sensor sets have success rates of 97, 92 and 100% which are very accurate and yet, the figures show that the sensors do not converge as the number of modes changes.

5.3.2 Sensor convergence test for BGA for the fine mesh problem

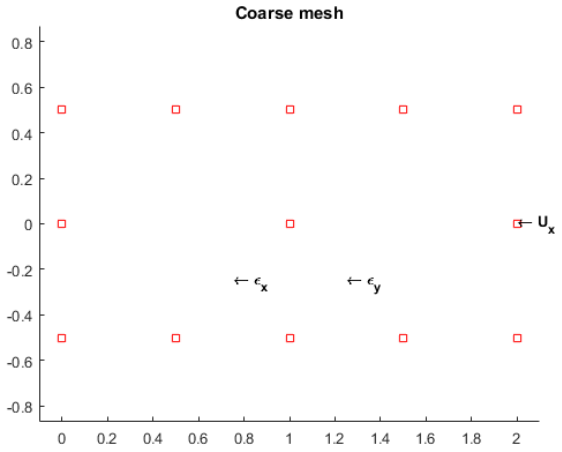
As discussed in Section 5.1, the BGA was implemented for the fine mesh problems to make the sensor optimisation more computationally tractable. Since the BGA does not guarantee that the optimal solution will be found, it was implemented ten times for every fine mesh problem and the best solutions among them were used. The setting for the BGA is the same as presented in Section 5.1. The sensor convergence test presents the ten sets of the selected sensors for the KHEH case of the fine mesh using F1S1 in Figure 5.10. This was how the sensors for the fine mesh problem with the F1S1 presented in Table 5.7 were chosen. The sensor placements depicted in the figure are optimised for $A = 1$ since that is the same number of mode chosen for the fine mesh result in Table 5.7. Figure 5.10 only shows a single set of three sensors and this means that all the ten sets of the sensors are identical. Hence, the sensor convergence for the BGA approach showed a confident result for the fine mesh problem since this was an exhaustive BGA. However, note that this still does not guarantee that this result is the global minimum.

5.3.3 Sensor convergence test for the model errors

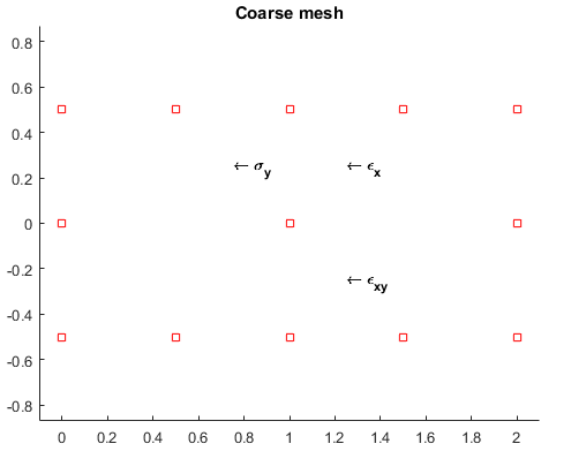
The convergence test for the sensors selected from F1S1, one of the best performing proxies, was conducted as the mesh was refined and coarsened. The sensors used to generate the performances for the coarse, medium and fine meshes presented in Table 5.7 for the KHEH case are depicted in Figures 5.11(a)-(c). Although Table 5.7 shows that the three sets of sensors have 100% accuracies, it does not seem obvious to understand the reason behind their placements. The figures show that the sensors do not converge as the mesh size is refined and coarsened for the case considered. Notice that both the positions and types of the sensors do not converge.

5.3.4 Graphical presentation of the selected sensors for various numbers of sensors

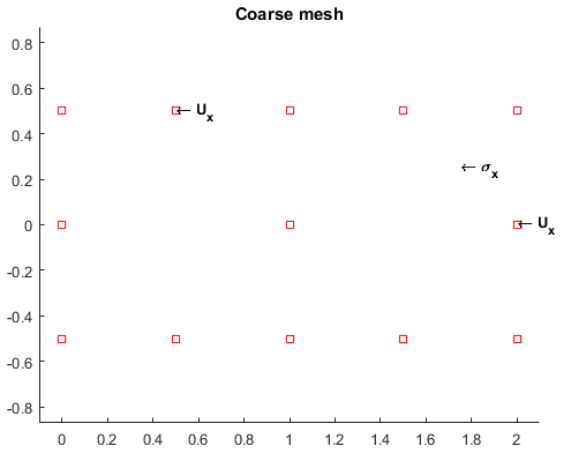
What happens to the sensors as the number of the sensors increases for the mode-based formulations was already discussed in Section 5.2. Hence, this section aims to visually investigate how the extra sensors get located in the problem. Tables 5.11a-5.11e present the performances of the sensors selected using the various proxies for the KHEH case for the coarse mesh problem. The sensors from F1S1 which is one of the best performing proxy as tabulated in Table 5.11a were chosen to present the placement of the extra sensors as the number of sensors increases for $A = 1$ and 2 in Figure 5.12. The spatial distribution of the sensors for the 2, 3 and 4 sensors for $A = 1$ and 2 is indicated in Figures 5.12(a)-(c) and 5.12(d)-(f), respectively. Similar to the other proxies from Table 5.11a, F1S1 chooses extra sensors as the number of sensor increases while keeping the other sensors chosen earlier unchanged when $A = 1$ since it only has a single mode. Hence, it only focuses on an adding new sensor which will help improve the function values of the proxies. On the other hand, when $A = 2$, it starts balancing between the modes and starts to swap the previously chosen sensors with new sets of sensors to satisfy the optimisation.



(a) $A = 1$



(b) $A = 2$



(c) $A = 3$

Figure 5.9: Sensor convergence test for changing the number of modes for the mode-based formulation F1S1 for the KHEH case when (a) $A = 1$, (b) $A = 2$ and (c) $A = 3$.

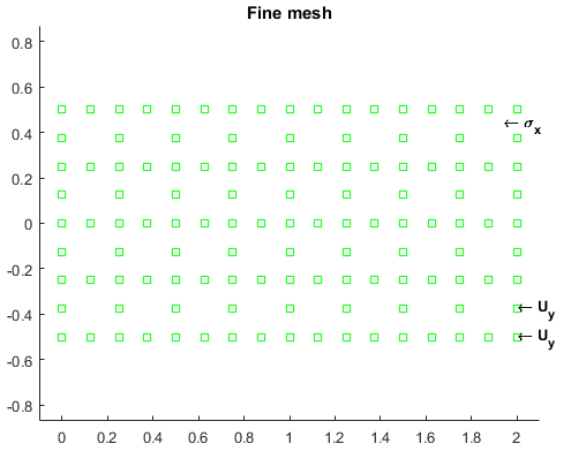


Figure 5.10: Sensor convergence test for BGA for the fine mesh problem using F1S1 for $A = 1$ over ten times of optimisation.

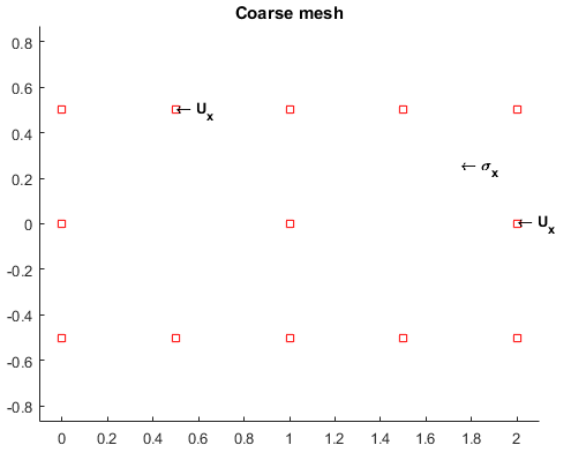
5.4 Causality

Before moving on to the next chapter to introduce the mode-free formulations, causality in SPO needs to be addressed. Many researchers who work with big data only try to find the correlations between measurement and target data as they work with data and not models. This is important as for many big data analyses the correlation does not imply causality. Typically correlation includes causal relationships and additional correlated but not causal variables. An easy and highly efficient way to consider causality between variables is to manipulate the model target variables independently at the same time while generating the data for analyses which is exactly what has been done up until now. Since the target variable FEM input values are unrelated with each other, it is possible to observe which target variables influence which sensors the most.

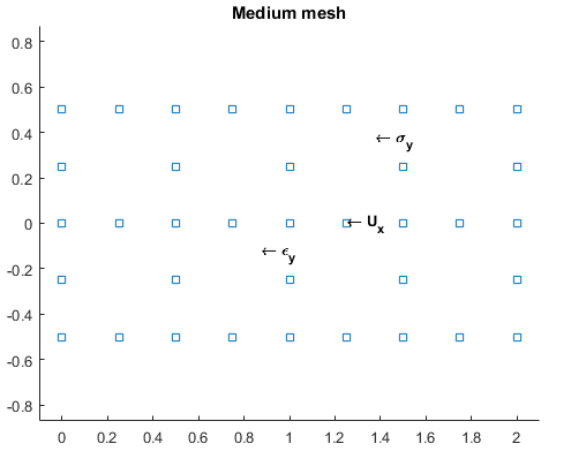
Although it has been shown so far that this way of generating data works for the numerical techniques, it is difficult for a person to see how the readings on the u, σ and ϵ change as the target variables (FEM input variables), E, k_1 and k_2 vary. Therefore, the measurement readings of the sensors were taken as E, k_1 and k_2 varied one at a time. The variations of the σ measurements when varying E, k_1 and k_2 are shown in Figures 5.13 - 5.15, respectively. Note that these σ measurements are measured at all eight Gauss points in the coarse mesh problems. Hence, each of the σ_x, σ_y and σ_{xy} has eight measurements presented in the figures. Since the discussion about the σ applies equally to both the u and ϵ as well, the results for u and ϵ are presented in Appendices.

5.4.1 Manipulation of E

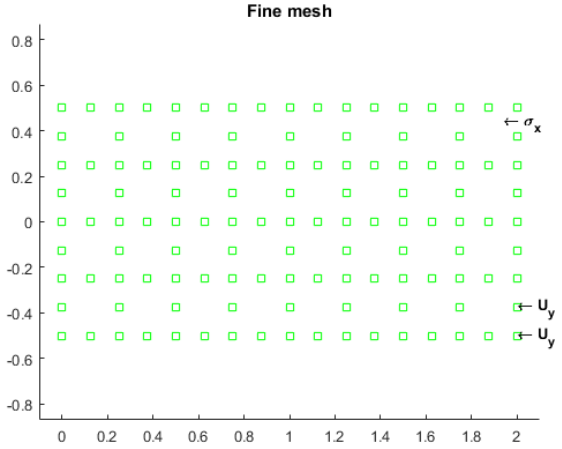
The changes in the σ due to the manipulation of E for all the three cases (KHEL, KHEH and KLEH) for the coarse mesh beam are shown in Figure 5.13. While Young's modulus, E was being manipulated within its ranges, the two k values were fixed at the middle value of their ranges. The middle values were chosen since the ranges are often set around the most confident estimates. Figure 5.13(a) shows the changes of σ for the KHEL case. As expected, the variation and sensitivity of σ decrease as E increases and the beam becomes stiffer. Figure 5.13(b) shows the changes of σ for KHEH. Since the ranges of E are considerably higher when compared to KHEL, σ shows ever lower sensitivity along the variation of E. Lastly, Figure 5.13(c) is for KLEH. As opposed to KHEL, the springs are softer but E is higher at the same time. Thus, σ becomes extremely insensitive to the change in E.



(a) Coarse mesh

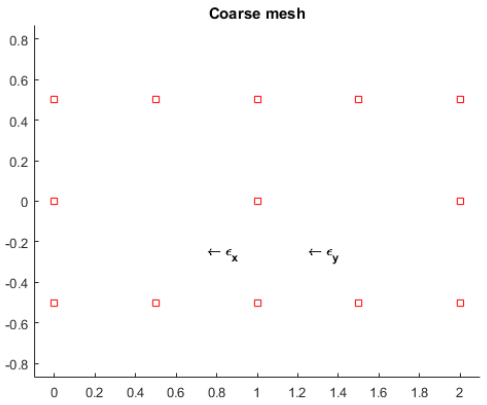


(b) Medium mesh

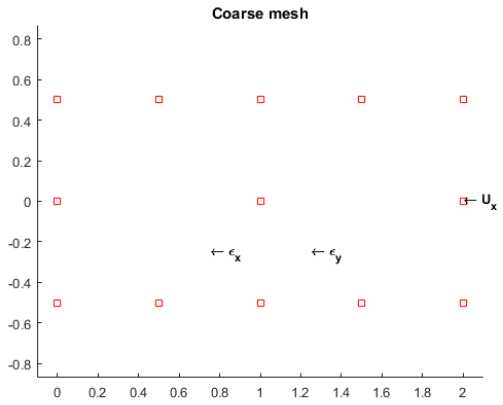


(c) Fine mesh

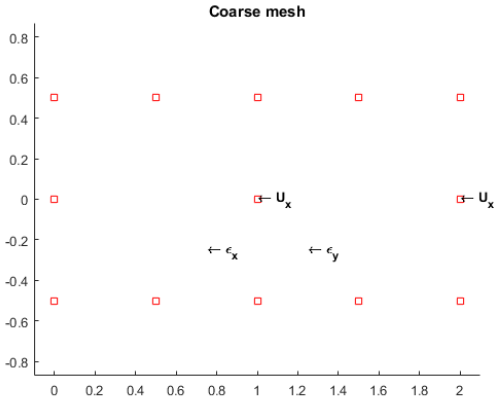
Figure 5.11: Sensor convergence test for the mode-based formulation F1S1 for the KHEH case for (a) coarse, (b) medium and (c) fine meshes.



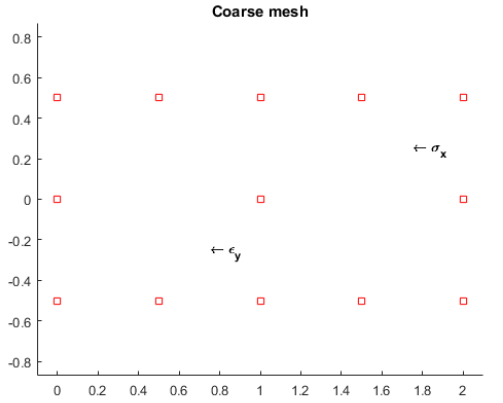
(a) 2 Sensors, $A = 1$



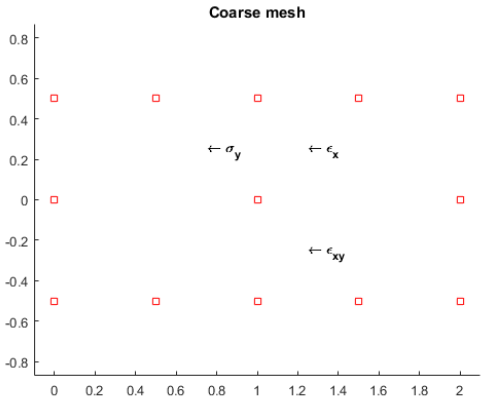
(b) 3 Sensors, $A = 1$



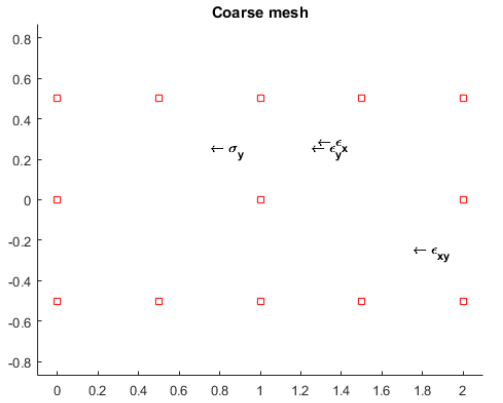
(c) 4 Sensors, $A = 1$



(d) 2 Sensors, $A = 2$

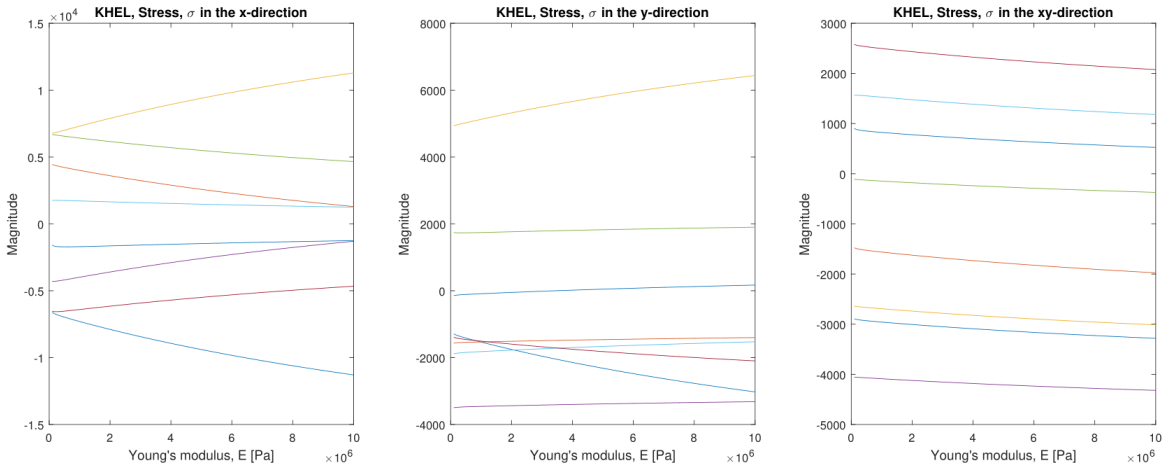


(e) 3 Sensors, $A = 2$

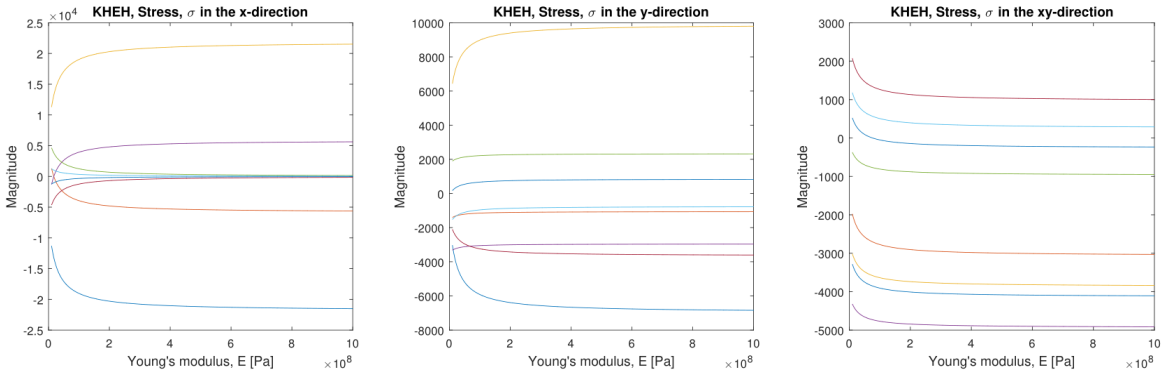


(f) 4 Sensors, $A = 2$

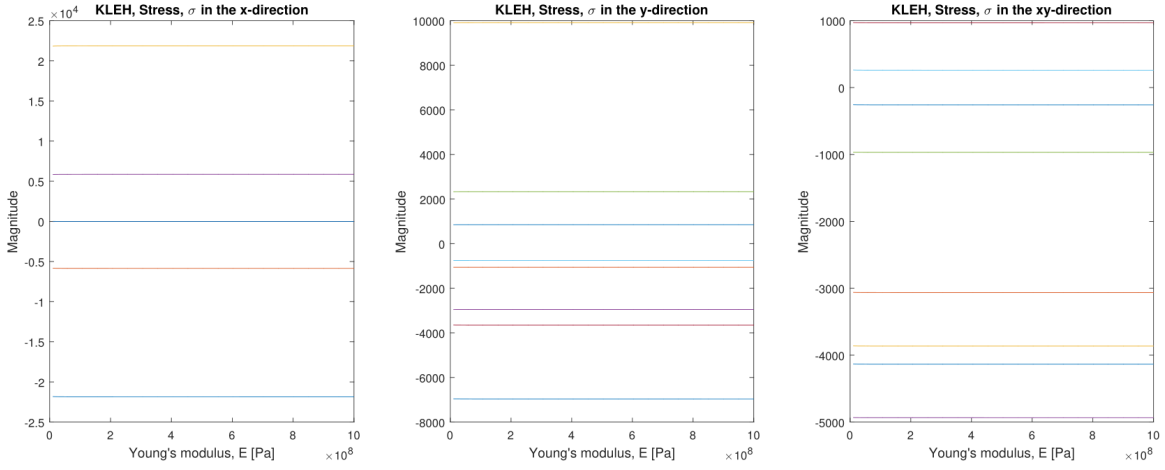
Figure 5.12: Spatial distribution of the sensors for increasing the number of sensors for (a) 2,(b) 3 and (c) 4 sensors when $A = 1$ and (d) 2, (e) 3 and (f) 4 sensors when $A = 2$.



(a) KHEL



(b) KHEH



(c) KLEH

Figure 5.13: Stress, σ sensitivities when adjusting Young's modulus, E for the (a) KHEL, (b) KHEH and (c) KLEH cases.

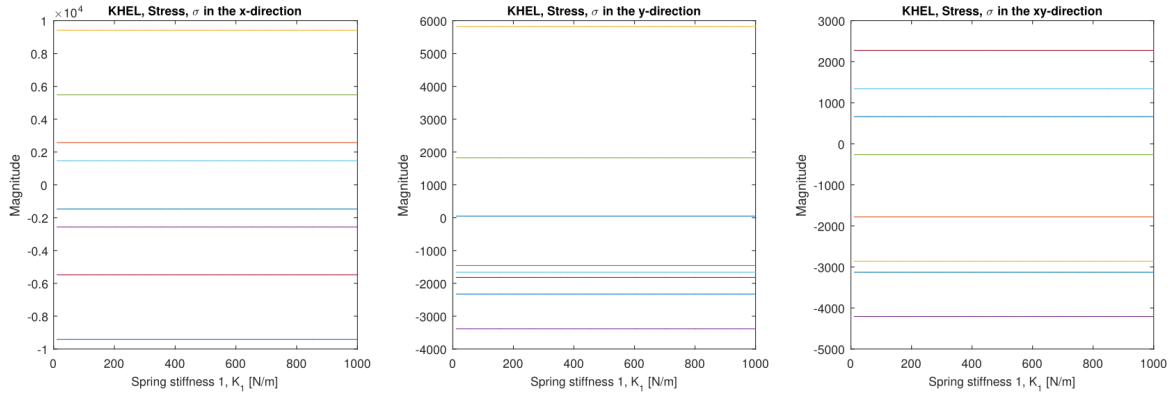
5.4.2 Manipulation of k_1

The changes in σ due to the manipulation of k_1 (the horizontal spring) for all the three cases are shown in Figure 5.14. Recall that k_1 spring is much softer than k_2 spring. Therefore, it is noticed that σ values look almost constant throughout the changes in the spring stiffness. This also means the difficulty of recovering k_1 is greater compared to the other target variables. There is no visible difference in σ even when E is low (KHEL) as shown in Figure 5.14(a).

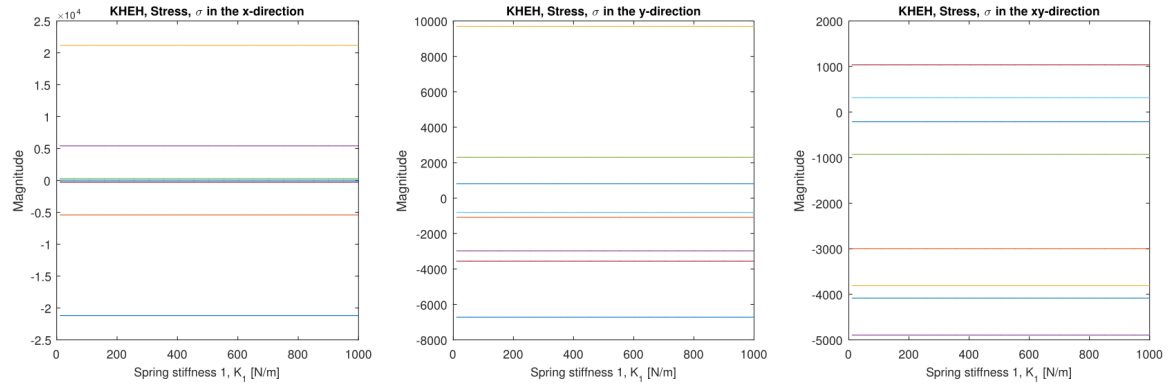
5.4.3 Manipulation of k_2

The changes in σ due to the manipulation of k_2 (the vertical spring) for all three cases are shown in Figure 5.15. The second spring k_2 is about 100 times stiffer than k_1. Thus, it was expected to affect σ along the manipulations. When the range of k is high but the range of E is low (KHEL), the highest sensitivity in σ is shown (see Figure 5.15(a)). This is due to the high compliance of the beam which makes the spring exert higher forces on the beam. When both the ranges of k and E are high (KHEH), it certainly shows visible but less variations in σ with variation in k_2 as shown in Figure 5.15(b). The most challenging case, KLEH still showed no visible difference in σ and the σ measurements seemed constant along the changes of k_2 as shown in Figure 5.15(c). This study was conducted to illustrate what actually happens in the measurement data and how the causalities between the measurement data and target variable data change while manipulating the target variables. In fact, the activities inside the training data are more complex since all the target variables are varied at the same time which saves a lot of time as compared to fixing the rest of the target values at every interval in their ranges while manipulating one variable. This shows the power of using the model instead of having only data. With data, only correlations can be studied but with models, causalities between variables can be examined.

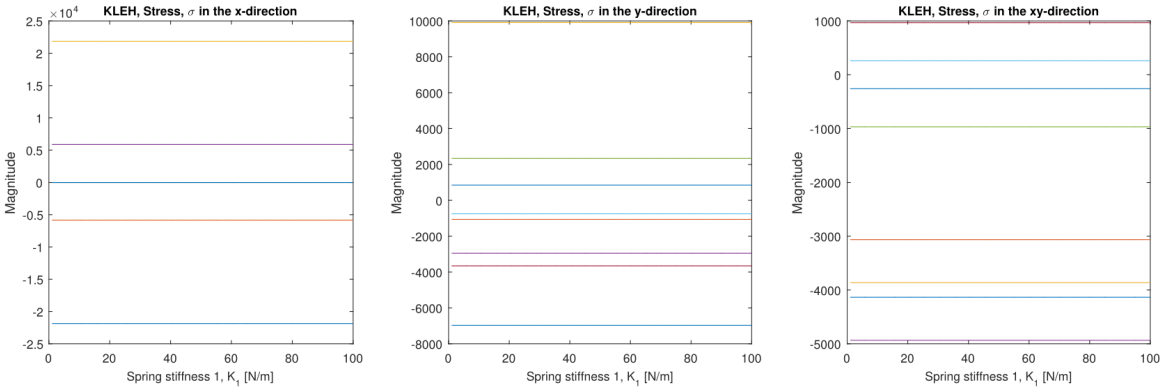
It is essential to note here that this causality test is actually an efficient strategy to identify sensors since it varies each target variable one at a time and can find the sensor with the most variation over the domain. Hence, this approach can be useful in generating initial starting points for the optimisation. The causality study is the first step towards mode-free sensor placement optimisation as a legit strategy to change each target variable independently and compute all the sensor responses. The optimal sensors are merely selected as the sensors with the highest gradient over the domain. However, there is a disadvantage of doing this, which is that although it can be efficient for small target ranges for E, k_1 and k_2, for large target ranges, the variables combinatorially blow up. In addition, one has to be able to directly manipulate the quantity of interest. Some inverse problems where one aims to predict the response at a spatial location based on the response at another location are examples of where such an approach will fail. Hence, our aim is to pursue this multivariate setting in which all sensors are changed at the same time to also include coupling of the variables in the responses.



(a) KHEL

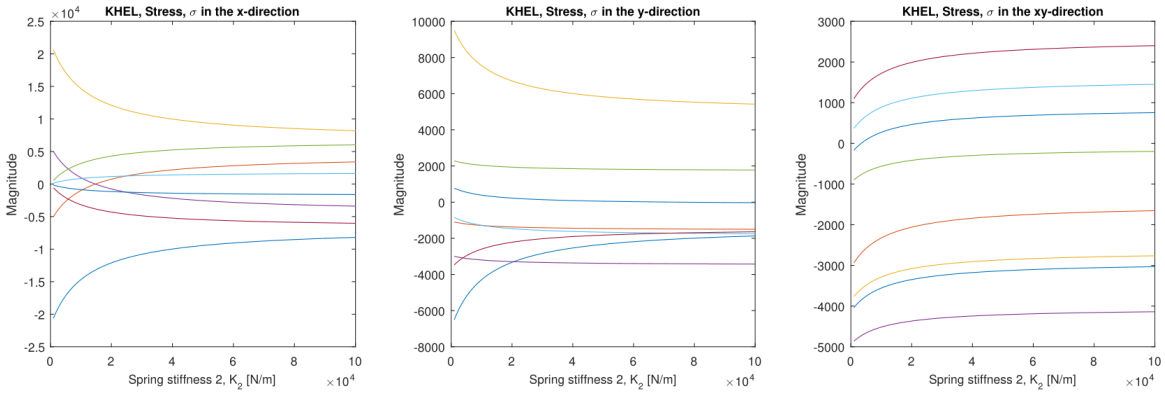


(b) KHEH

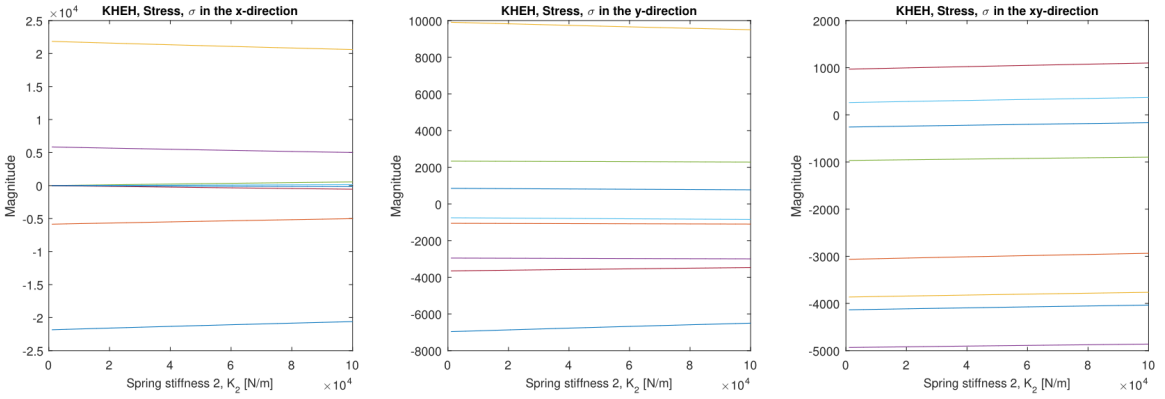


(c) KLEH

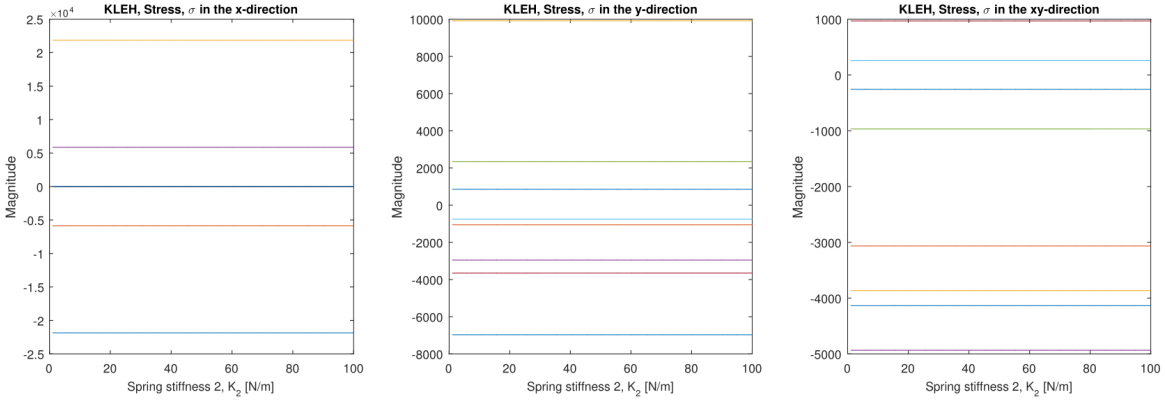
Figure 5.14: Stress, σ sensitivities when adjusting spring stiffness 1, k_1 for the (a) KHEL, (b) KHEH and (c) KLEH cases.



(a) KHEL



(b) KHEH



(c) KLEH

Figure 5.15: Stress, σ sensitivities when adjusting spring stiffness 2, k_2 for the (a) KHEL, (b) KHEH and (c) KLEH cases.

Chapter 6

Mode-Free Formulation

The mode-based formulations proved to be robust and capable of placing useful sensors for inverse experiments. However, they require the right number of modes to be identified, which usually increase as the number of sensors increase. The concept of mode-free formulations was briefly demonstrated at the end of the previous chapter, by varying the target variables over a range and observing how the stress, strain and displacement sensors vary to identify the most sensitive sensors. As discussed, this decoupled approach of the target variables has some disadvantages in that combinatorially this can become expensive and that you have to be able to manipulate the target variables independently. The aim of this chapter is to propose a mode-free formulation that addresses these issues.

From the previous chapters, the following characteristics of the inverse problems were found:

1. The information that the sensors measure should be relevant to the inverse problem at hand, it should be related to the target variables that need to be recovered which is mathematically expressed by

$$rank\left(\left(\frac{X_c - \bar{X}_c}{SD(X_c)}\right)^T \left(\frac{Y - \bar{Y}}{SD(Y)}\right)^2\right). \tag{6.1}$$

2. The sensors chosen for the inverse problem should measure unique information. This is important to ensure that each variable contributes new information about the problem. This can be expressed mathematically by

$$rank\left(\left(\frac{X_c - \bar{X}_c}{SD(X_c)}\right)^T \left(\frac{X_c - \bar{X}_c}{SD(X_c)}\right)^2\right). \tag{6.2}$$

3. Inverse problem requires sensors that result in large variance in the target variables to be recovered. The more sensitive the sensor the better the chances for recovery. This is compactly expressed by

$$\prod_{i=1}^{N_{sensors}} var(x_{c,i}). \tag{6.3}$$

The modes-free formulations consist of the three objectives Eqs. 6.1, 6.2 and 6.3. Notice that Eqs. 6.1 and 6.2 include Z-score which allows for normalisation of the covariance matrices. To combine these objectives with various orders of magnitudes together to form an unconstrained problem, the weights for each objective need to be considered. As an alternative, we rather consider a constrained maximisation problem by strictly considering the sensor sets with the full ranks for Eqs. 6.1 and 6.2. This allows for the constrained optimisation with the single objective function Eq. 6.3 which maximises on the product of the variances of the full ranked sensor sets. The maximum possible number of rank matches the number of the target variables given that the number of sensors is not less than the number of the target

variables. Recall from Section 5.2 that in order for the recovery of the target variables to be successful, the number of sensors should not be less than the number of target variables. Only considering the full rank sensors enhances the computational speed required. Moreover, the mode-based formulations do not require any mode extraction from the measurement and target datasets. Therefore, the computational speed of the mode-free formulation is much faster than the mode-based formulations. This is later proven in Section 6.3. For this reason, the exhaustive combinatorial approach was considered to be computationally tractable even for the fine mesh model. Hence, the exhaustive combinatorial approach was mainly implemented throughout the chapter. The addition approach discussed in Section 3.4 was also implemented to compare its performance with the global maximum found by the exhaustive combinatorial approach.

6.1 Performance comparison between the combinatorial and addition approaches

We now compare the performance between the exhaustive combinatorial approach and the addition approach. The main motivation is to inquire whether the addition approach is sensible as it is computationally tractable for large number of variables. The computational performance of the two approaches to solve the three beam model problems (coarse, medium and fine meshes) with the various target range cases (KHEL, KHEH and KLEH) is investigated as well as their ability to recover optimal solutions. The chosen sensors from these problems were tested on 100 test problems in the identical way as in Chapter 5. Note that both exhaustive combinatorial and addition approaches solve the constrained optimisation to find optimal sensors. Hence, for the exhaustive combinatorial approach, only the sensor sets with full ranks of 3 for Eqs. 6.1 and 6.2 were taken to compute their variances and for the addition approach, the selection of the first sensor only aims for maximising the variance of the sensors since comparing the ranks of single sensor is pointless. From the second sensor selection onwards, the two rank constraints were active to match the number of sensors to the ranks of the sensor sets while maximising the product of the variances of each sensor.

Figure 6.1 shows the performances of the sensors selected using the exhaustive combinatorial approach and the addition (fast) approach discussed in Chapter 3. The combinatorial approach guarantees a global maximum but is computationally intractable for a large system. The addition approach does not guarantee a global maximum but it is computationally efficient. The sub-figures show the accuracies and the numbers of iterations for the different mesh sizes (coarse, medium and fine) for the three cases (KHEH, KHEL and KLEH). The graphs show that the combinatorial approach often managed to obtain 100% accuracies and at its worst, 95%. In addition, the number of iterations required to solve the inverse problem ranged between 224 and 270 iterations, which constitutes the global maximum solutions to the problem. The addition approach is able to recover competitive solutions to the global minimum in all cases. The required number of iterations to solve the inverse problem ranged between 224 and 277. Figure 6.2 shows the comparisons between the combinatorial approaches and the randomly chosen sensors for the coarse, medium and fines meshes, respectively. The success rate for the randomly selected sensors varied between 31% and 92%. The lowest success rate for the addition approach was 93%. In addition, the number of iterations required to solve the inverse problem by randomly selecting sensors ranged between 230 and 360 iterations. The results of randomly chosen sensors coloured in yellow show visibly poorer results than those for the mode-free formulations. Therefore, this clearly proves the robustness of the mode-free formulations.

6.2 Convergence of sensors using mode-based formulations

When using the mode-based formulation, even though the mesh sizes of the beam problems got finer to the fine meshes, it was not enough for the formulations to show any convergence

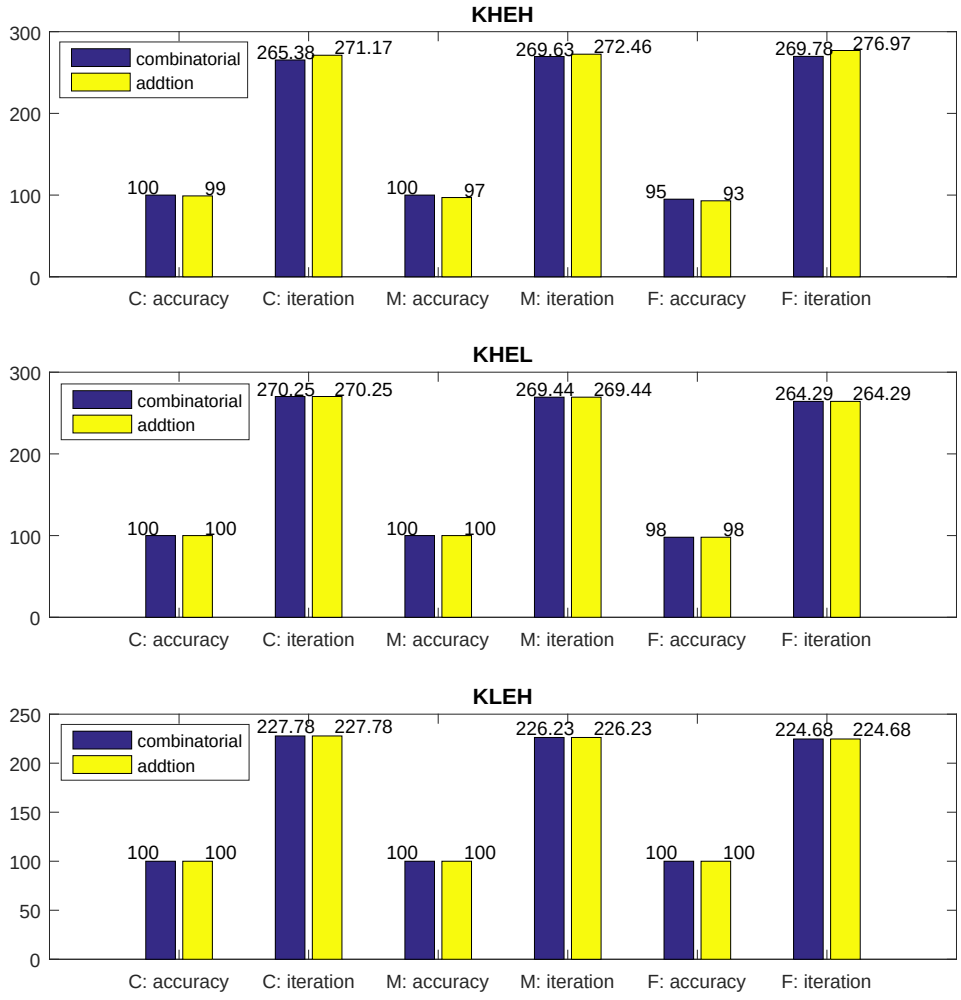


Figure 6.1: Accuracy test of sensors found using combinatorial approach and addition approach.

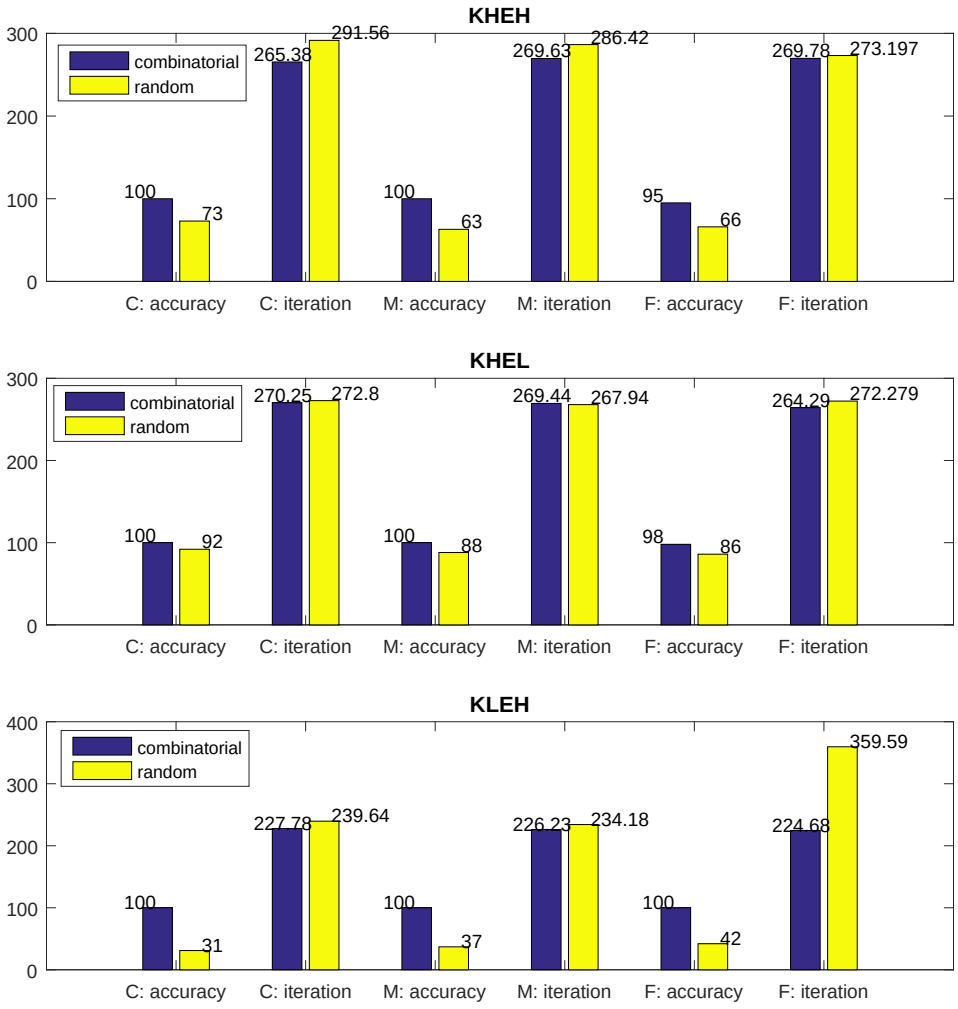


Figure 6.2: Accuracy test of sensors found using combinatorial approach and random sensors.

and the sensor location seems to move around inconstantly as illustrated in Section 5.3.3. Therefore, convergence of the sensors can only be noticed when the model error of a problem decreases to the point where the sensitivities of the sensors are starting to converge to the point where the rankings are consistent. The associated errors might still move around. The three beam cases (KHEH, KHEL and KLEH) with the various mesh sizes as in the previous chapter were considered to observe the convergence of mode-free formulations. Whereas mode-based formulations can vary its optimal sensor positions depending on the numbers of modes as illustrated in Section 5.3.1, the results of the constrained mode-free formulations with the exhaustive combinatorial optimisation approach clearly show its convergence as the sizes of mesh get less as shown in Figures 6.3, 6.4 and 6.5.

6.2.1 KHEL

The convergence of the three sensors for KHEL is shown in Figures 6.3(a)-(c) for the coarse to fine meshes. Note that squares in the figures indicate the nodes of the elements in the beam. Since KHEL has the most flexible beam among the three cases, a large deflection is expected from the same amount of load applied at the middle of the beam. Therefore, the two σ_x sensors are placed close to the root and the u_y sensor are placed where the load is applied. The two σ sensors are mainly for recovering E and k_1 and the u_y is placed there due to the high compliance of the beam. The medium mesh and the fine mesh beams (see Figures 6.3(b) and (c)) clearly show convergence in the sensor positions. It is noticed that the two σ_x sensors have moved more towards the top and the bottom of the beam where the variance of stress is greater.

6.2.2 KHEH

The convergence of the three sensors for KHEH is shown in Figure 6.4. The coarse mesh beam has its two σ_x sensors located at the bottom close to the root of the beam and in the top middle. The u_y sensor is placed at the tip of the beam. Recall that KHEH has the high ranges of both E and k. Since the vertical springs k_1 has almost an unnoticeable significance in the measured data as shown in Figure 5.14, the σ_x at the top is measured closer to the tip than the other σ sensor. The u_y sensor is placed where it can measure the change in u due to k_2 effectively. Figure 6.4(b) shows that the σ_x sensors moved slightly outwards from their positions where σ magnitude is larger. However, there is neither change in the types of sensors nor large movement in the spatial location of the sensors. Figure 6.4(c) shows that the fine beam has two σ_x sensors even closer to the top and the bottom. The position of u_y has changed to the bottom of the beam but still close to the tip of the beam. Since each sensor is ranked by its variance, there can be slight changes in the position of u_y such as this. This shows that mode-based formulation certainly shows convergence for the cantilever beam problems.

6.2.3 KLEH

Lastly, the convergence of the three sensors for KLEH is shown in Figures 6.5(a)-(c) for the coarse to fine meshes. The KLEH beam has high stiffness yet the springs are relatively weaker. Therefore, as compared to KHEL, u_y has moved to the tip of the beam where the displacement and variance is the greatest and the two σ_x sensors are placed at the root where σ is the greatest. It is noticed that all the sensors focus on finding the spatial location that results in the most variance as dictated by the optimisation problem. The medium and fine mesh beams (see Figure 6.5(b) and (c)) clearly show the convergence of the sensors' positions. Again, the two σ_x sensors are moved towards the corners at the root.

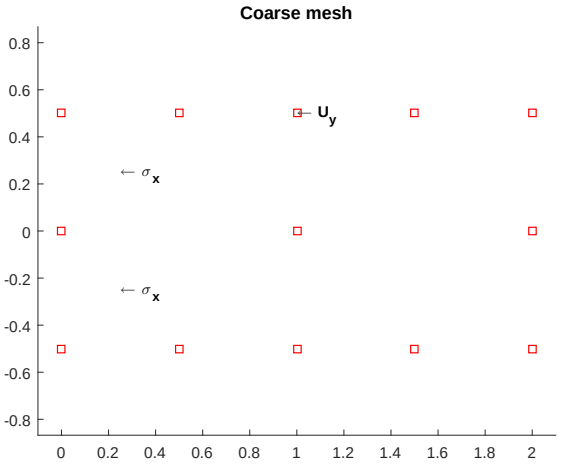
Each case has the different ranges of E and k. Thus, although they require the same types of the sensors (two σ_x and a u_y sensors), the sensors are located differently depending on the behaviours of the structure and the measurements. It turns out that the cantilever beam problems do have the converged solutions with the mode-free sensor searching formulation

while the mode-based formulation did not manage to have convergence for the various sizes of elements.

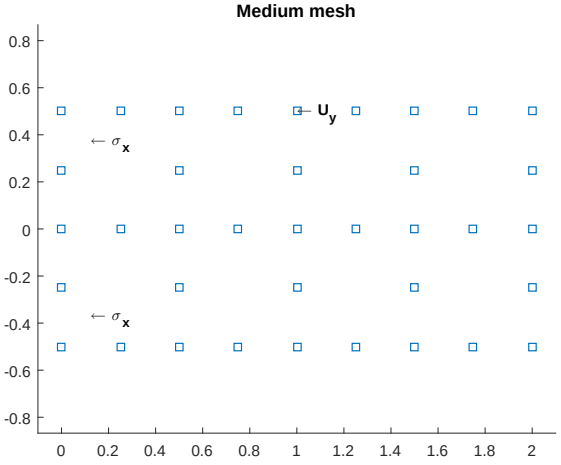
6.3 Computational speed test

The computational speed of SPO becomes critical when there is a large number of sensor options available for an experiment and a lot of sensors are used since the total number of combinations grows exponentially depending on both the numbers of sensor to select from and the number of sensors to use for the experiment. Note that using extremely fine meshes does not have to mean that SPO takes long to compute as the number of sensor locations can be decided by the user. To allow for a comprehensive comparison on the computational performance, we consider the two best performing mode-based formulations, the exhaustive combinatorial approach as well as the addition approach.

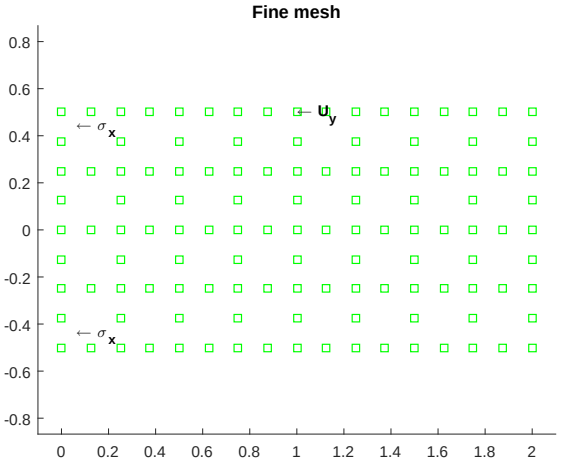
The computational time required for the two best performing functions of mode-based formulations, F1 and F2, is compared to the time required for the exhaustive combinatorial and mode-free addition method. In Figure 6.6, note a log_{10} is used. The addition approach is three orders faster than the combinatorial approach for our fairly small problem. The mode-based formulations appear to be one order slower than the combinatorial approach. As expected, mode-based formulations take longer since they need to be implemented for all the possible numbers of A, which both the number of modes that performed best as well as the number of sensors are three for this example. The computational time difference between mode-free and mode-based would grow larger if the numbers of the sensors increase since the mode-based formulations require to test all possible modes and identify the right number of modes. F1 is slightly slower than F2 as the regression errors can only be computed after computing the modes which express the amount of the variance. The speed of the mode-free addition method is over six orders (a million times) faster than that of the mode-based combinatorial method for the fine mesh problems. If one is slightly concerned about the performances of the mode-free addition method and wants to make sure it would succeed in solving the inverse problems given that the problems are not ill-posed, adding an extra sensor would easily resolve the issue and this option would still be computationally tractable. Importantly, the role of signal to noise ratio is investigated on the ability to recover sensors.



(a) KHEL: Coarse mesh

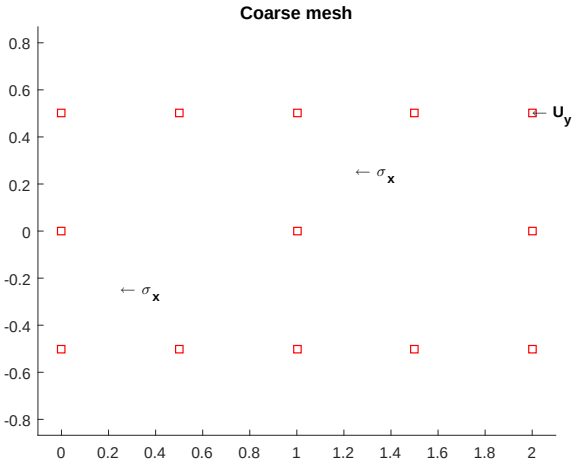


(b) KHEL: Medium mesh

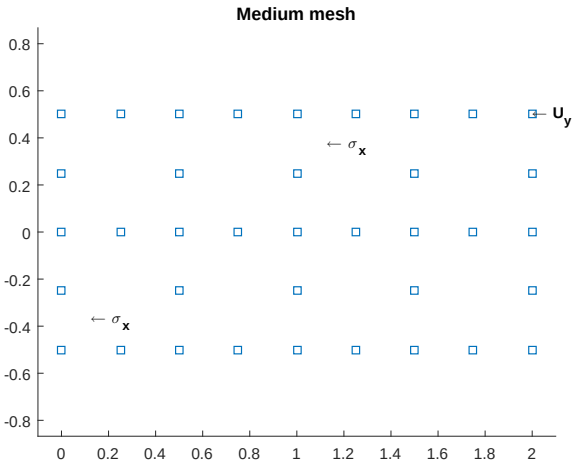


(c) KHEL: Fine mesh

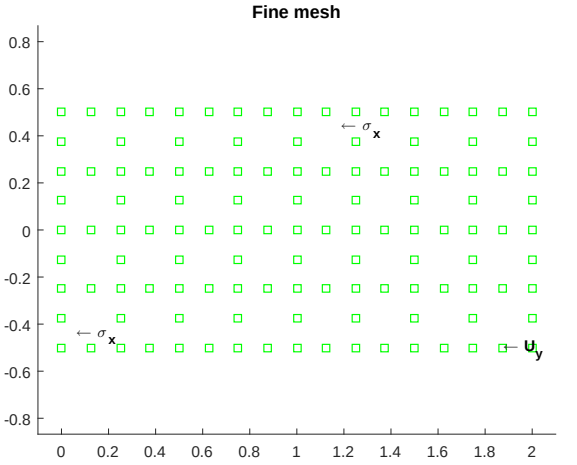
Figure 6.3: Optimal sensor positions of KHEL obtained from the mode-free formulation optimised using the exhaustive combinatorial approach for (a) coarse, (b) medium and (3) fine mesh models.



(a) KHEH: Coarse mesh

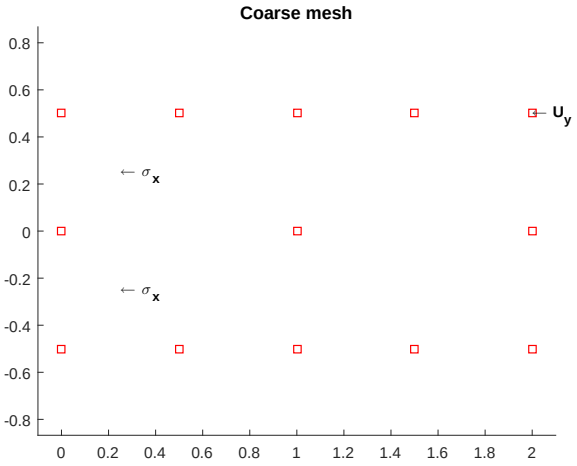


(b) KHEH: Medium mesh

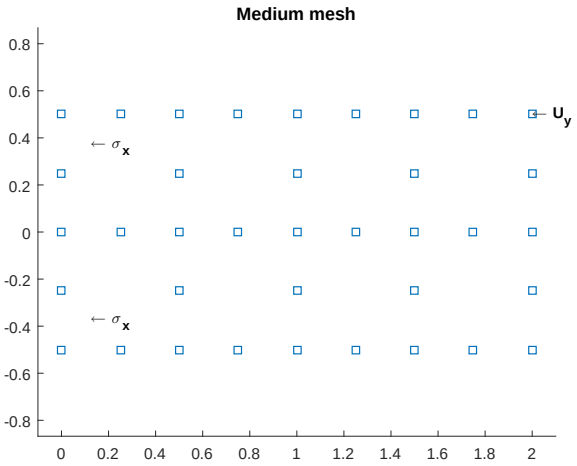


(c) KHEH: Fine mesh

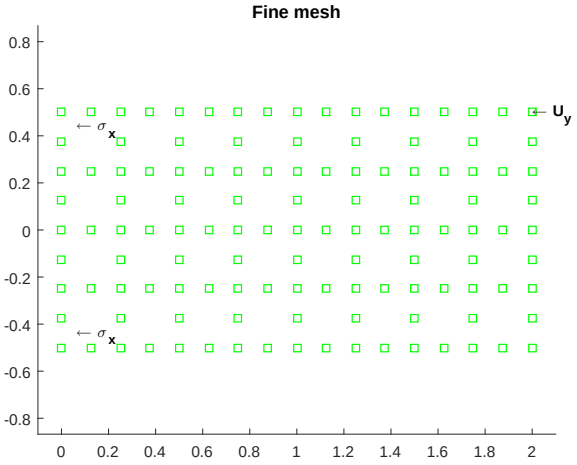
Figure 6.4: Optimal sensor positions of KHEH obtained from the mode-free formulation using the exhaustive combinatorial approach for (a) coarse, (b) medium and (3) fine mesh models.



(a) KLEH: Coarse mesh



(b) KLEH: Medium mesh



(c) KLEH: Fine mesh

Figure 6.5: Optimal sensor positions of KLEH obtained from the mode-free formulation using the exhaustive combinatorial approach for (a) coarse, (b) medium and (3) fine mesh models.

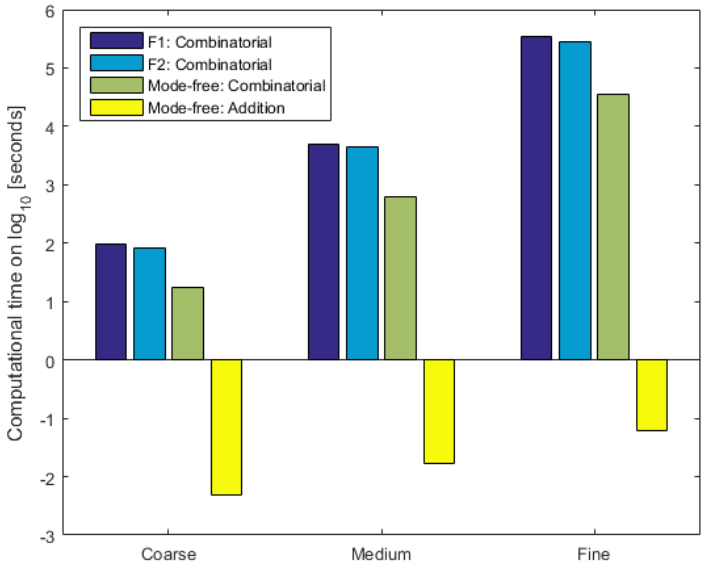


Figure 6.6: Computational speed test for the computation time taken for the various SPO methods.

Chapter 7

Effects of Stochastic Noise

Real life experimental setups involve stochastic noise. Two types of noise are considered in this paper: (1) class noise and (2) point noise.

- Class noise: some sensors may measure the same variables just in different positions. For instance, σ can be measured close to the root or the tip of a cantilever beam and they would have different sensitivity to the FEM input variables or target variables of the inverse problems. Class noise mimics experimental noise that contributes an equal amount of noise to all sensors in the same sensor class. Therefore, whereas the information from the σ sensors close to the tip of the beam may be obscured by noise, the sensors at the root still offer valuable information in the presence of noise. In this study, the maximum noise magnitude is taken as a fraction of the sensor with the largest variance.

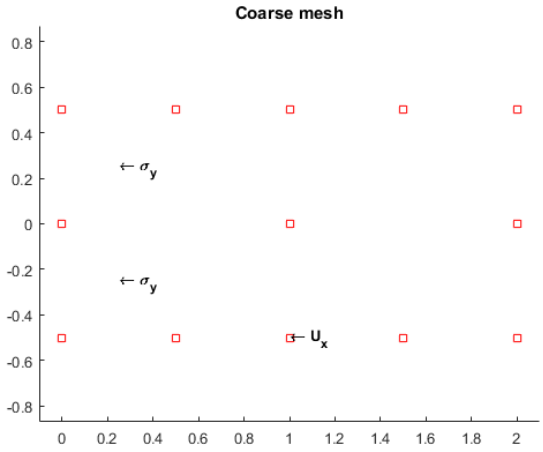
- Point noise: point noise adds a certain percentage of variance of the sensor as noise. Therefore, sensors with larger variance would have larger noise than the sensors with lesser variance.

Note only the effect of stochastic measurement noise is considered in this chapter. As model errors have already been dealt when we considered the various mesh sizes, the first two investigations in Sections 7.1-7.2 are conducted using the optimal sensors optimised using the mode-based and mode-free formulations in the absence of stochastic noise. Finally, SPO is conducted in the presence of stochastic noise and conclusions drawn.

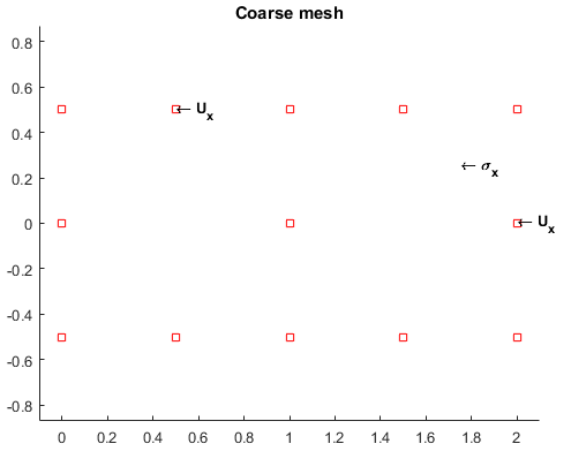
7.1 Varying E and k

What kind of benefits can one get from this experiment by adding stochastic noise for the inverse problems? Firstly, it is possible to simulate the real experiment. Secondly, by doing so, one can find out which systems or problems are more ill-posed or well-posed. If the problem is suddenly unsolvable as soon as some amount of noise is added, then it clearly shows that that problem is ill-posed. It also gives an indication of the signal to noise ratio that can be tolerated which in turn may have bearing on the quality of the sensors to be employed.

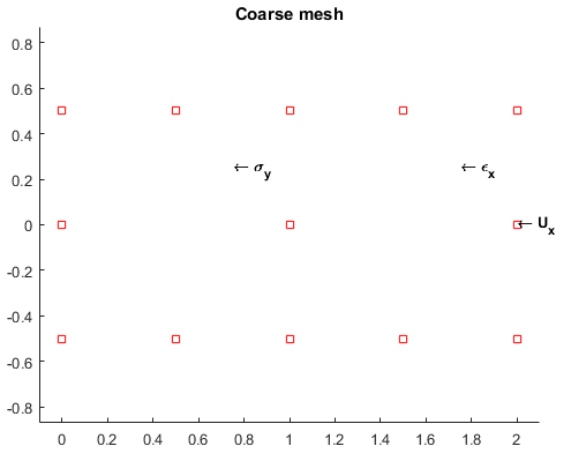
Reconsider the coarse mesh cantilever beam with two springs k_1 and k_2 shown in Figure 3.1. We aim to recover E, k_1 and k_2. However, this time, 1% of normally distributed class noise and point noise are present in the test measurement data in order to investigate how the sensors selected from the noiseless datasets using the mode-based and mode-free formulations perform. For this study, the best performing sensors were chosen from Tables 5.1-5.3 for the mode-based formulations for the KHEL, KHEH and KLEH cases, respectively and they are depicted in Figure 7.1. Note that some of selected sensors by the mode-based formulation such as σ_y close to the root and u_x may not necessarily have large variances compared to other measurements. The sensors chosen from the mode-free formulations are depicted in Figures



(a) KHEL

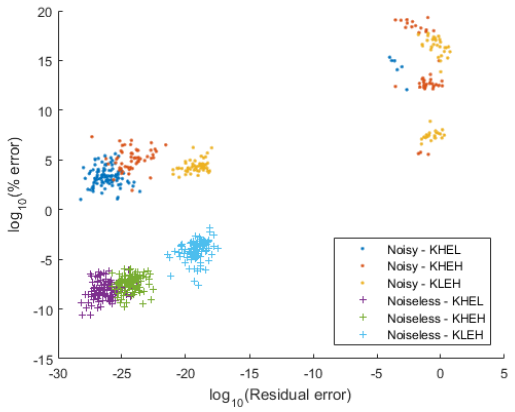


(b) KHEH

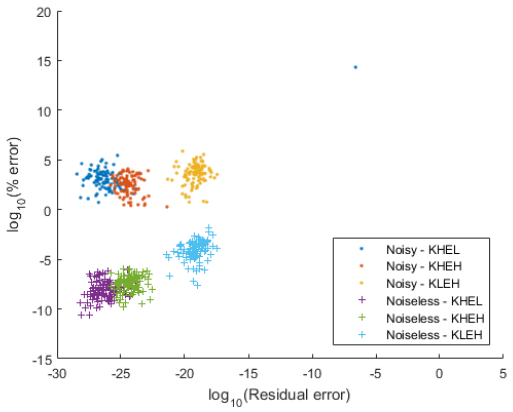


(c) KLEH

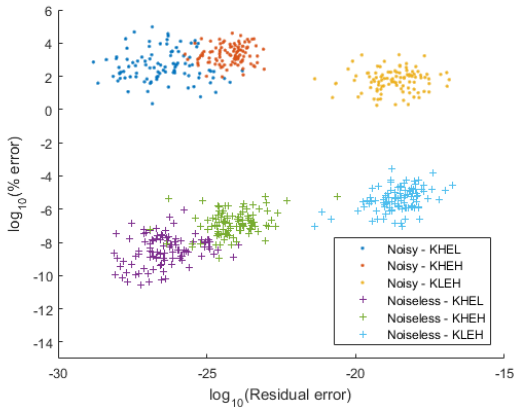
Figure 7.1: Spatial distribution of the best performing sensors from the mode-based formulations for (a) KHEL, (b) KHEH and (c) KLEH.



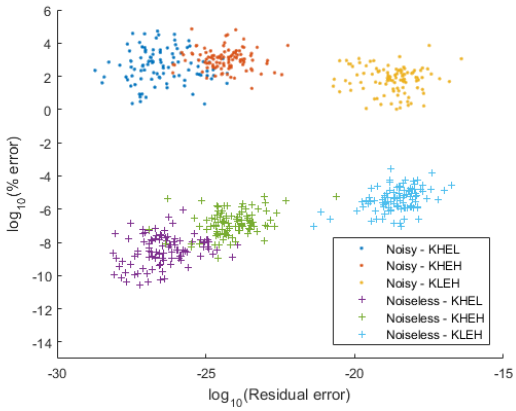
(a) Mode-based formulation and class noise



(b) Mode-based formulation and point noise

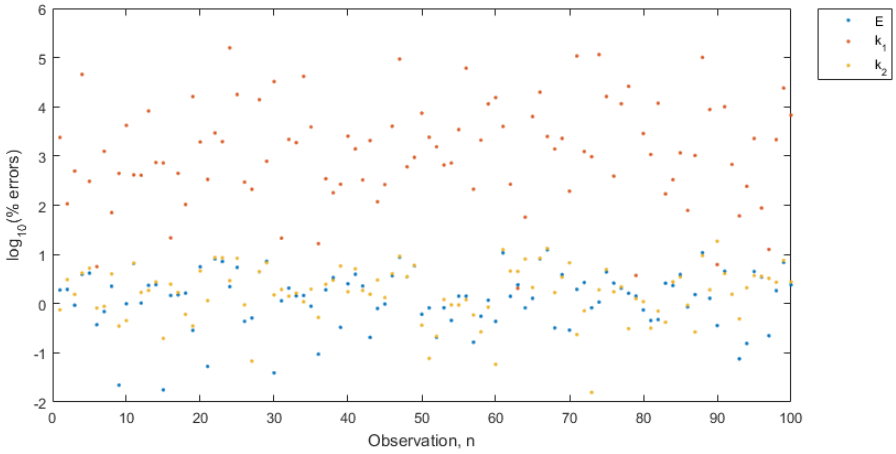


(c) Mode-free formulation and class noise

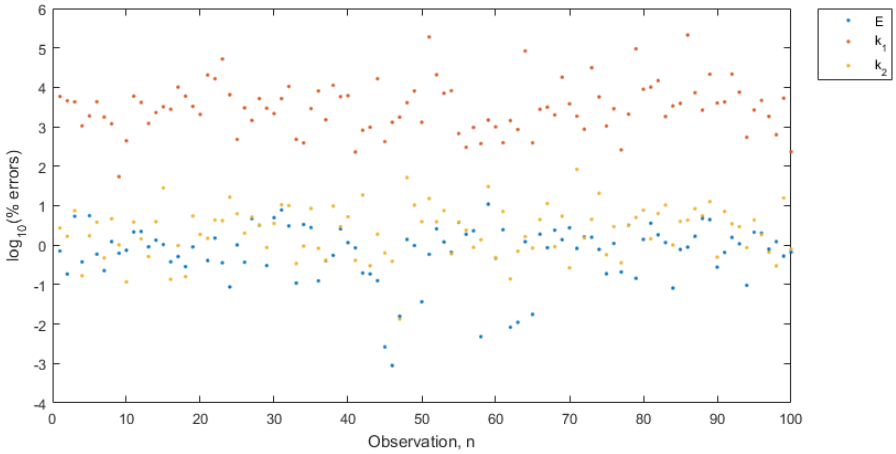


(d) Mode-free formulation and point noise

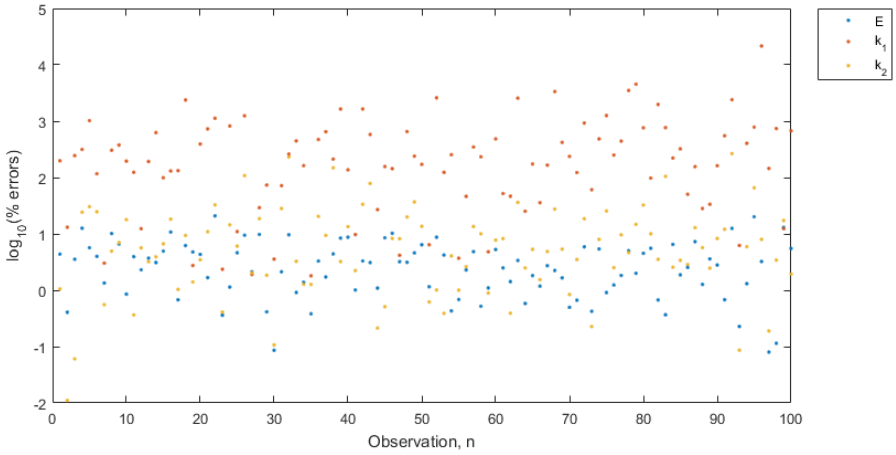
Figure 7.2: Behaviour of sensors in the presence of noise for an ill-posed problem; (a) mode-based formulation and class noise, (b) mode-based formulation and point noise, (c) mode-free formulation and class noise and (d) mode-free noise and point noise.



(a) KHEL



(b) KHEH



(c) KLEH

Figure 7.3: Percentage errors of recovered target variables with class noise of 1% in the test measurement data for (a) KHEL, (b) KHEH and (c) KLEH.

6.3(a), 6.4(a) and 6.5(a). Note that these sensor sets were selected in the previous chapters using the exhaustive combinatorial approach from the noiseless datasets. Figures 7.2(a)-(d) depict the distance from the solution (% error) as a function of the obtained objective value (residual error) after solving for the inverse problem with the noisy data and noiseless data for the selected sensors. Note that both axes are on a log_{10} scale.

In the noisy solutions shown in Figure 7.2, it is evident that the objective function values are small but that the distance to the desired solution is large when compared to the noiseless solutions. This clearly indicates that a 1% variation in the measured response has an effect on orders of magnitude variation on the recovered solution, which formerly makes this problem ill-posed. The difference between the mode-based and the mode-free formulations is that whereas the mode-based formulations often contained the large residual errors (see Figures 7.2(a)-(b)), the mode-free formulations always have small residual errors. Also note that the performance of the mode-based formulations is highly dependent on the types of noise whereas the mode-free formulations show constant performance since it chooses its sensors based on the variance of the sensors. Figure 7.2(a) demonstrates that the mode-based formulation experiences many cases where it had both large % error and residual error when tested with the class noise. However, the significant improvement in the results was observed with the point noise in Figure 7.2(b).

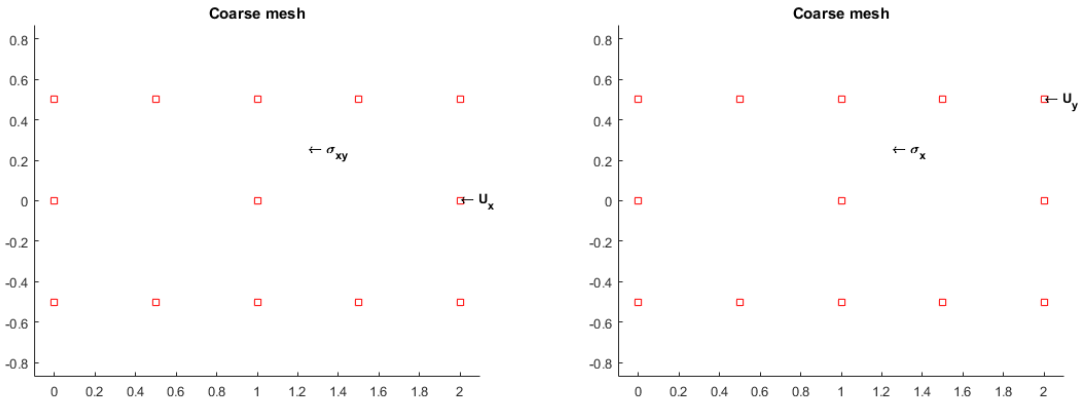
To investigate which target variables are recoverable and which ones are ill-posed, we depict the percentage error for all 100 observations in Figures 7.3 for KHEL, KHEH and KLEH, respectively. In all of the cases, it is clearly noticed the k_1 variable always has the largest percentage errors. Recall that the k_1 is the weaker spring and it has the smallest range among the target variables. The ability to recover k_1 is poor and the primary reason why the problem is ill-posed. The % variation in the other variables is within the expected 1%-2% range for a 1% variation in the virtual experimentally measured response. Although this experiment is unable to recover all three target variables, two of the three can be successfully recovered.

7.2 Effect of noise magnitude

For this investigation, we consider the coarse mesh single spring cantilever from Chapter 2, which is not so ill-posed. The inverse problem is solved in the presence of normally distributed class and point noise with magnitudes that vary between 0.01 and 5% for the KHEH coarse mesh case using the sensors obtained from the exhaustive combinatorial mode-based and mode-free formulations. F2S1 proxy was chosen for the mode-based formulation since it is one of the best performing proxies. Note that these sensors were selected from the noiseless datasets for both formulations and they are depicted in Figure 7.4. The performances of these sensors with varying noise for the less ill-posed problem are shown in Figures 7.5(a)-(b). Notice that the class noise heavily interrupts the performance of the mode-based formulation since the sensors chosen by the mode-based formulation do not always have high variances. Hence, notice from Figure 7.5(a) that the sensors from the mode-free formulation have much less % errors. This is because the mode-free formulation maximises the variance of the sensors. As expected, the inverse problem is more successfully solved as the noise magnitude decreases. Figure 7.5(b) shows the point noise results for which the inverse problem performs better for both the mode-based and mode-free formulations. Again, since the sensors from the mode-free formulation have larger variance compared to the mode-based sensors, they are better suited as the results demonstrate.

7.3 SPO with stochastic noise

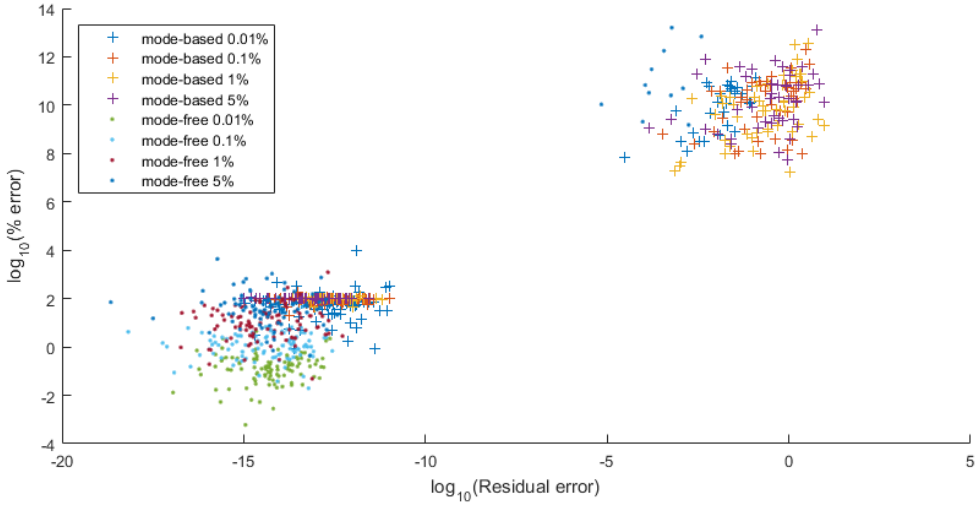
The previous sections considered sensors optimised using the mode-based and mode-free formulations in the absence of stochastic noise. In this section, SPO is conducted in the presence of stochastic noise of 0.01% for the single spring cantilever beam. The aim of



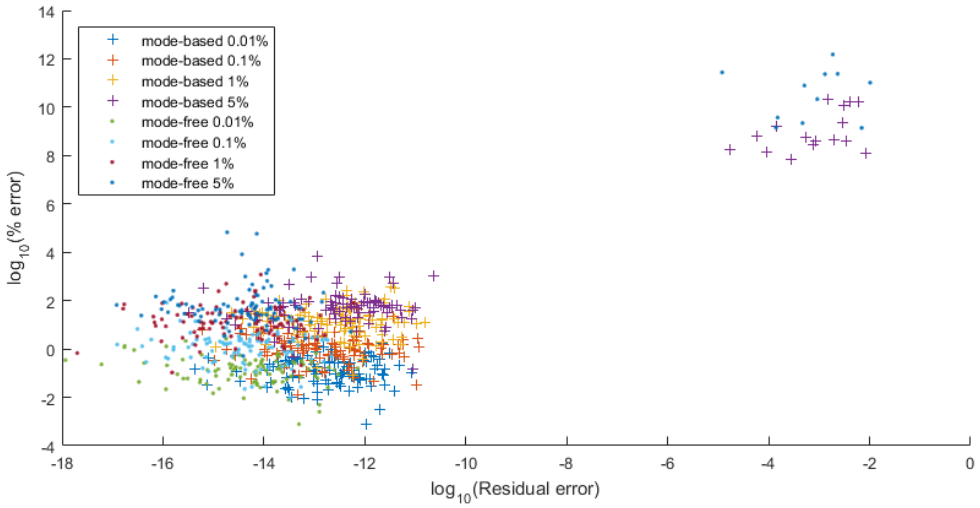
(a) Mode-based formulation

(b) Mode-free formulation

Figure 7.4: Spatial distribution of the sensors selected using (a) mode-based and (b) mode-free formulations for the single spring cantilever beam.

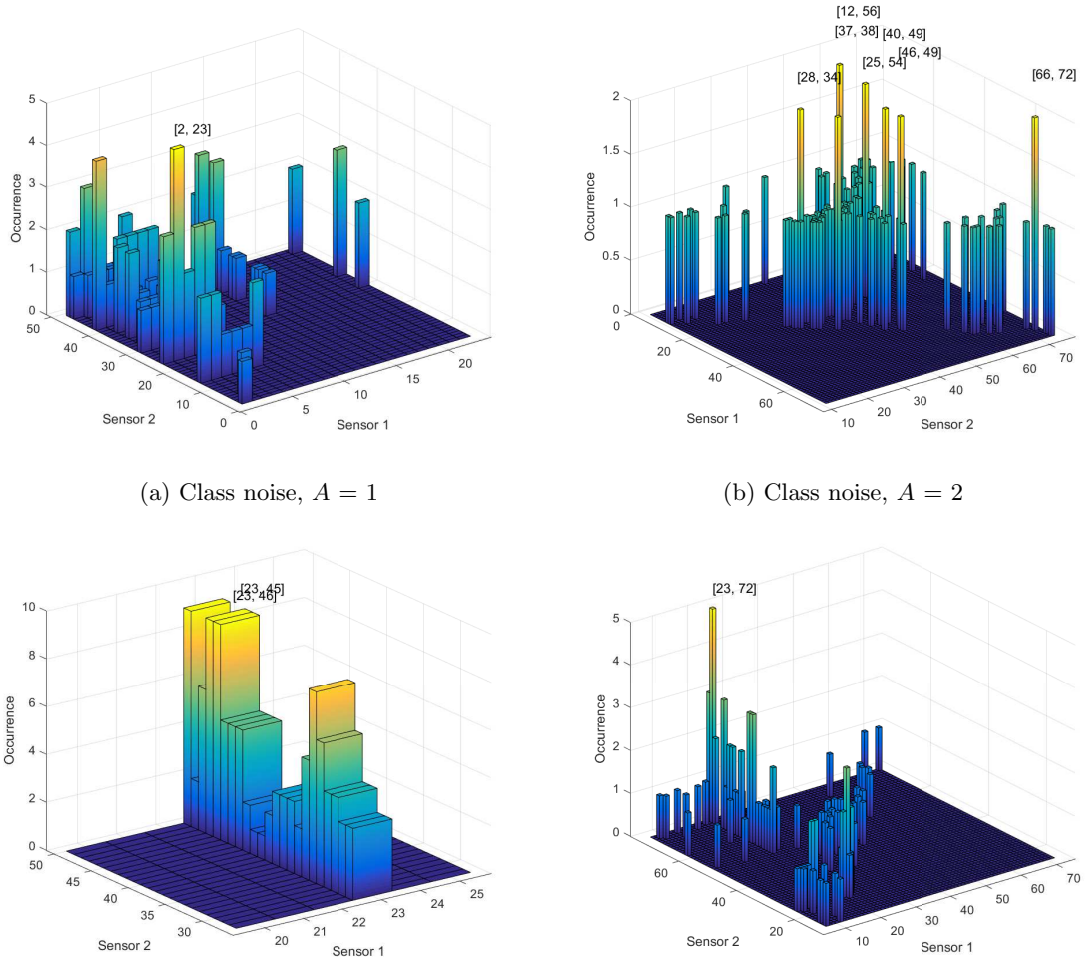


(a) Class noise



(b) Point noise

Figure 7.5: Behaviours of mode-based sensors and mode-free sensors with (a) class noise and (b) point noise for a less ill-posed problem.



(a) Class noise, $A = 1$

(b) Class noise, $A = 2$

(c) Point noise, $A = 1$

(d) Point noise, $A = 2$

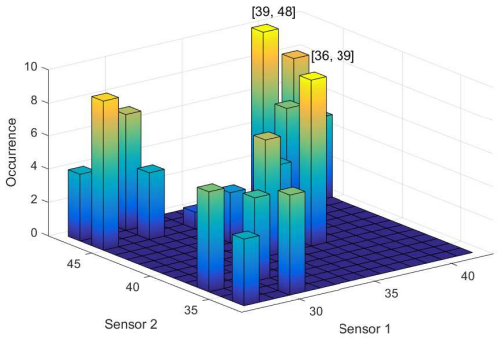
Figure 7.6: Mode-based formulation results for class noise and point noise of 0.01%; (a) class noise and $A = 1$, (b) class noise and $A = 2$, (c) point noise and $A = 1$ and (d) point noise, $A = 2$.

Method	F2S1	F2S1	Mode-free	RTMF
Mode	A = 1	A = 2		
Sen. 1	2	25	39	26
Sen. 2	23	54	48	33
Acc.	**0**	**0**	**0**	**96**
Iter.	**0**	**0**	**0**	**122.22**

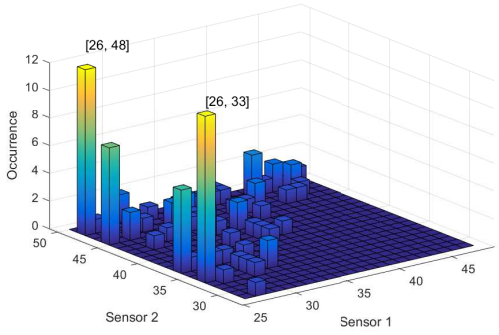
Table 7.1: Performances of sensors found with noise on class noise of 0.01%

Method	F2S1	F2S1	Mode-free	RTMF
Mode	A = 1	A = 2		
Sen. 1	23	23	26	26
Sen. 2	46	72	48	33
Acc.	**96**	**1**	**96**	**98**
Iter.	**121.3**	**134**	**122.94**	**120.97**

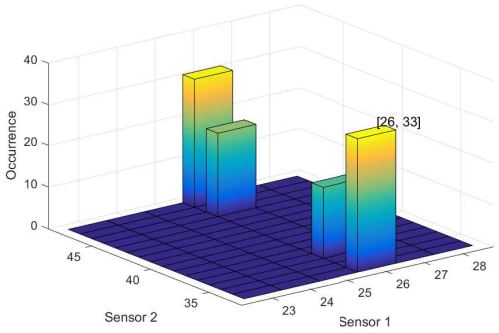
Table 7.2: Performances of sensors found with noise on point noise of 0.01%



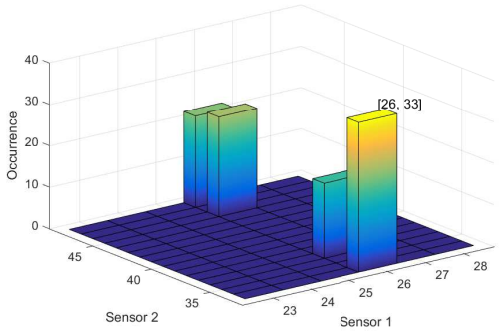
(a) Class noise, mode-free formulation



(b) Point noise, mode-free formulation



(c) Class noise, RTMF



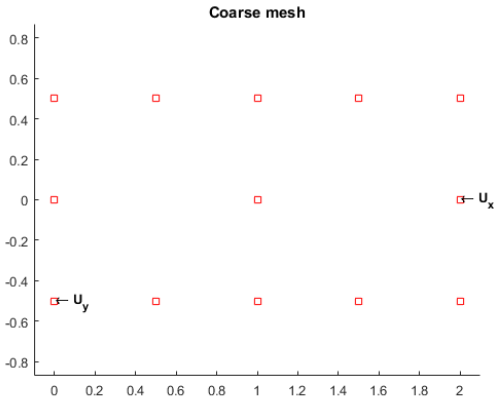
(d) Point noise, RTMF

Figure 7.7: Mode-free formulation results for class noise and point noise of 0.01%; (a) class noise and mode-free formulation, (b) point noise and mode-free formulation, (c) class noise and RTMF and (d) point noise and RTMF.

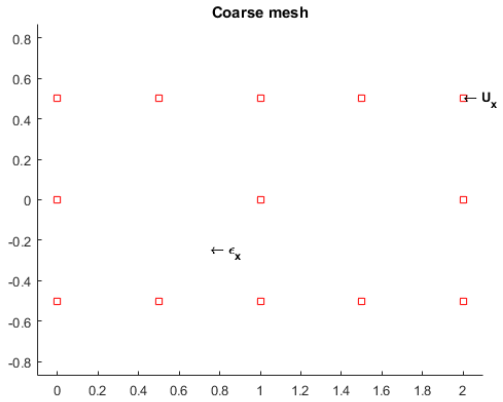
this study is to investigate if the placement of sensors optimised with stochastic noise is any different and if it is, how it performs with the noisy test measurement data. Note that the noise considered here is normally distributed and the amount of noise involved for optimising the sensors matches the noise added to the measurement data for the inverse problem. Since the stochastic noise is random, it makes the optimised sensors inconsistent. Hence, the optimised sensors were chosen from the most frequent results after 100 independent optimisation runs. Figures 7.6 and 7.7 show the 100 runs for the mode-based and mode-free formulations for both class noise and point noise as histograms and the ID numbers of the most frequent sensor sets are shown in the figures.

To avoid having various results and increase the convergence of the results, the new rank tolerance mode-free formulation (RTMF) is introduced and this adjusts the rank tolerance of the two rank constraint functions (Eqs. 6.1 and 6.2) to $tol = 10^{ln(S_1)}$ where S_1 is the first singular value of the covariance matrices of the measurement and target datasets. The tolerance function simply changes the base of S_1 from exp to 10. This allows for a larger rank tolerance which makes rank counting less sensitive to the stochastic noise. Figure 7.6 shows the sensor results for F2S1 which is one of the best performing mode-based formulations. It is clear that the sensors are often not consistent for the class noise when $A = 1$ and 2. The point noise results are more organised and consistent. Figure 7.7 shows the sensor optimised results for the original mode-free formulation (Figures 7.7(a)-(b)) and RTMF (Figures 7.7(c)-(d)). The figures again show extremely inconsistent sensor selections in the class noise but more consistent results for the point noise for the original mode formulation. This again demonstrates that the point noise problems are more well-posed than the class noise problems. On the other hand, it is noticed that the consistency of the sensor selections for RTMF is much higher.

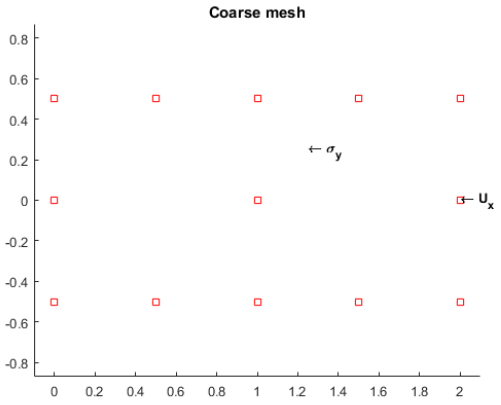
The most frequent sensors selected from each formulation were spatially plotted in Fig-



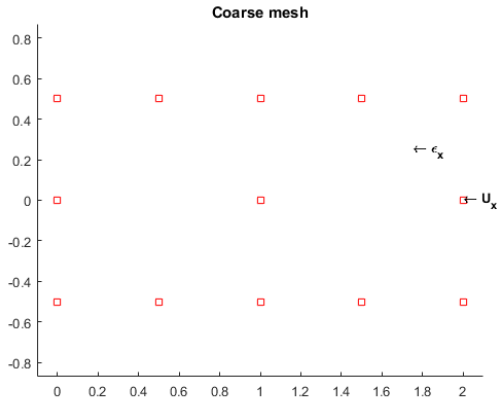
(a) Class noise, $A = 1$



(b) Class noise, $A = 2$

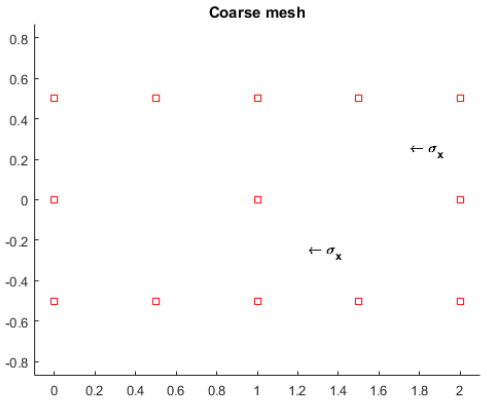


(c) Point noise, $A = 1$

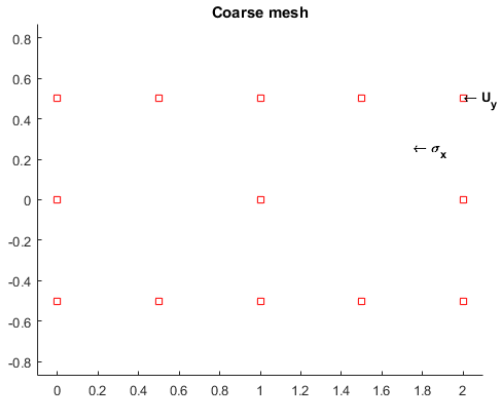


(d) Point noise, $A = 2$

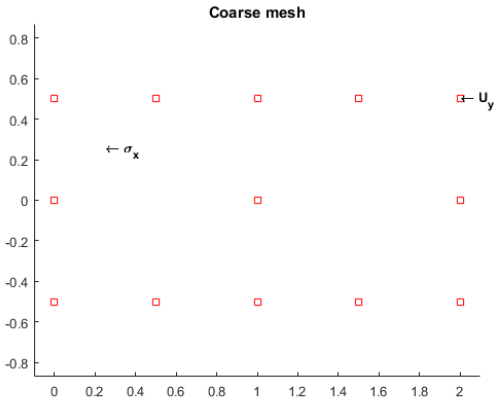
Figure 7.8: Spatial distribution of the sensors selected from the mode-based formulations; (a) class noise and $A = 1$, (b) class noise and $A = 2$, (c) point noise and $A = 1$ and (d) point noise and $A = 2$.



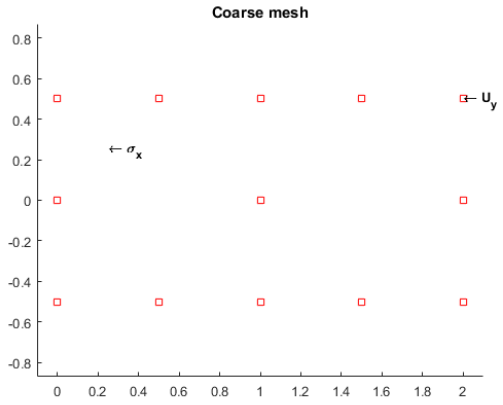
(a) Class noise, mode-free formulation



(b) Point noise, mode-free formulation



(c) Class noise, RTMF



(d) Point noise, RTMF

Figure 7.9: Spatial distribution of the selected sensors from the mode-free formulations; (a) class noise and mode-free formulation, (b) point noise and mode-free formulation, (c) class noise and RTMF and (d) point noise and RTMF.

ures 7.8 and 7.9. Note that when there are more than one equally frequent sensor sets for a formulation such as shown in Figure 7.6(b), single set was randomly chosen from the outcomes and tested. Stochastic noise of 0.01% is small and yet, it is noticed that none of the mode-based formulation results depicted in Figure 7.8 look similar to the sensors selected with noiseless optimisation depicted in Figure 7.4(a). On the other hand, the mode-based formulation results shown in Figure 7.9 show a high similarity to the sensors selected with noiseless optimisation. Especially, the sensors from the original mode-based formulation with point noise depicted in Figure 7.9(b) look almost identical to the noiseless optimisation result except the σ_x sensors which moves slightly towards the tip of the beam. Note and remember that the sensors from RTMF for class noise and point noise are identical to each other and have its σ sensor located close to the root as depicted in Figure 7.9(d).

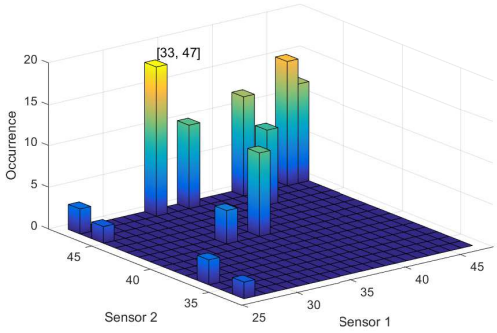
Tables 7.1 and 7.2 show the performance of the sensors optimised for the class and point noise, respectively. It is noticed that RTMF shows promising results against the class noise when the other formulations fail. RTMF also shows the best results against the point noise, even better than the original mode-based formulation with point noise which almost looked identical to the sensors selected from the noiseless datasets.

7.4 Performance comparisons

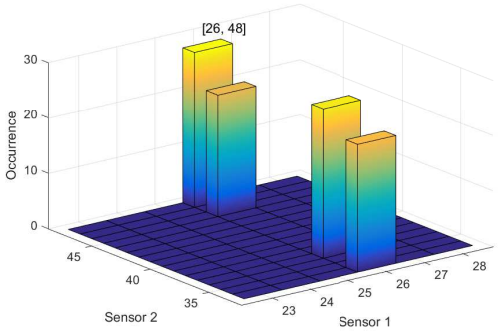
In the previous section, the performance of RTMF was demonstrated to be the most promising against both class noise and point noise. RTMF has also shown its improved consistency with optimising sensors from the noisy datasets. Hence, RTMF specifically was tested with varying magnitude of noise in this section. The same procedure as before was followed for 0.1, 1 and 5% noise for both class noise and point noise. The occurrences of the results over 100 runs are depicted in Figure 7.10. Firstly, the class noise results presented in Figures 7.10(a), (c) and (e) have less consistency in the selection whereas the point noise results shown in Figures 7.10(b), (d) and (f) are more consistent.

The sensors selected for the various noise are depicted in Figure 7.10. Note that the spatial distribution of the sensors for the class noise depicted in Figures 7.11(a), (c) and (e) shows that their sensors only measure stress whereas for the point noise depicted in Figures 7.11(b) and (d) also measure displacement except when the point noise is 5%, the noise starts to dominate the problem which results in selection of insensible sensors at the fixed nodes as shown in Figure 7.11(f).

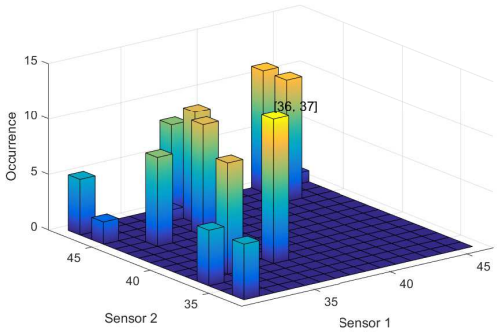
The performance of the original mode-free formulation for both noiseless and noisy dataset optimisation was compared to the performance of RTMF when varying the magnitude of noise as shown in Tables 7.3 and 7.4 for class noise and point noise, respectively. It clearly demonstrates that the sensors optimised with class noise perform much worse than those optimised without any noise. However, RTMF could still perform when the noise is relatively small. The performance of the sensors with point noise (Table 7.4) shows that whereas the original mode-free formulation with noisy optimisation fails from the small amount of test measurement noise, RTMF performs almost as well as the mode-free formulation with noiseless optimisation. Hence, the result shows that it is the better choice not to optimise sensors with noise but when the noise is involved, RTMF is highly recommended for the inverse problem.



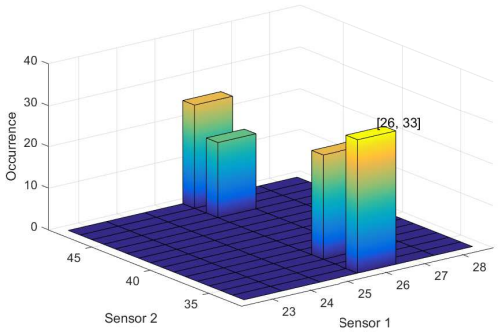
(a) Class noise, 0.1%



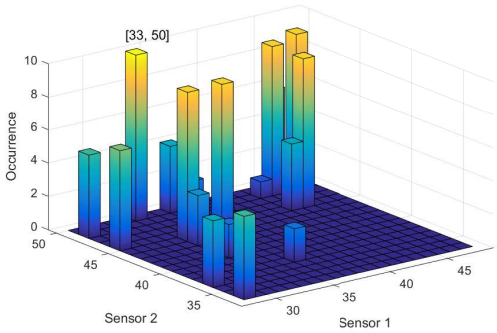
(b) Point noise, 0.1%



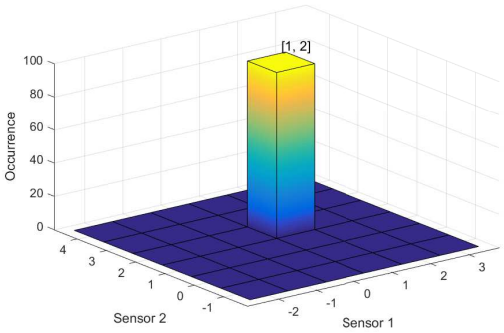
(c) Class noise, 1%



(d) Point noise, 1%

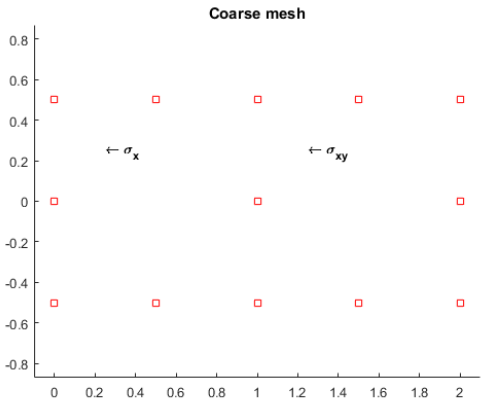


(e) Class noise, 5%

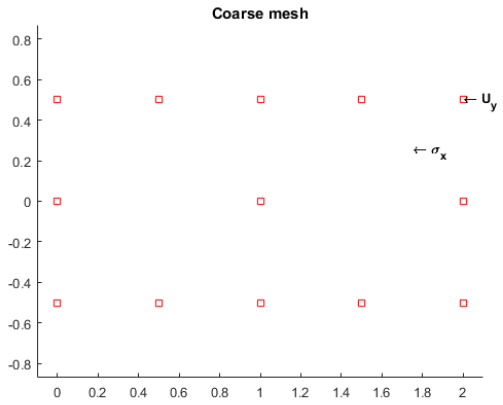


(f) Point noise, 5%

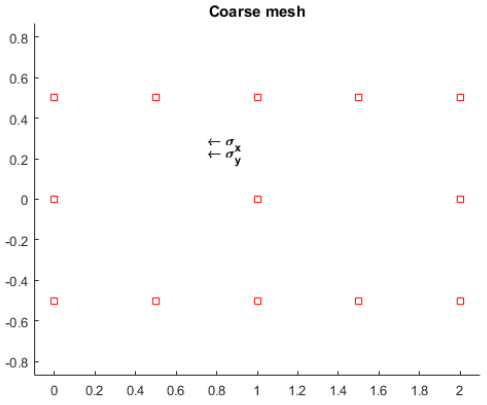
Figure 7.10: RMTF results for 0.1% (a) class noise, (b) point noise, 1% (c) class noise, (d) point noise, 5% (e) class noise and (f) point noise.



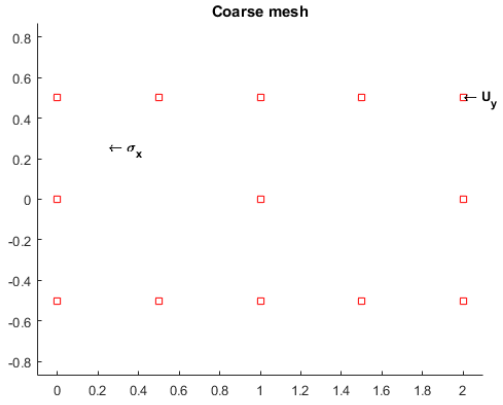
(a) Class noise, 0.1%



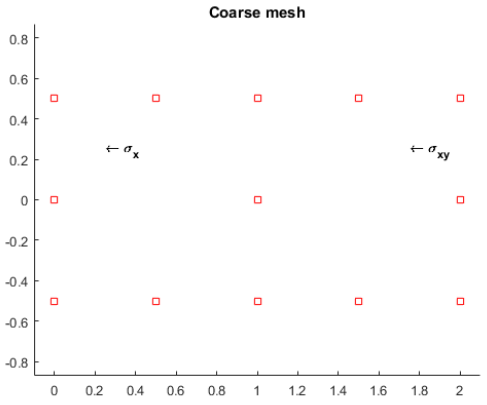
(b) Point noise, 0.1%



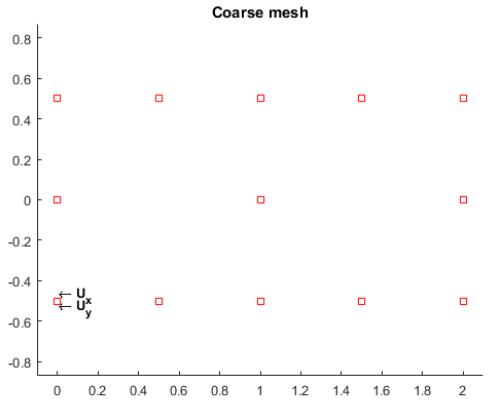
(c) Class noise, 1%



(d) Point noise, 1%



(e) Class noise, 5%



(f) Point noise, 5%

Figure 7.11: Partial distribution of the RMTF results for 0.1% (a) class noise, (b) point noise, 1% (c) class noise, (d) point noise, 5% (e) class noise and (f) point noise.

Method	Mode-free		RTMF
Noise	Optimisation without noise	Optimisation with noise	Optimisation with noise
0.01%	96	0	96
0.10%	42	0	0
1%	7	0	0
5%	2	0	0

Table 7.3: Class noise results for varying noise magnitude

Method	Mode-free		RTMF
Noise	Optimisation without noise	Optimisation with noise	Optimisation with noise
0.01%	94	96	98
0.10%	41	0	35
1%	6	0	4
5%	2	0	0

Table 7.4: Point noise results for varying noise magnitude

Chapter 8

Simplified Mode-Free Formulation

This chapter briefly introduces simplified mode-free formulation (SMF). The critical learnings from the studies about the mode-based formulations led to the development of the mode-free formulations and the issues when stochastic noise is added to the training data could be resolved by adjusting the tolerances of the rank computations. During the detailed research about the mode-based formulation, it was found that slight modification of the covariance rank function (Eq. 6.1) can surpass the features presented by combining the two functions (Eq. 6.1 and 6.2) used for the constrained optimisation problem with a single objective for maximising the product of variance of the sensors. The key to SMF is to adjust the tolerances of the rank of only the Z-score transformed covariance matrix between the measurement and target datasets (X^TY) until there is only one or few sensor sets left to satisfy the full rank. In this way, SMF can find the sensor sets where each sensor has its own maximum uniqueness in the correlation with the target variables. It may look equivalent to RTMF which simply increases the tolerance of the rank function but it is done slightly differently. Therefore, this section will show why they are different and how SMF performs.

8.1 Principles behind SMF

Since SMF uses the Z-score transformed covariance matrix, the covariance matrices show correlations between the measurement variables and the target variables. The number of rows of the correlation coefficient matrix (or Z-transformed covariance matrix) corresponds to the number of variables in X_c (the sensors used), while the number of columns is the number of the variables in Y (the target variables). Therefore, each column shows how well these sensors are related to each target variable. The maximum possible rank which can be obtained from the covariance matrix is simply the number of variables in Y, since the number of the sensors should not be less than the number of the target variables. Hence, as SMF tries to maximise the rank tolerance of the normalised covariance and still look for the sensor sets with the maximum rank, it eventually finds the sensor sets that are uniquely correlated to the target variables to avoid collinearity within the covariance matrix. Abstractly, we define the distance d as a measure of how uncorrelated each column of the covariance matrix is to others as presented in Figure 8.1. SMF aims to maximise the distances by simply maximising the rank of the tolerance.

SMF is a modified version of one of the rank constraints in the original mode-free formulation, Eq. 6.1. SMF also naturally maximises the uniqueness of the sensors, since the uniqueness of the correlation of each sensor to the target variables is maximised, while the target variables are fixed with the sensors as variables. Hence, SMF performs the same task as the two rank constraints, 6.1 and 6.2. However, the disadvantage of this formulation is that since SMF does not choose sensors based on their variance, the selected sensors may be sensitive to noise.

In comparison, while the original mode-free formulation and RTMF are constrained optimisations, SMF is an unconstrained optimisation. SMF does not require an exhaustive

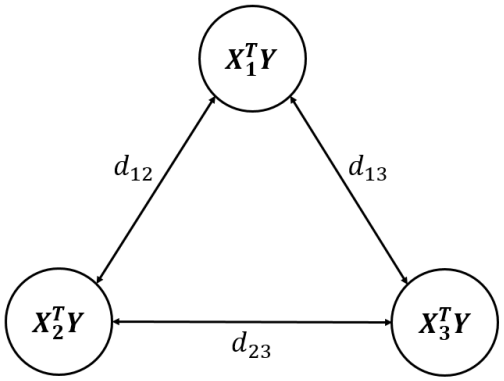


Figure 8.1: Uniqueness of the sensors and correlations.

Method	Simplified mode-free			
Noise	Class noise		Point noise	
	Optimisation without noise	Optimisation with noise	Optimisation without noise	Optimisation with noise
0.01%	2	2	97	97
0.10%	0	0	49	38
1%	0	0	4	3
5%	0	0	1	1

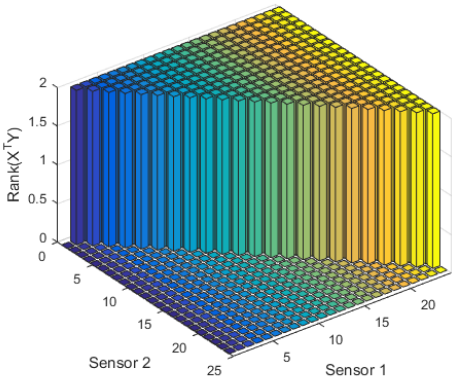
Table 8.1: Performance of SMF with measurement noise

combinatorial approach to solve the optimisation problem, but it does require the tolerance to be increased until rank of the covariance matrix is full with a minimal sensor set. This problem can be solved efficiently.

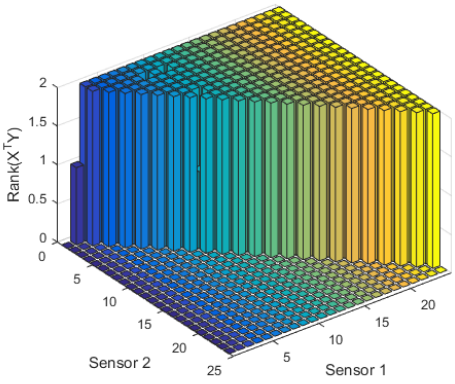
8.2 Performance of SMF

SMF formulation was tested on the single spring cantilever beam with the reduced number of sensors depicted in Figure 2.12. The test results shown in Figure 8.2 take various tolerances for SMF: 10^{-9}, 10^{-8}, 10^{-6}, 10^{-5}, 5.71328 and 5.72. It is noticed that even slight changes in the tolerance can eliminate the sets of sensors which have collinear information. Figure 8.2(c) with tol. $= 10^{-6}$ looks almost similar to the success rates shown in Figure 2.13. Figure 8.2(d) with tol. $= 10^{-5}$ shows that only the $< \sigma, \epsilon >$ and $< \sigma, u >$ combinations have full rank of two. As the tolerance is iterated to have only one full rank set as shown in Figure 8.2(e), it is noticed that the sensor set shown there is indeed the best sensor found using the inverse problem with 100% success rates and the number of iterations of 119.14 which is the global minimum. As the tolerance is increased just a bit to tol. $= 5.72$, there were no full rank sets of sensors remaining (Figure 8.2(f)).

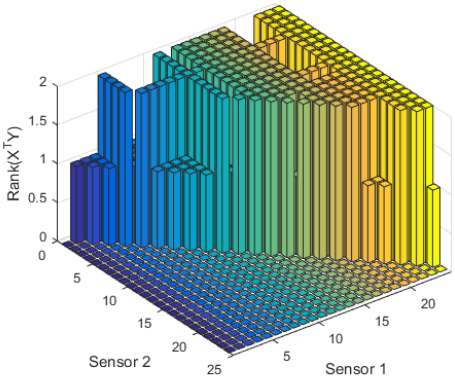
The performance of SMF for both class noise and point noise in the measurement data with and without noise during the sensor optimisation is shown in Table 8.1. Although it was demonstrated that SMF is capable of finding the best solution for the inverse problem when there is no measurement noise, since SMF omits the variance maximisation term which is essential for the class noise, it is noticed that SMF struggles to recover the target variables with less than 1% error, even with the smallest amount of class noise tested. However, SMF is still capable of finding the useful sensors for the point noise. The performances for the point noise for both the noisy and the noiseless training data are as good as or slightly better than those for the original mode-free formulation and RTMF (Tables 7.3 and 7.4).



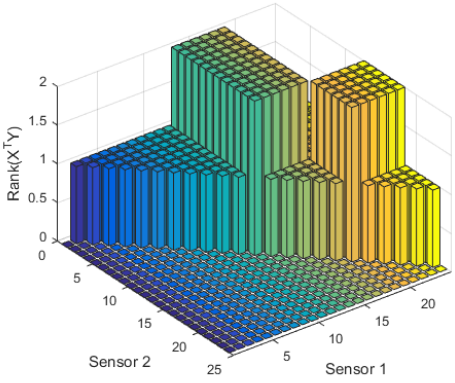
(a) Tol. $= 10^{-9}$



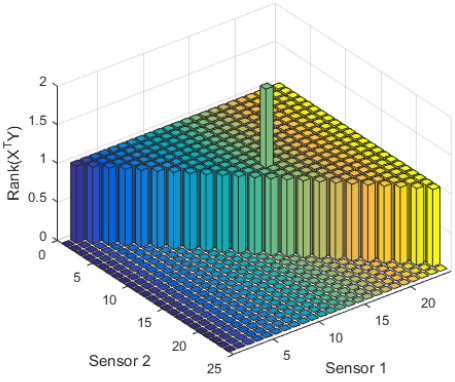
(b) Tol. $= 10^{-8}$



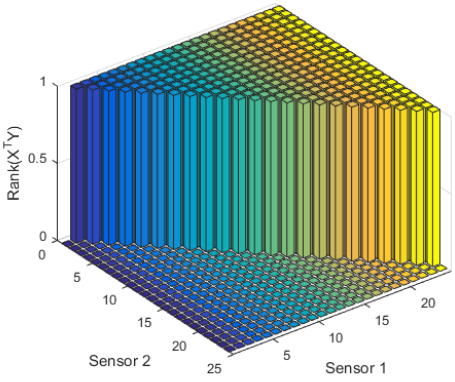
(c) Tol. $= 10^{-6}$



(d) Tol. $= 10^{-5}$



(e) Tol. $= 5.71328$



(f) Tol. $= 5.72$

Figure 8.2: SMF results of the single spring cantilever beam problem with rank tolerance of (a) 10^{-9}, (b) 10^{-8}, (c) 10^{-6}, (d) 10^{-5}, (e) 5.71328 and (f) 5.72.

Chapter 9

Conclusions

Sensor placement optimisation (SPO) strategies for designing experiments using simulations have been investigated and developed in this study. SPO scrutinises the relationships between the sensor measurement variables and the target variables, which are the variables to be characterised, in order to optimally place sensors. This therefore eliminates the need for trial and error approaches to design complicated inverse experiments, saving both time and effort, by ensuring the most sensitive output from the measurements as well as uniqueness of information. This increases the chances to successfully solve inverse problems. In practice, the benefit is two-fold, it saves time for an experienced engineer to design an inverse experiment but more importantly a less experienced engineer can now design an inverse experiment that performs well.

SPO formulations in this study are categorised into two groups; namely, mode-based formulations and the mode-free formulations. Mode-based formulations use either SIMPLS or SVD to extract the modes which explain the causalities between the measurement variables and the target variables. The study has shown that maximising the variance explained in the target variable data using SIMPLS is the most robust mode-based formulation among the objective functions considered. Similarly, minimising the regression error of SIMPLS also proved to be robust. Scaling of the data was also considered but here the regression error proved to be highly sensitive to scaling of the data. In cases where the performance of the mode-based formulations was poor, additional sensors usually improved the success rate of solving the actual inverse problem as well as decreasing the required number of iterations to solve the inverse problem. As modes need to be extracted, the mode-based methods usually scale linearly with the number of sensors to be placed. Hence, the main concerns of the mode-based methods are:

1. formulations are sensitive to data scaling,
2. the number of modes needs to be determined,
3. computational cost can be high when a large number of sensors need to be optimally placed.

The mode-free formulations were developed to resolve the main issues of the mode-based formulation by examining the critical learnings obtained from the studies of the mode-based formulations. Since the mode-free formulation does not require extracting modes, its computational speed is almost ten times faster when combinatorial approaches are used. The proposed addition approach requires six orders less computational time than the mode-based combinatorial methods to select three sensors from over a thousand available sensor locations. The first proposed mode-free formulation selects sensors by checking the ranks of covariance matrix between the two datasets in order to measure the uniqueness of the correlations. Secondly, the ranks of covariance matrix X^TX are maximised so that information measured between the sensors is unique. Instead of formulating a multi-objective problem, the two objectives related to rank are used as constraints in the optimisation formulation resulting in

a constrained optimisation problem with a single objective. The objective here is to maximise the product of the variance of each selected sensor. The mode-free formulation was not able to find effective sensors when the training data were noisy but this issue was resolved by introducing the rank tolerance mode-free (RTMF) formulation. This approach adjusts the tolerances when computing the rank of the covariance matrices. By increasing the tolerance, the ranks are better distinguished and leads to an improved SPO, referred to as RTMF. This concept was refined and SMF developed.

Two types of noise were considered; namely, point noise and class noise. Point noise refers to stochastic noise that is a certain percentage of the variance of each individual sensor, while class noise refers to experimental noise that contributes an equal amount of noise to all sensors in the same sensor class. The simplified mode-free formulation (SMF) is capable of finding the optimal sensors and is highly correlated with the performance of solving the actual inverse problem. It shows outstanding performance with and without the presence of point noise. However, since SMF does not maximise the variance of the sensors, it struggled in the presence of class noise. Although the various formulations can be implemented for different cases, RTMF is the most robust against both point and class noise. The addition optimisation approach proved to be computationally efficient, although not always optimal. Therefore, an additional sensor often gives near optimal performance still within a computational tractable time.

For future work, the convexification of mode-free formulations will be considered using the least absolute shrinkage and selection operator (Lasso) [52], which regularises linear regression such that variable selection is performed. Another approach to convexify the problem would be to maximise the trace (or nuclear) norm of the covariance matrix, since the trace norm, which is defined as the sum of singular values, is an excellent proxy of the matrix rank while the associated optimisation problem is convex [53, 54].

Chapter 10

Appendices

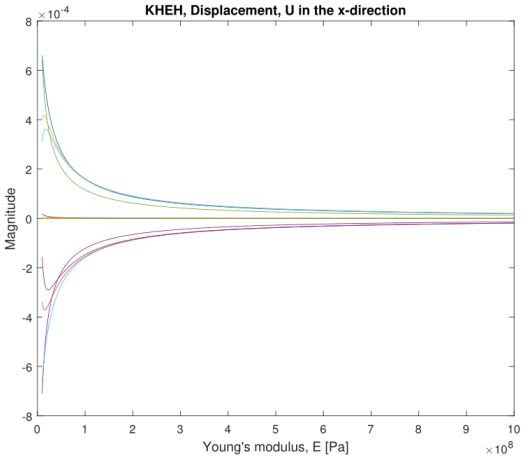
10.1 Graphs explaining causality

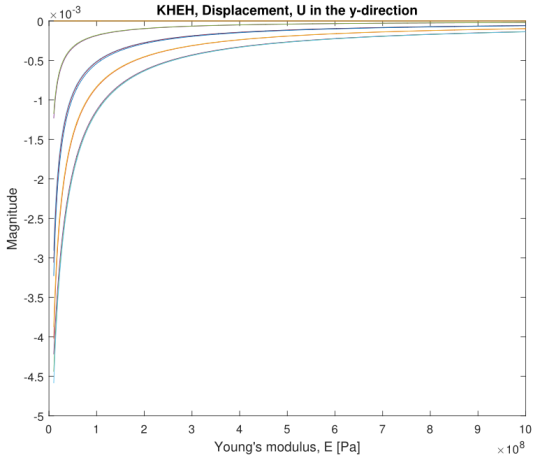
10.1.1 Displacements, u

Displacements measured for varying E, k_1 and k_2 individually are shown in Figures 10.1, 10.2 and 10.3, respectively. The same phenomena as in section 5.4 are observed from the figures. Therefore, the same interpretation applies to this section.

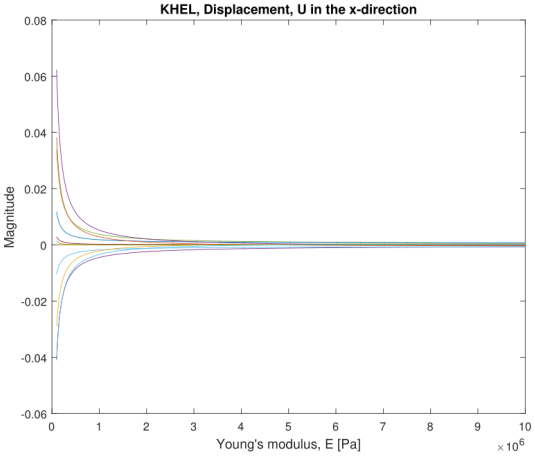
10.1.2 Strains, ϵ

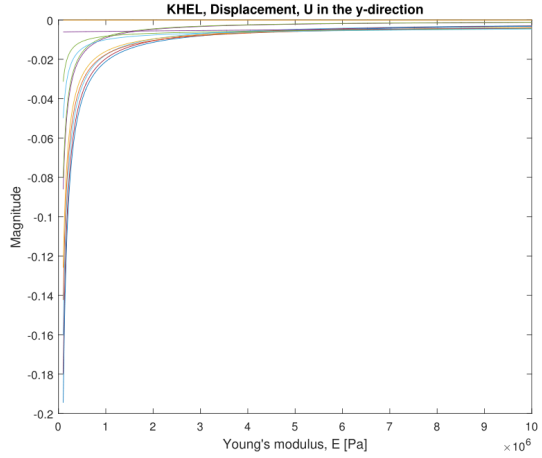
Strains measured for varying E, k_1 and k_2 individually are shown in Figures 10.4, 10.5 and 10.6. The same phenomena as in section 5.4 are observed from the figures. Therefore, the same interpretation applies to this section.



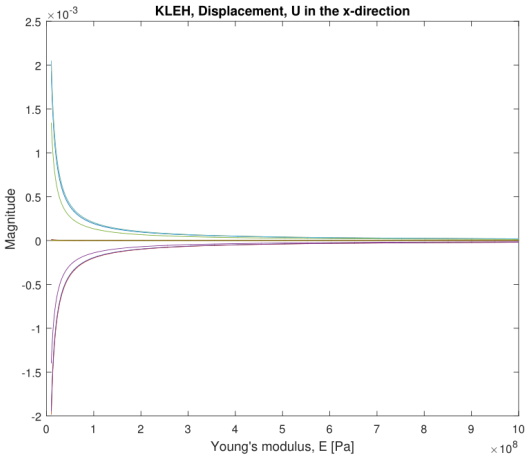


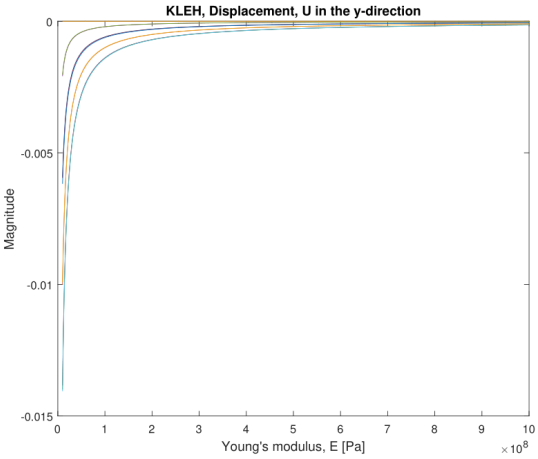
(a) KHEH





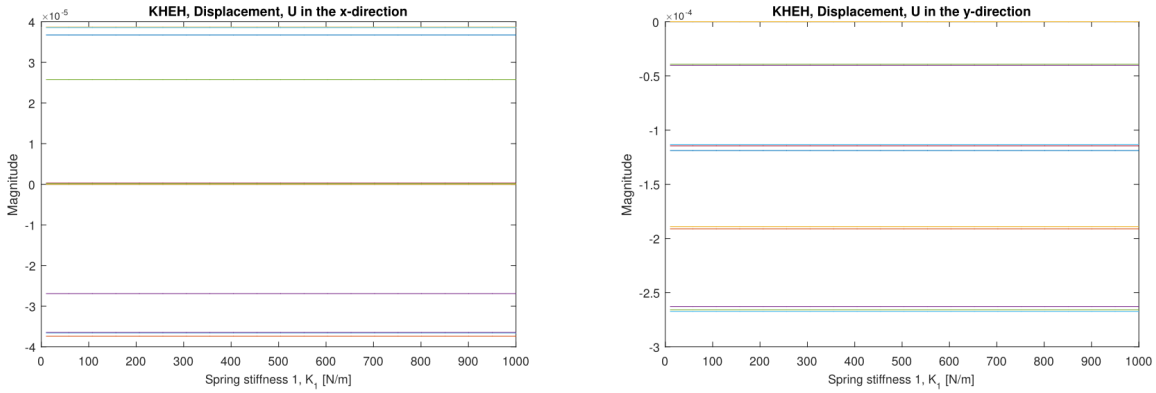
(b) KHEL



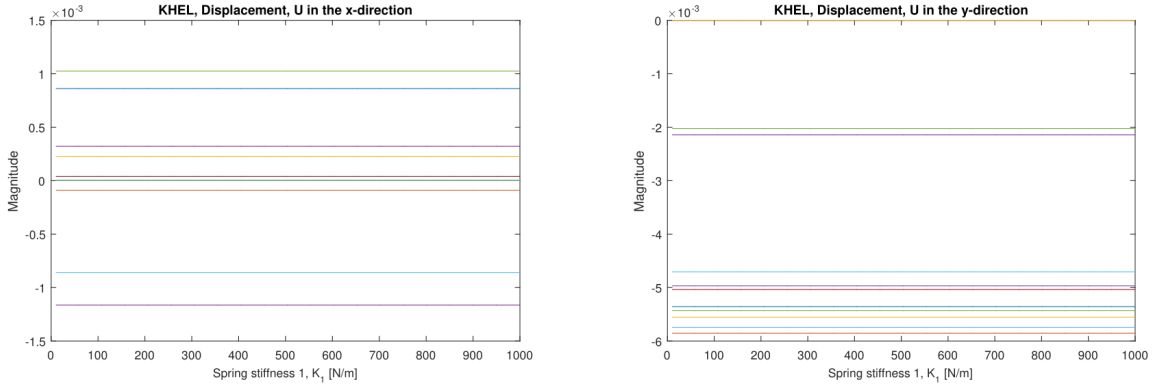


(c) KLEH

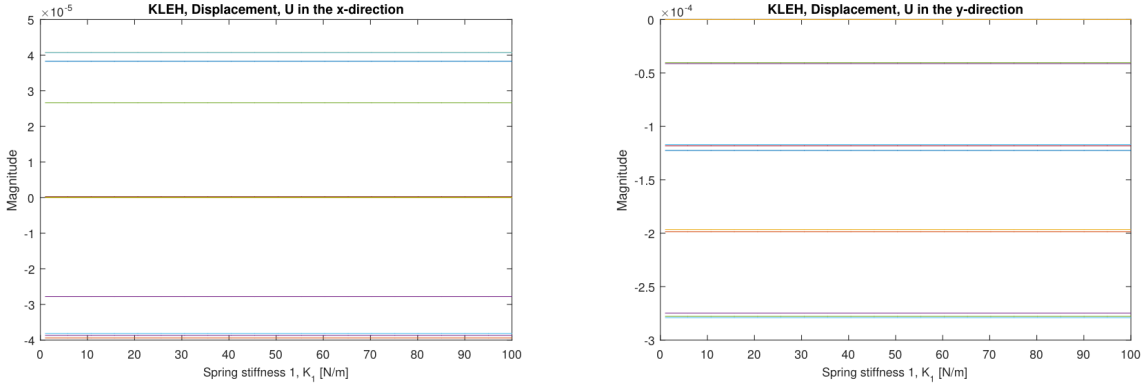
Figure 10.1: Displacement, u sensitivities when adjusting Young's modulus, E.



(a) KHEH

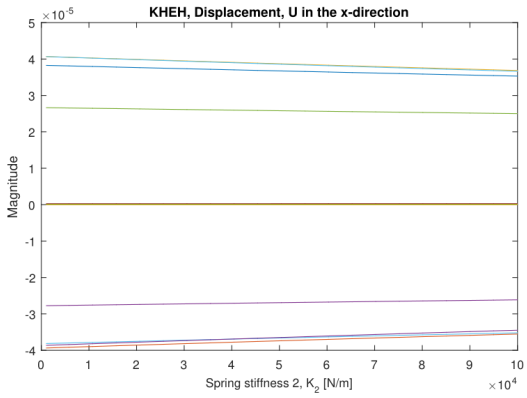


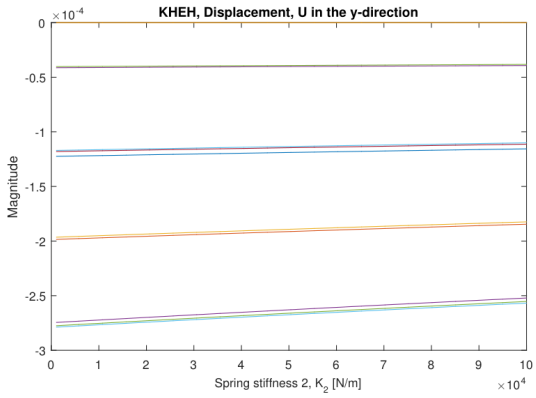
(b) KHEL



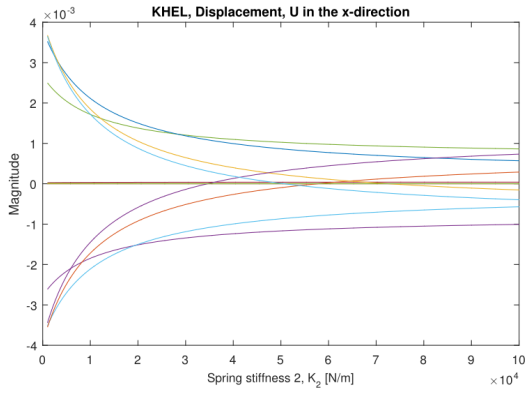
(c) KLEH

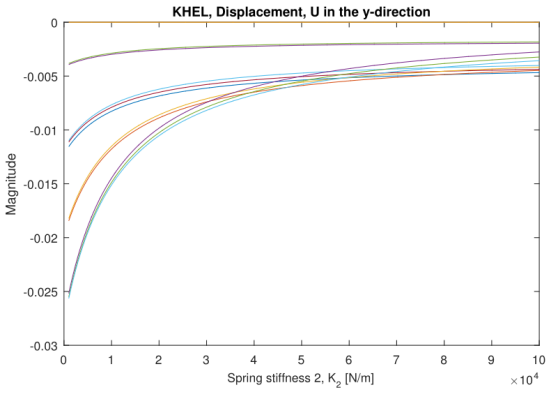
Figure 10.2: Displacement, u sensitivities when adjusting spring stiffness 1, k_1.



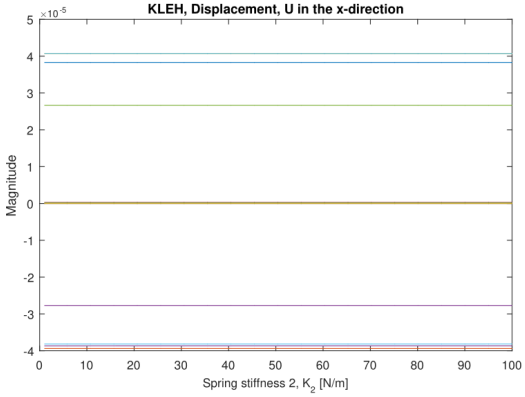


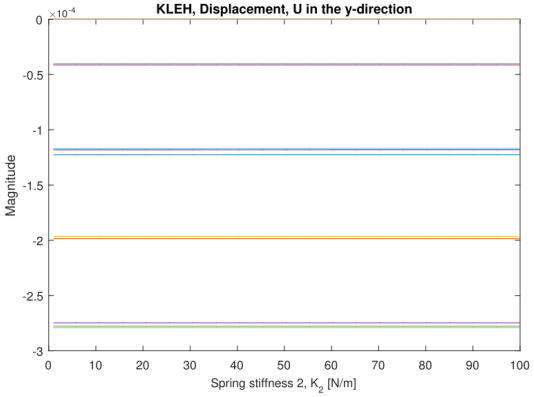
(a) KHEH





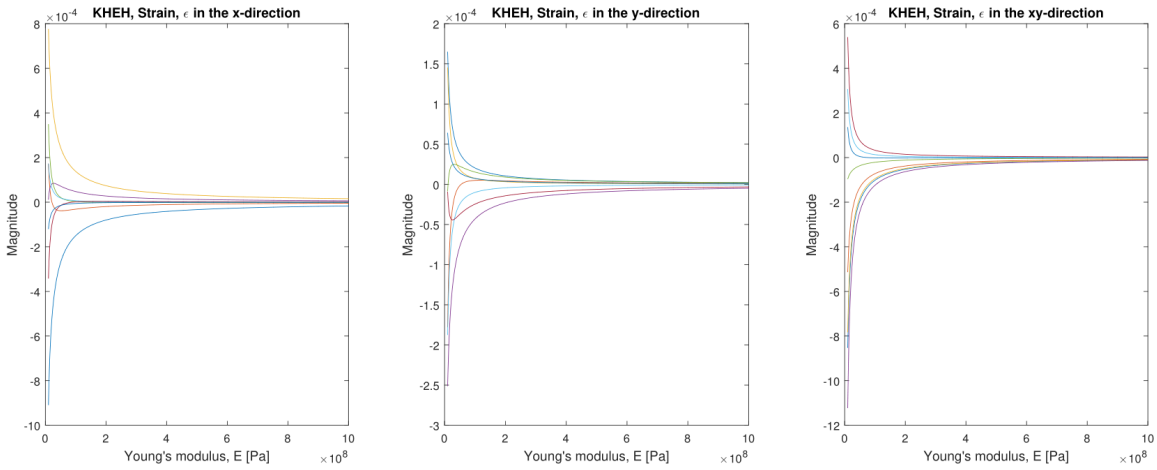
(b) KHEL



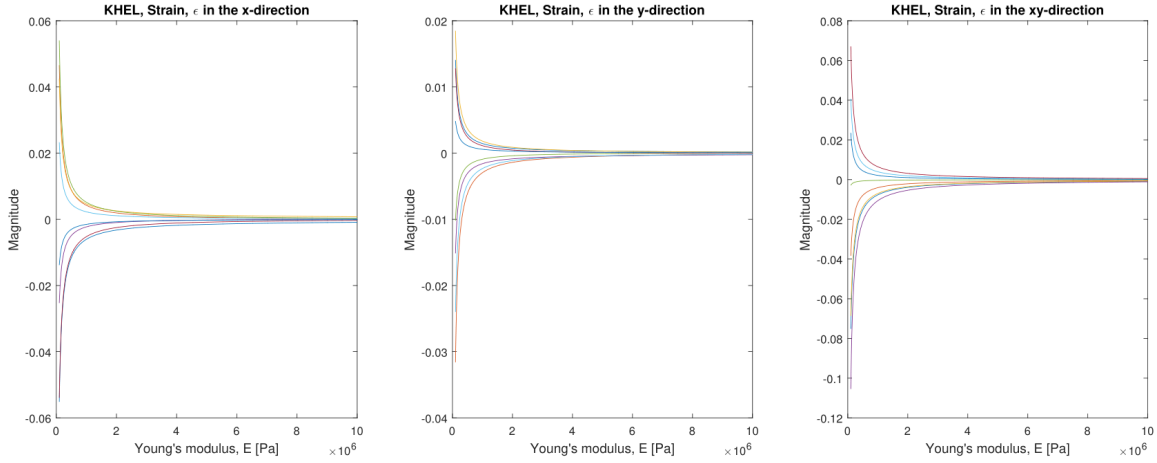


(c) KLEH

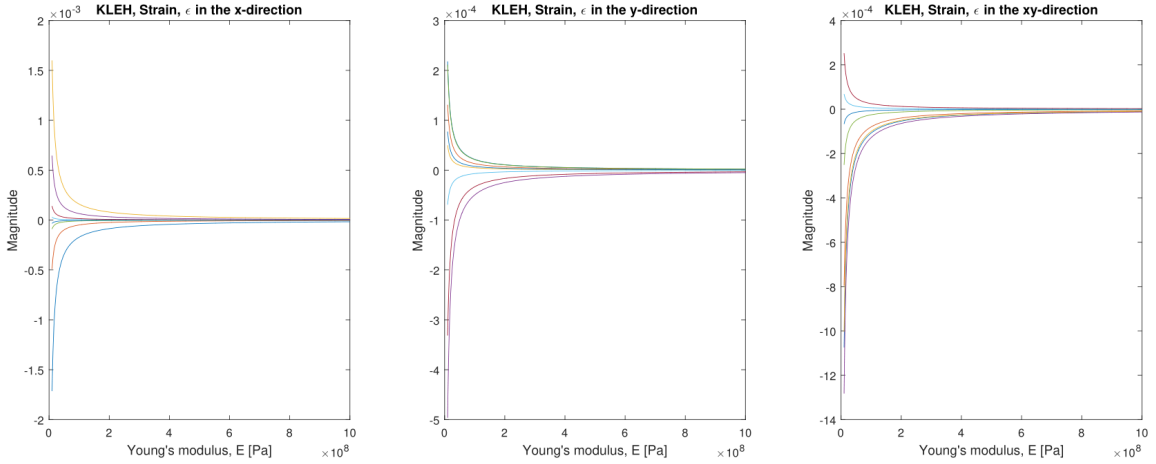
Figure 10.3: Displacement, u sensitivities when adjusting spring stiffness 2, k_2.



(a) KHEH

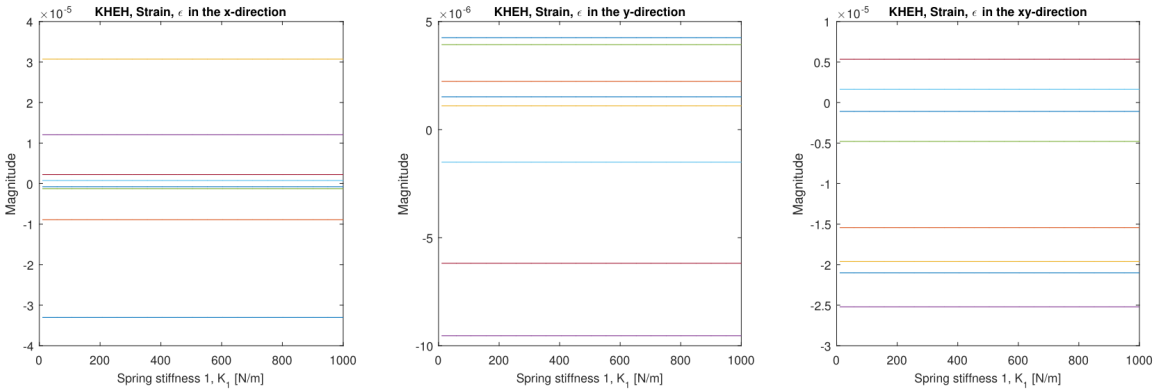


(b) KHEL

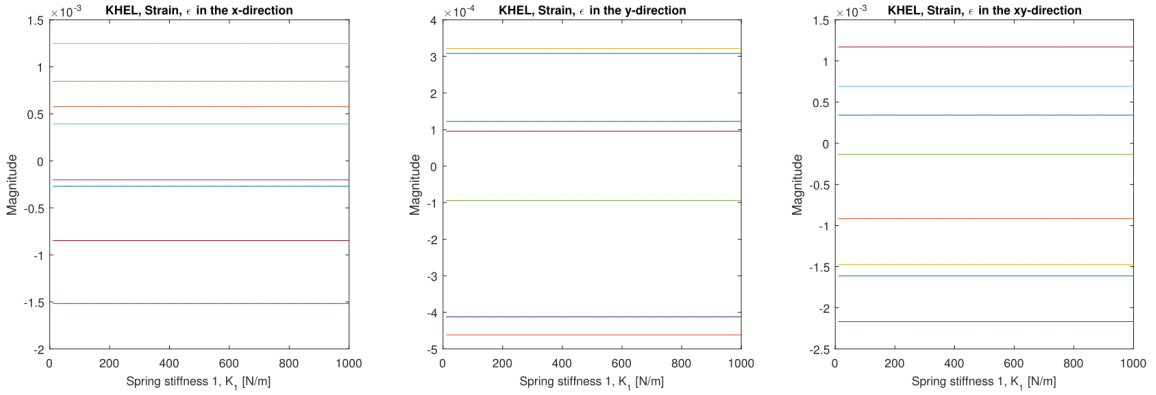


(c) KLEH

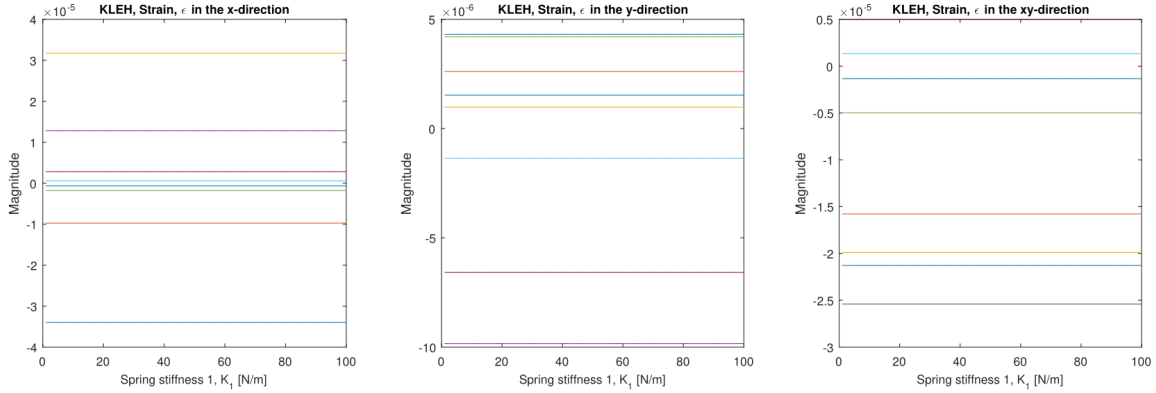
Figure 10.4: Strain, ϵ sensitivities when adjusting Young's modulus, E.



(a) KHEH

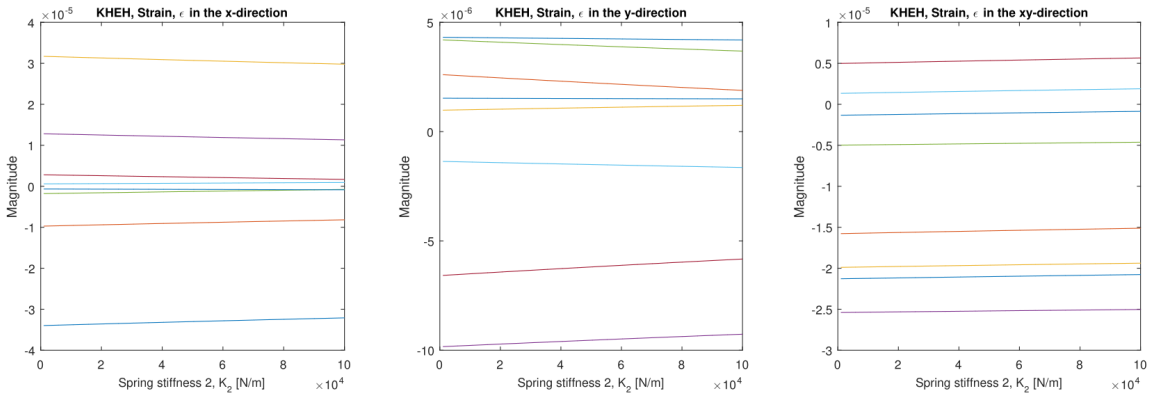


(b) KHEL

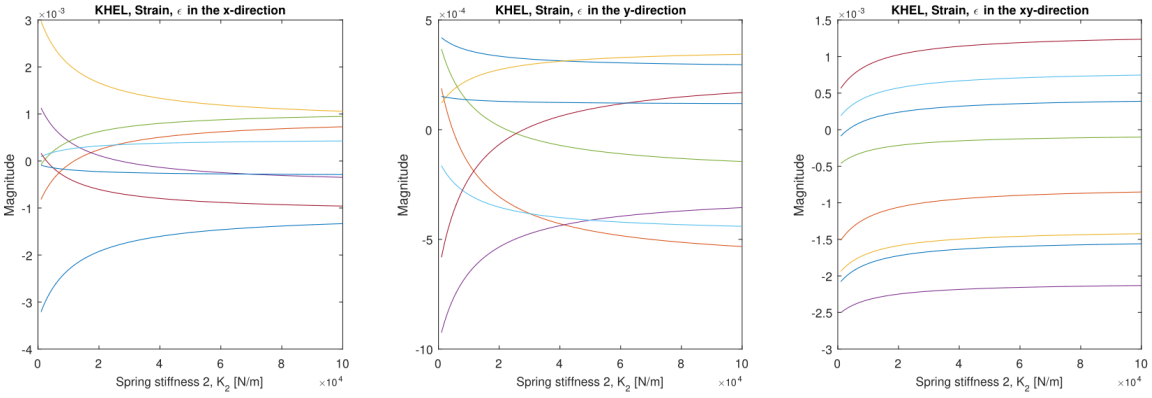


(c) KLEH

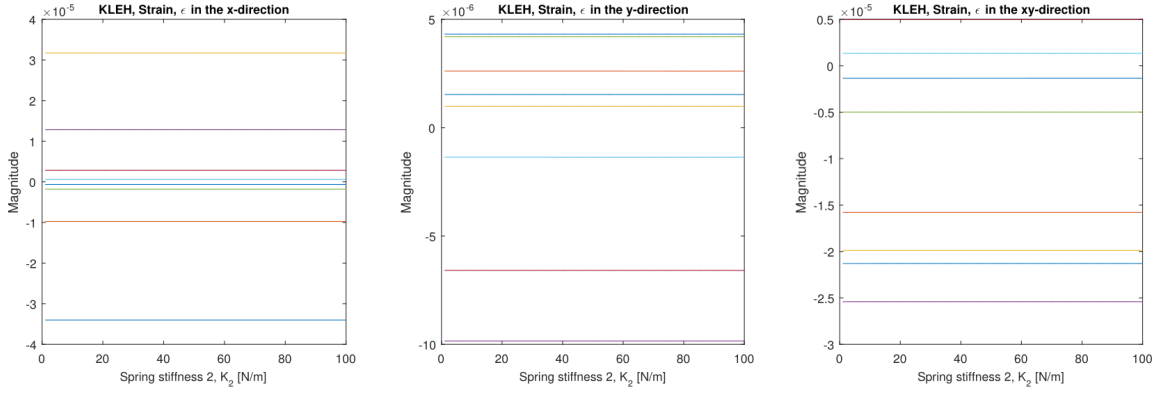
Figure 10.5: Strain, ϵ sensitivities when adjusting spring stiffness 1, k_1.



(a) KHEH



(b) KHEL



(c) KLEH

Figure 10.6: Strain, ϵ sensitivities when adjusting spring stiffness 2, k_2.

Bibliography

[1] Fraden, J. (2015). *Handbook of modern sensors: Physics, designs, and applications: 2016*. 4th edn. New York: Springer International Publishing AG.

[2] Song, J. and Jee, G. (2016). Performance enhancement of land vehicle positioning using multiple GPS receivers in an urban area. *Sensors*, 16(10), p. 1688.

[3] Gil, G. (2002). Enhancement in the sensitivity of a gas biosensor by using an advanced immobilization of a recombinant bioluminescent bacterium. *Biosensors and Bioelectronics*, 17(5), pp. 427-432.

[4] Fu, L., Dallas, P., Sharma, V.K. and Zhang, K. (2016). Sensors for environmental monitoring. *Journal of Sensors*, 2016, p. 1.

[5] Han, L., Voloshin, A. and Coulter, J. (1995). Application of the integrating fiber optic sensor for vibration monitoring. *Smart Materials and Structures*, 4(2), pp. 100-105.

[6] Chen, T. and Bowler, N. (2010). Analysis of a concentric coplanar capacitive sensor for nondestructive evaluation of multi-layered dielectric structures. *IEEE Transactions on Dielectrics and Electrical Insulation*, 17(4), pp. 1307-1318.

[7] Btefisch, S., Bttgenbach, S., Kleine-Besten, T. and Brand, U. (2001). Micromechanical three-axial tactile force sensor for micromaterial characterisation. *Microsystem Technologies*, 7(4), pp. 171-174.

[8] Imai, M. and Feng, M. (2012). Sensing optical fiber installation study for crack identification using a stimulated Brillouin-based strain sensor. *Structural Health Monitoring*, 11(5), pp. 501-509.

[9] Dhingra, N., Jovanovic, M. and Luo, Z. (2014). An ADMM algorithm for optimal sensor and actuator selection. *53rd IEEE Conference on Decision and Control*.

[10] Tryby, M., Propato, M. and Ranjithan, S. (2010). Monitoring Design for Source Identification in Water Distribution Systems. *Journal of Water Resources Planning and Management*, 136(6), pp.637-646.

[11] Yu, T.K. and Seinfeld, J.H. (1973). Observability and optimal measurement location in linear distributed parameter systems. *International Journal of Control*, 18(4), pp. 785-799.

[12] Sanayei, M., Onipede, O. and Babu, S.R. (1992). Selection of noisy measurement locations for error reduction in static parameter identification. *AIAA Journal*, 30(9), pp. 2299-2309.

[13] Frieden, R.B. (2004). *Science from Fisher information: A unification*. 2nd edn. Cambridge, England: Cambridge University Press.

[14] Sanayei, M., Javdekar, C.N. (2002). Sensor Placement for Parameter Estimation of Structures using Fisher Information Matrix. *Proceedings of the International Conference on Applications of Advanced Technologies in Transportation Engineering*, pp. 386-393.

[15] Fadale, T.D., Nenarokomov, A.V. and Emery, A.F. (1995). Two approaches to optimal sensor locations. *Journal of Heat Transfer*, 117(2), pp. 373-379.

[16] Kammer, D. (1991). Sensor placement for on-orbit modal identification and correlation of large space structures. *Journal of Guidance, Control, and Dynamics*, 14(2), pp.251-259.

[17] Ducoffe, M., Portelli, G. and Precioso, F. (2016). Scalable batch mode Optimal Experimental Design for Deep Networks. *29th Conference on Neural Information Processing Systems.*

[18] Mitchell, T.J. (2000). An algorithm for the construction of d-optimal experimental designs. *Technometrics*, 42(1), p. 48.

[19] Uciski, D. and Patan, M. (2007). D-optimal design of a monitoring network for parameter estimation of distributed systems. *Journal of Global Optimization*, 39(2), pp.291-322.

[20] Yanai, H., Takeuchi, K. and Takane, Y. (2011). *Projection matrices, generalized inverse matrices, and singular value decomposition.* New York: Springer-Verlag New York.

[21] Burden, R.L. and Faires, D.J. (2010). *Numerical analysis.* 9th edn. Boston, MA: Brooks/Cole, Cengage Learning.

[22] Flaherty, P., Arkin, A. and Jordan, M. I. (2005). Robust design of biological experiments. *In Advances in neural information processing systems*, pp. 363-370.

[23] Kammer, D.C. and Tinker, M.L. (2004). Optimal placement of triaxial accelerometers for modal vibration tests. *Mechanical Systems and Signal Processing*, 18(1), pp. 29-41.

[24] Stephan, C. (2012). Sensor placement for modal identification. *Mechanical Systems and Signal Processing*, 27, pp.461-470.

[25] Udwadia, F.E. (1994). Methodology for optimum sensor locations for parameter identification in dynamic systems. *Journal of Engineering Mechanics*, 120(2), pp. 368-390.

[26] Chisari, C., Macorini, L., Amadio, C. and Izzuddin, B. (2016). Optimal sensor placement for structural parameter identification. *Structural and Multidisciplinary Optimization*, 55(2), pp.647-662.

[27] Jazwinski, A. (2007). *Stochastic processes and filtering theory.* 1st ed. Mineola, N.Y.: Dover Publications.

[28] Li, Y., Xiang, Z., Zhou, M. and Cen, Z. (2007). An integrated parameter identification method combined with sensor placement design. *Communications in Numerical Methods in Engineering*, 24(12), pp. 1571-1585.

[29] Krause, A., Guestrin, C., Gupta, A. and Kleinberg, J. (2006). Near-optimal sensor placements. *Proceedings of the fifth international conference on Information processing in sensor networks - IPSN 06.*

[30] Wu, X., Liu, M. and Wu, Y. (2012). In-situ soil moisture sensing. *ACM Transactions on Sensor Networks*, 8(4), pp. 1-30.

[31] Du, W., Xing, Z., Li, M., He, B., Chua, L.H.C. and Miao, H. (2014). Optimal sensor placement and measurement of wind for water quality studies in urban reservoirs. *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks.*

[32] Du, W., Xing, Z., Li, M., He, B., Chua, L. and Miao, H. (2015). Sensor Placement and Measurement of Wind for Water Quality Studies in Urban Reservoirs. *ACM Transactions on Sensor Networks*, 11(3), pp.1-27.

[33] Kahn, H. and Marshall, A.W. (1953). Methods of reducing sample size in Monte Carlo computations. *Journal of the Operations Research Society of America*, 1(5), pp. 263-278.

[34] Iman, R.L. (2014). *Latin Hypercube sampling*. Wiley StatsRef: Statistics Reference Online.

[35] Abdi, H. (2003). Partial least square regression (PLS regression). *Encyclopedia for research methods for the social sciences*, pp.792-795.

[36] Korte, B. and Vygen, J. (2012). *Combinatorial optimization theory and algorithms*. 5th edn. Heidelberg: Springer-Verlag Berlin Heidelberg.

[37] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983). Optimization by simulated Annealing. *Science*, 220(4598), pp. 671-680.

[38] Arora, J.S. (2011). *Introduction to optimum design*. 3rd edn. Boston, MA: Elsevier Science.

[39] Goldberg, D.E. (1989). *Genetic algorithms in search, optimization, and machine learning*. 16th edn. Reading, MA: Addison-Wesley Educational Publishers.

[40] Worden, K. and Burrows, A.P. (2001). Optimal sensor placement for fault detection. *Engineering Structures*, 23(8), pp. 885-901.

[41] Mallardo, V., Aliabadi, M.H. and Khodaei, Z.S. (2013). Optimal sensor positioning for impact localization in smart composite panels. *Journal of Intelligent Material Systems and Structures*, 24(5), pp. 559-573.

[42] Joshi, S. and Boyd, S. (2009). Sensor Selection via Convex Optimization. *IEEE Transactions on Signal Processing*, 57(2), pp.451-462.

[43] Callister, W.D. (2010). *Materials science and engineering*. 8th edn. Chichester, United Kingdom: John Wiley & Sons.

[44] Broyden, C.G. (1967). Quasi-Newton methods and their application to function Minimisation. *Mathematics of Computation*, 21(99), p. 368.

[45] Powell, M.J.D. (1973). On search directions for minimization algorithms. *Mathematical Programming*, 4(1), pp. 193-201.

[46] Mathews, J.H. and Fink, K.D. (2003). *Numerical methods using MATLAB*. 4th edn. Harlow, United Kingdom: Prentice-Hall.

[47] de Jong, S. (1993). SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18(3), pp. 251-263.

[48] Edwards, A. (1976). *An introduction to linear regression and correlation*. 1st ed. San Francisco: W.H. Freeman and Co.

[49] Jolliffe, I.T. (2002). *Principal component analysis*. 2nd edn. New York: Springer-Verlag New York.

[50] Kiranyaz, S., Ince, T. and Gabbouj, M. (2013). *Multidimensional particle swarm optimization for machine learning and pattern recognition*. Berlin: Springer-Verlag Berlin and Heidelberg GmbH & Co. KG.

[51] Haupt, R.L. and Haupt, S.E. (2004). *Practical genetic algorithms with DC-Rom*. 2nd edn. New York: Wiley, John & Sons.

[52] Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3), pp.273-282.

[53] Hiriart-Urruty, J. and Le, H. (2011). Convexifying the set of matrices of bounded rank: applications to the quasiconvexification and convexification of the rank function. *Optimization Letters*, 6(5), pp.841-849.

[54] Chandrasekaran, V., Sanghavi, S., Parrilo, P. and Willsky, A. (2011). Rank-Sparsity Incoherence for Matrix Decomposition. *SIAM Journal on Optimization*, 21(2), pp.572-596.