GERMAN - MONGOLIAN INSTITUTE FOR RESOURCES AND TECHNOLOGY

The present work was submitted to the Faculty of Engineering

THE OPTIMAL SENSOR PLACEMENTS FOR VIBRATION-BASED DAMAGE DETECTION OF WIND TURBINE TOWER

Bachelor Thesis

by

Baigalmaa Purevdorj

Supervisor 1 / Examiner 1	**Ph.D. Sungchil Lee**
Supervisor 2 / Examiner 2	**Ph.D. Odbileg Norovrinchen**

Ulaanbaatar/Nalaikh, 05/16/2022

Abstract

Objective of this thesis is to find optimal sensor placements regarding vibration-based damage by utilizing modal parameters of a structure. The location of sensors is crucial for correct identification of the mode shapes of complicated mechanical structures. In this study, a specific model of a 100m wind tower was analyzed to determine the optimal sensor placement algorithm for vibration-based damage detection. The vibration-based damage detection was conducted to the system and damage was localized using a mode shape-based damage identification technique. The optimal sensor placements were obtained by minimization of weighted off-diagonal elements, QR decomposition, the genetic algorithm with maximum error and the genetic algorithm with weighted off-diagonal criteria. Among these three algorithms, the genetic algorithm with weighted off-diagonal criteria yielded the most effective sensor placements for the highest damage detection accuracy.

Table of Content

List of figures

List of tables

Abbreviations

SHM	Structural Health Monitoring
FEM	Finite Element Method
HAWT	Horizontal Axis Wind Turbine
OSP	Optimal Sensor Placement
DOFs	Degrees Of Freedom
MAC	Modal Assurance Criteria

Vectors and Matrices

M	mass matrix of a structure
K	stiffness matrix of a structure
Φ	rectangular mode shape matrix

Scalars

N	mass matrix of a structure
M	stiffness matrix of a structure

Operators

∑	summation symbol
| |	absolute value
$[\,]^T$	transpose of a matrix

Acknowledgement

I want to thank all the people who shared their knowledge with me for the development of this thesis. I would like to express my gratitude for my supervisor, Prof. Lee for his vision and valuable suggestions through his guidance. In many cases, he challenged me to clarify ideas and formulations and encouraged me to deliberate my approach in the full spectrum. I would like to express my thanks to Prof. Odbileg for everything he taught which built the fundamental knowledge for this thesis work. Furthermore, I am very grateful for the DAAD Sur Place Scholarships that the German Academic Exchange Service (DAAD) and GMIT have granted me for whole four years. Lastly, my family and my friends have constantly encouraged me during this project. Without these invaluable supports, I would not have completed the thesis on time.

Statutory Declaration

Purevdorj Baigalmaa	15011489819664
Last Name, First Name	Student ID Number

I hereby affirm in lieu of an oath that I provided the submitted bachelor thesis

THE OPTIMAL SENSOR PLACEMENTS FOR VIBRATION-BASED DAMAGE DETECTION OF WIND TURBINE TOWER

I did not use any sources other than those stated. In case that the work is additionally submitted on a data medium, I declare that the written and the electronic form are completely identical. The work was not submitted in the same or similar form to any examination authority.

Place, Date	Signature

1. Introduction

The research and development of wind energy systems are significantly increasing due to energy needs in the world. The main concern is to be able to ensure the safety and durability of the wind turbine towers. Structural health monitoring (SHM) can help bridges last longer, boost safety between planned inspections, and allow for a prioritized assessment following exceptional events. One of the most difficult tasks is determining whether problems may be diagnosed (i.e., detected or localized), which is necessary for determining the value of a SHM system before it is deployed and optimizing sensor location. Optimizing sensor placement has a critical effect on SHM. This thesis presents an optimal sensor solution using current research developments in the field of structural health monitoring.

1.1. Why Wind Energy

The energy is mostly supplied from fossil fuels, basically coal, oil, and gas. These sources in the end will run out. Additionally, the environment has been damaged by the intensive use of these fossil fuels resulting in global warming, extinction of species, and health problems. Wind energy and other renewable energy sources' impact on the environment is significantly lower compared to fossil fuels.

Mongolia is heavily dependent on coal energy and it is responsible for around 66% of the greenhouse gas emissions in Mongolia. This has created severe air pollution in the country accompanying health problems. 93% of total power generation is from coal-fired combined heat and power, which is providing more than 80% of the domestic demand. Aging CHP plants are facing challenges during peak hours in winter exceeding 90%. The energy supply shortage is an urgent challenge that the country is facing. Furthermore, additional mining energy demand is coming into place. Mongolia's wind energy potential could be realized at 1100GW. TsogtTsetsii 50MW, Sainshand Wind Farm 55MW, Salkhit wind farm 49.6MW are currently generating wind energy in Mongolia. Further wind farms could be invested if the country improves its grid system and regulation reserves.

1.2 Anatomy of wind turbines

Today's commercial wind turbines are primarily horizontal-axis designs with three blades, as shown in Figure 1. The primary subsystems of a horizontal axis wind turbine can be divided into five categories: rotor, nacelle, tower, foundation, and electrical system. The blades and hub are on the rotor, while the gearbox, drive train, control elements, and yaw system are on the nacelle. Depending on the type of turbine and whether it is onshore or offshore, the tower structure and foundation are developed.

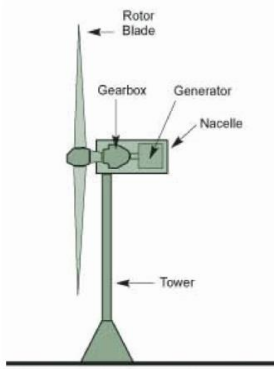

Figure 1: Horizontal wind turbine diagram


Figure 2: Tubular steel wind turbine tower

The purpose of the wind tower is to support the rotor and nacelle's mass. Wind turbine tower costs around 35% of the total investment. Minimizing the mass of the tower contributes to avoiding the big costs for the tower. Nowadays, wind turbine towers can be made of many different materials in many types. As for the material, towers have been made from concrete, structural steel, or a combination of these materials. Generally, towers are divided into lattice, cylindrical, concrete, and hybrid.

Steel cylindrical towers can be seen most commonly in the world. These types of towers are economical and also lighter than other types. The main reason to choose this structure is that it is much stronger. The tower's sections are connected with bolts. The diameter of the structure gets smaller from bottom to top. The thickness of the tower changes due to the loads. One of the important issues is transporting components, especially in Mongolia, which is a road problem. Steel cylindrical towers are most favored in Mongolia. Hence, this type of tower is selected for the test structure in the study.

1.3. Objectives of the Research

The objectives of this thesis are to study the vibration-based damage detection algorithm and optimization of sensor placements and evaluate the results of those. In this thesis, damage types of the wind turbine tower and sensor types used in vibration damage detection are described. Vibration based damage detection algorithms and optimization of sensor placements are studied. After that, mode shape-based damage detection method is used to detect damage on different damage scenarios. As for the sensor placement optimization, minimization off diagonal elements, QR decomposition and genetic algorithms are studied and used to find optimal sensor placements. The results of these optimization algorithms are compared.

2. Literature review

This section is divided into two parts; vibration-based damage detection and optimal sensor placement algorithm.

2.1 Vibration-based damage detection

The vibration-base damage detection's logic primarily relies on the fact that stiff, mass, and damping properties of the structure changes when the damage is present. Due to the health and safety of structures depend on monitoring the occurrence, creation, and dissemination of structural damage, structural monitoring has been a persistent focus of research. Several studies on vibration-based structural damage detection systems to identify and quantify structural damage have been published. Vibration-based damage detection is used to identify, locate, and quantify damage in the structure. The table shows the overview of the damage detection methods(1).

Features	Methodology	
Modal parameters	Natural frequency	• Frequency changes • Residual force optimization
	Mode shapes	• Mode shape changes • Modal strain energy • Mode shape derivatives
Matrix methods	Stiffness-based	• Optimization techniques • Model updating
	Flexibility-based	• Dynamically measured flexibility
Machine learning	Genetic algorithm	• Stiffness parameter optimization • minimization of the objective function
	Artificial neural network	• Backpropagation network training • time-delay neural network • Neural network systems identification with neural network damage detection

Table 1: Vibration-based damage detection methodology

Frequency-based approaches are based on the idea that resonant frequency changes occur when the structure's structural properties alter. Frequency information alone does not offer

reliable results, particularly when multiple different types of structural damage(2) can induce the same natural frequency variations. The limited sensitivity of frequency shifts to damage needs either very precise frequency variation readings or large amounts of damage for massive civil engineering structures.

To improve sensitivity to damage detection and localization, methods that use mode shapes generally compare acquired mode shapes directly, or the characteristics like curvature or modal strain energy. The expansion of acquired modal data to meet a finite element model was proposed by Law et al.(3), and the measured modal frequency changes were employed to evaluate damage severity using a sensitivity-based technique. Kim et al. (4) proposed the damage index approach to detect changes in modal strain energy to identify, locate, and quantify damage, where mode shapes were employed to identify the as-built and existing structures' modal stiffness qualities. The method was founded on the idea that during a minor damage event, the sensitivity of fractional modal strain energy of a possible damaged element is invariant.

The ability to find and assess the level of damage by comparing damaged and undamaged system matrices is inherent in the assembly and evaluation of system matrices from modal parameters. However, due to insufficient measurements or the inability to identify all modes of the structure, system matrices are only partially filled. Damage is identified, located, and quantified using stiffness reduction estimation.

Escobar et al. (5) introduced a transformation matrix approach for reducing a structure's global stiffness matrix to a condensed state with only the main degrees of freedom, which correspond to the structure's rigid body movements. With the eigensystem realization algorithm, Sivico et al. (6) created a damage detection algorithm that used a discrete state-space model of the time domain structural response to estimate stiffness and damping parameters.

Park and Reich (7) proposed a flexibility-based method based on the subdivided flexibility equation for detecting various sub-structural features by comparing observed global flexibilities before and after damage. A structure's global flexibility matrix was calculated using experimentally obtained modal characteristics. The link between measured global and local FRFs was used to create a sub-structural flexibility matrix. An increase in flexibility was discovered, localized, and quantified via a comparison of localized flexibilities of the undamaged and present state of a structure.

Machine learning is the creation of learning computing systems, often known as expert systems (8). Because of their effectiveness and robustness in dealing with uncertainty,

insufficient information, and noise, computational intelligence methods such as neural networks and genetic algorithms are appealing procedures in the area of structural damage identification (8). Inductive learning, or learning by example, is a common method for the application of machine learning algorithms to structural damage detection. For the characterization of damage in vibration-based damage detection of structures, artificial neural networks (ANN) and genetic algorithms (GA) are used.

With the primary components of the FRF as input in a neural network, Zang and Imregun (9) used principal component analysis, a linear data compression technique for dimensionality reduction. Hung and Kao (10) used experimental data to train a neural damage detection network (NDDN) to distinguish partial derivatives of accelerations, velocities, and displacements from a finite element update. It was presumed that some a priori knowledge of the system existed.

Static displacement measurements were used by Chou and Ghaboussi (11) for structural damage diagnosis, which was posed as an optimization problem with unknown stiffness factors. To determine the best-fit solution for the unknown stiffness values, GA was used. Hao and Xia (12) proposed using GA to compare changes in measured modal parameters from the undamaged and damaged structure to those of an analytical model, with the goal of estimating the stiffness reduction factor for each structural element while keeping the objective function as small as possible.

It's worth noting that a number of the approaches covered in this section illustrate the capacity to use numerical simulations to identify, locate, and assess the severity of damage in structures. However, due to signal noise, the lack of field validations reduces their effectiveness, limiting their potential for damage detection of FRP repaired bridge structures. Machine learning techniques keep receiving a lot of attention in the field of structural damage detection, with their ability to identify, locate, and estimate damage severity. However, the training of the network and the quantity of input information required for GA and ANN are still unresolved issues, and no methods have demonstrated robustness with field-measured data.

In this study, from above mentioned methods for vibration-based damage detection, mode-shape-based damage detection (MBDD) Method (13) is studied to localize the damage of the wind turbine tower

2.2 Optimization of sensor placement (OSP)

Wind turbines must have optimized dynamic behavior in order to operate reliably and economically over their expected lifetime. The modal dynamics of a wind turbine are thoroughly investigated using numerical simulation models for a wide range of operational situations in order to minimize component loads and achieve optimal overall system control behavior.

Extensive experimental modal analysis on a prototype under operational conditions is done to validate the realization of the design objectives for the dynamic behavior of a new wind turbine. The wind turbine's modal properties, such as eigenfrequencies, damping, and mode shapes, are computed using data from acceleration sensors dispersed throughout the structure. The optimal placement of a particular set of sensors on the structure in order to clearly identify the mode shapes in the important frequency range is a fundamental feature of experimental modal analysis.

The importance of sensor placement for experimental modal analysis is briefly described in Figure 1 using the mode shape one and two of the model structure. The goal is to use two acceleration sensors to detect the first and third mode shapes. Sensor position 2 is inconvenient since the two-mode shapes are difficult to distinguish. Sensor position 3 is also not optimal, with the third mode shape being imperceptible due to the sensor being placed on its node. Sensor locations 1 and 4 are sufficient to distinguish the two-mode shapes.

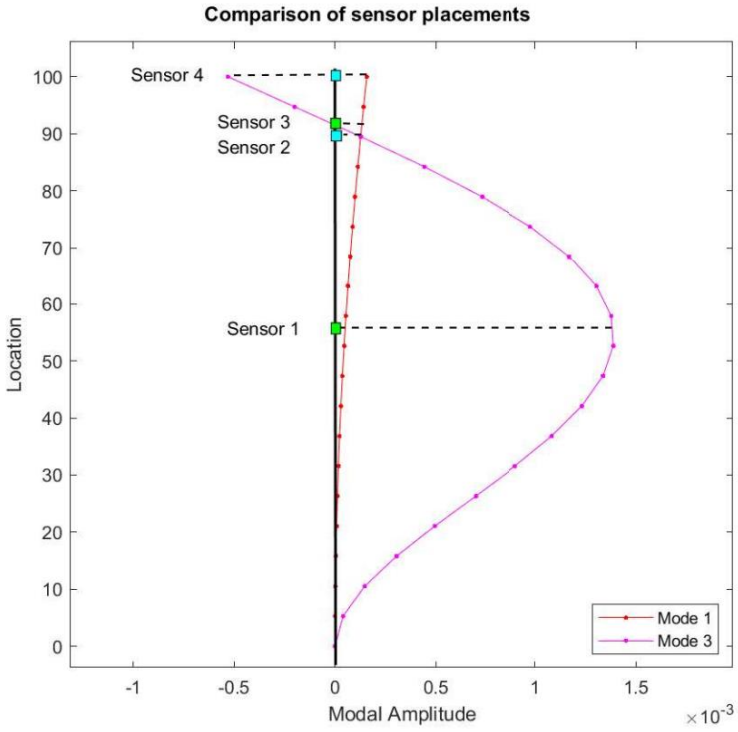

Figure 3: Basic sensor placements comparison

Using numerical data from a finite element model, optimal sensor placements on a complex mechanical structure are often obtained. Multibody models, on the other hand, are best for simulating the dynamic behavior of wind turbines under operational settings. Multibody models allow time-domain simulations of complex overall systems since they have a far smaller number of degrees of freedom than FE models. Wind turbine multibody models are built up of

elastic bodies defined by modally reduced FE models or lumped-mass models. In practice, only operational modal analysis can be used to do the experimental modal analysis of wind turbines.

The study compares three numerical methods for discovering appropriate sensor placements on the wind tower from a numerical modal analysis of the elastic multibody model for a 5MW wind turbine. This takes advantage of multibody model's ability to precisely describe the wind turbine's operational behavior.

3. Wind turbine tower modeling

3.1 Description of the test structure

An WTT structure was used in this study to assess the viability of the damage detection method and sensor placement optimization algorithms. The target structure was a 5 MW onshore WTT. The structure stands 100 meters tall and is made up of two segments. The both base and upper steel towers are 50 meters long. Table 2 shows how the thickness of the section changes as the height rises. At the lowest and top levels, the outer diameters were 4.572 m and 8.839 m, respectively (14). The WT head and tower weights are 480076 kg and 51816 kg.

Height [m]	0 ÷ 50	51 ÷ 100
Thickness [m]	0.0365 ÷ 0.0349	0.0349 ÷ 0.0222
Diameter [m]	8.839 ÷ 6.706	6.706 ÷ 4.572

Table 2: Thickness and diameter of the tower model(14)

Input variables that used in the FE modelling of tower is described in the table 3.

Input variable into FE model	Value
Height of the tower	$100\ m$
Hub mass	$480076\ kg$
Density of the material	$8700\ kg/m^3$
Elasticity of the steel	$2.1 * 10^{11} N/m^2$
Number of elements	50

Table 3: Input variables for FE model on MATLAB

3.2 Finite Element modeling

The numerical studies of WTT model carried out within MATLAB, in which the finite element method is applied. Tower model selected as it has two degrees of freedom (one translation

motion in the x direction and one rotational motion in the y axis) at each node. Dynamic response of the tower model uses modal parameters represented by finite-element models. Normal mode orthogonality and eigen value equation describes as follows (15)

$$\Phi^T M \Phi = I, \tag{1}$$

$$\mathrm{K}\Phi = \mathrm{M}\Phi\Omega^2. \tag{2}$$

The Φ is a normal modal matrix, natural frequency Ω^2 is a diagonal matrix. In the equations 3, 4 element stiffness and mass matrices (16) are described.

$$Ke = \frac{EI}{l^3}\begin{bmatrix} 12 & 6l & -12 & 6l \\ 6l & 4l^2 & -6l & 2l^2 \\ -12 & -6l & 12 & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{bmatrix}, \tag{3}$$

$$Me = \frac{\rho Al}{420}\begin{bmatrix} 156 & 22l & 54 & -13l \\ 22l & 4l^2 & 13l & -3l^2 \\ 54 & 13l & 156 & -22l \\ -13l & -3l^2 & -22l & 4l^2 \end{bmatrix} \tag{4}$$

The system is constructed using element matrices and lumped mass is included at the top of tower model using head mass of the WT. Eigenvalues and eigenvectors are computed by MATLAB's "eig()" function. The natural frequency Ω^2 is computed by

$$\Omega^2 = \frac{\sqrt{eigenvals([M]^{-1}[K])}}{2\pi}. \tag{5}$$

The unscaled (not mass normalized) mode shape matrix Ψ is generated as eigenvectors. The normalized mode shape characteristic indicated in the equation 6. If the mode shape does not show this characteristic, it is then mass normalized by equation 8 using factor α (17) calculated in the equation 7.

$$\{\Psi^T\}[M]\{\Psi\} = [I] \tag{6}$$

$$\alpha = \frac{1}{\sqrt{\{\Psi^T\}[M]\{\Psi\}}} \tag{7}$$

$$\{\Phi\} = \alpha \cdot \{\Psi\} \tag{8}$$

Figure 5 and 6 indicate first three mode shapes and cylindrical tower model structure. The mass of the hub is inserted in the system mass matrix as lumped mass system.

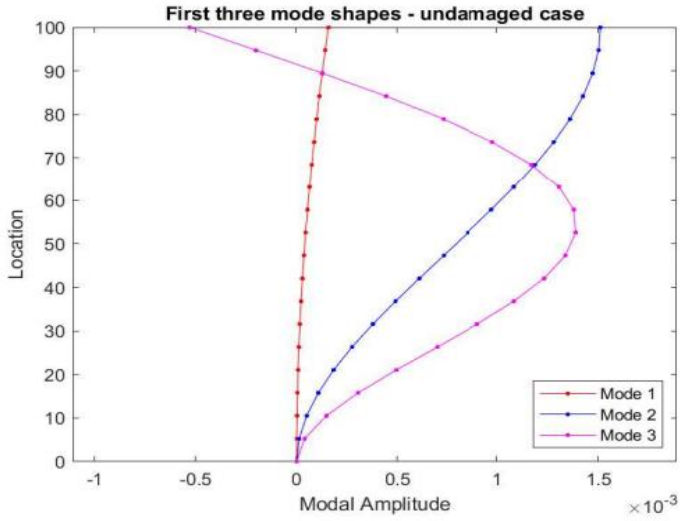

Figure 4: The first three mode shapes of the tower

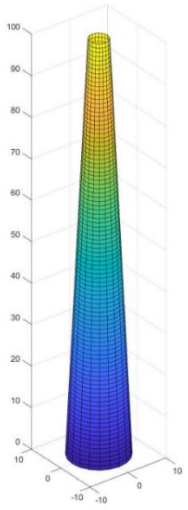

Figure 5: The FE tower model

4. Damage Identification and Optimization of Sensor Location

4.1. Damage types of wind turbine tower

Wind turbine towers are exposed to a range of loads such as wind, earthquakes, temperature, and weight. Because most wind turbine towers are vertical, horizontal loads have a significant impact. As a result, wind speed and direction are frequently considered important control elements. Many accidents and failures have shown how a single action can set off a chain of events that leads to incidents or failure. A list of lessons learned addressing the fields of structural design assessment, construction and quality control, and scientific document analyses are collected, with practical suggestions for risk assessment and actions. Tower collapse is most usually caused by severe winds, insufficient bolt strength, and poor bolt quality inspection during construction. Bolt failure and wind turbine tower buckling were the two most common failure types for wind turbine towers (18).

Wind turbine tower wall buckling was responsible for 52 percent of wind turbine tower collapses, while wind turbine tower bolt failure was responsible for just 36 percent. Wind turbine tower wall buckling has been the most common cause of wind turbine collapse around the world.

	Types of wind turbine disasters	Frequency of occurrence [%]
1	Blade failure	22
2	Fire	18
3	Environmental damage	16
4	Transport	14
5	Structural failure	11
6	Human injury	10
7	Fatal incidents	8
8	Ice throw	1

Table 4: Wind turbine disasters and its frequencies from 2007 to 2017 (19)

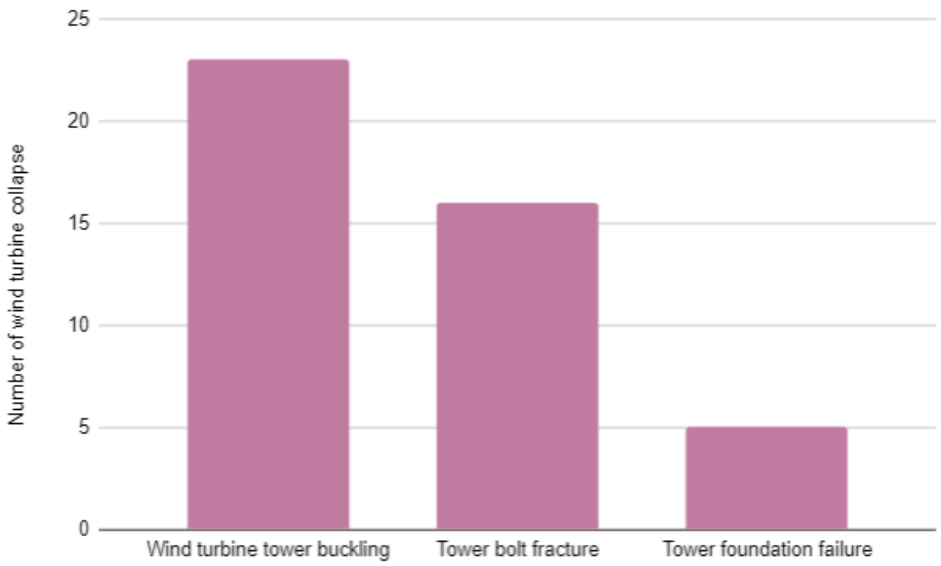

Figure 6: Statistics of wind turbine tower fracture

The 81% of collapses (19) occurred between the foundation and the third of the height of the wind turbine tower, which indicates that lower part of the tower is primary location of the damage. The wind turbine blades are parallel to the wind at wind speed of 50 m/s, wind turbine tower experiences great bending moment under excessive stress resulting in wall buckling, tower bolt fractures, and foundation overturn. Hence, pitch system failure causes a chain reaction that results in collapse of wind turbine tower.

4.2. Damage identification of the tower

The mode shape-based damage localization algorithm developed by Kim et al. is introduced to localize damage. The method uses a damage index approach to detect changes in modal strain energy to identify, locate, and quantify damage, where mode shapes were

employed to identify the as-built and existing structures' modal stiffness qualities. The method is based on the idea that during a minor damage event, the sensitivity of fractional modal strain energy of a possible damaged element is invariant. It is described by (20)

$$F_{ij} = F_{ij}^{*} \tag{9}$$

where F_{ij} is the fractional strain energy change contribution of the jth member to the ith mode, F_{ij}^{*} indicates fractional strain energy change for the damaged case.

In terms of modal shape approach, a finite element structural system with NE elements with NM nodes is considered. Damage localization indicator β_j for the jth member is calculated by (20)

$$\beta_j = \frac{\sum_{i=1}^{NM}(\Phi_i^{*T} C_{jo} \Phi_i^{*} + \sum_{k=1}^{NE} \Phi_i^{*T} C_{ko} \Phi_i^{*}) K_i}{\sum_{i=1}^{NM}(\Phi_i^{T} C_{jo} \Phi_i + \sum_{k=1}^{NE} \Phi_i^{T} C_{ko} \Phi_i) K_i^{*}} \tag{10}$$

The ith modal stiffness K_i, the ith modal stiffness for damaged case K_i^{*}, system total stiffness matrix (considering only geometrical properties) C_{jo} are calculated by equations 11, 12, and 13. The Φ_i is the ith modal vector, Φ_i^{*} is for the damaged structure and C is the total system stiffness matrix.

$$K_i = \Phi_i^{T} C \Phi_i \tag{11}$$

$$K_i^{*} = \Phi_i^{*T} C \Phi_i^{*} \tag{12}$$

In the equation 13, E_j is the element stiffness and C_j is element stiffness matrices.

$$C_j = E_j C_{jo} \tag{13}$$

Values of the indicator β_j are normalized according to the rule described in the equation 14, where $\bar{\beta}$ is the mean of the β_j population, and σ_β is the standard deviation of the population. Level of significance defined as 2 meaning if Z_j is greater than k, the jth member is damaged, if it is less, otherwise.

$$Z_j = \frac{\beta_j - \bar{\beta}}{\sigma_\beta} \tag{14}$$

Damage severity α_j is calculated by equation 15.

$$\alpha_j = \frac{1}{\beta_j} - 1 \tag{15}$$

4.3. Sensor types used for damage detection

New sensors have been developed as a result of the increased usage of SHM techniques to improve the efficiency of detection, location, and characterization systems (21). One of the most significant aspects of damage detection is sensor selection. Damage detection, as well as damage location, quantification, and classification, are all possible with the right sensor (22).

Piezoelectric materials are made of ceramics and polymers and exhibit both direct and inverse piezoelectric effects (23).This is why vibration-based sensors and actuators are frequently made from these materials. These piezoelectric sensors are also adaptable to the geometry of the structure in the area where they are installed. It is possible to measure vibration and collect information about several variables, such as deformation(24) or corrosion(25)(26), using these sensors.

Fiber optics are employed in applications requiring great precision and immunity to electromagnetic interference(27). Fiber optics is based on the notion of white-light interference(28). Any physical variable can be used to relate the absolute shifting of a signal emitted by a light source(29). Deformation, temperature, material concentrations, acceleration, rotation, pressure, vibrations, and shifting are all measured using this type of sensor. Fiber Bragg grating (FBG) and fiber optics sensors (FOS) are the most common deformation sensors. A single-axis FBG acceleration sensor is ideal for measuring minor vibrations at low frequencies. The sensor is appropriate for measuring structural responses to dynamic stimuli, such as earthquake-damaged structures, wind turbines under gust loading, and ship hulls under wave loading(30).

Deformation can be measured in three ways: (i) point sensors or discrete deformation, which can locate the deformation(31); (ii) quasi-distributed deformation sensors, which are a collection of point sensors(32); and (iii) distributed deformation sensors, which can determine a complete profile of deformations(33). Wire-based sensors are frequently put at different points on the tower wall, usually the platforms or the nacelle, with easy access to the supplied energy supply.

The goal of most acceleration recordings is to evaluate a vibration spectrum and a shift in eigenfrequencies, which indicates a change in structural stiffness, i.e., potential damage. The vibration spectrum, on the other hand, contains a global picture of a structure, whereas damages have only local impacts(34). The closer a sensor is nearby a point of damage, the more likely it is to detect the local effect.

4.4. Optimization of Sensor Placements

i. Auto Modal Assurance Criterion AutoMAC

AutoMAC is a measure that determines whether a given set of sensor positions is able to distinguish the modal shapes from each other. n0-dimensional mode shape vectors are obtained from a modal analysis of the multibody model of the wind turbine with n0 degrees of freedom (DOF). By measuring p DOF, where p is the number of sensors available, the goal is to identify m of these mode shapes, which are referred to as target mode shapes. Not all of the numerical model's n0 degrees of freedom can be employed as measurement points. For example, acceleration sensors can't measure rotational DOF, and the simulation model has DOF that can't be measured. The number of possible sensor positions is reduced to n by removing those degrees of freedom. The goal of this research is to locate p measurement degrees of freedom out of n degrees of freedom in order to identify m target mode forms. The AutoMAC compares a group of modes to each other and can be used to see if the position of a set of p measurement DOF is adequate to identify the modes(35).

$$\Phi = [\varphi_1 \quad \varphi_2 \quad \dots \quad \varphi_m] \tag{16}$$

$$MAC(ij) = \frac{\left(\varphi_i^T \varphi_j\right)^2}{\left(\varphi_i^T \varphi_i\right)\left(\varphi_j^T \varphi_j\right)}, \qquad i,j = 1,\dots,m \tag{17}$$

Since the mode shapes are associated with themselves in the event $i = j$, all diagonal members of the AutoMAC matrix are equivalent to one, however, in the case $i \neq j$, the off-diagonal elements have values from 0 to 1 depending on the linear dependency between the mode shape I and j. The optimal sensor placement is a solution that generates an AutoMAC matrix with the lowest off-diagonal elements.

ii. Minimizing weighted off-diagonal elements

Breitfeld (36) first introduced this optimization approach. A matrix ${}^{k}B = dia(1,\dots,1,0,1,\dots,1)$ with a zero at the k-th element is built up to explain the elimination of one measurement point with index k, starting with a sensor set that includes all reduced n DoF. The AutoMAC matrix proposed by Brietfiels is given by(35)

$${}^{k}MAC(i,j) = \frac{\left(\left({}^{k}B\varphi_i\right)^T\left({}^{k}B\varphi_j\right)\right)^2}{\left(\left({}^{k}B\varphi_i\right)^T\left({}^{k}B\varphi_i\right)\right)\left(\left({}^{k}B\varphi_j\right)^T\left({}^{k}B\varphi_j\right)\right)}, \qquad i,j = 1,\dots,m \tag{18}$$

The objective function is used to determine the impact of removing point k from the AutoMAC matrix(35).

$$Z(k) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} kMC(i,j)|i-j| \tag{19}$$

It is the weighted sum of the top AutoMAC elements that are off-diagonal. Due to the term $|i-j|$, the correlations between mode shapes that are far apart in frequency have less impact on Z if the mode forms in the mode shape matrix from (1) are ordered according to their eigenfrequencies. Mode forms with neighboring eigenfrequencies must be easily recognized, this weighting is critical for effective sensor placement.

The technique of eliminating one measurement point k with kB, assessing the AutoMAC matrix kMAC I j), and measuring the impact of the removal with Z(k), is repeated for each potential measurement point. Because its removal causes off-diagonal elements of the AutoMAC matrix to increase the most, the position k with the largest value of Z(k) is selected as a measurement point. This point will be removed from the selection procedure, and the process will repeat until the required amount of measurement points, p, is reached.

iii. QR decomposition

Link et al. devised the QR Decomposition (QRD) approach, which tries to find a subset of structural DOFs as measurement points to maximize the linear independence of the mode shapes to be measured. The underlying concept is that the modal matrix's most linearly independent rows imply the DOFs that should be selected as measuring points since they produce the smallest possible modal matrix, which results in a MAC matrix with the minimum off-diagonal terms, making it easier to differentiate between similar mode shapes. By QR decomposition of a transposed mode shape matrix in computation, the QRD method extracts those rows to produce the effective subset(37).

Schedlinski et al. (35) were the first to propose using the QR decomposition to determine ideal sensor placements. The QR decomposition minimizes an initial sensor set of all n potential sensor positions to an optimal set of the required p sensor positions, similar to the prior method. For the model matrix Φ QR decomposition is performed according to

$$\Phi^T P = QR, \tag{20}$$

where $P \in R^{n\times n}$ is a permutation matrix, $Q \in R^{m\times m}$ orthogonal matrix, and $R \in R^{m\times n}$ an upper triangular matrix. The element $R_{j+1,j+1}$ in R is a measure for the linear dependency between the j-th and $j+1$ -th columns of the transposed modal matrix T, as defined by the QR decomposition. The greater the $R_{j+1,j+1}$ value, the more linearly dependent those columns are. Because each column of T reflects information obtained from the associated potential

measurement point, the transposed modal matrix is examined in the equation. T columns that are linearly independent correspond to linearly independent measurement signals. The permutation matrix P is constructed using the components in R and arranges the diagonal elements in R in descending order. $\widetilde{\Phi}^T$'s columns are arranged by linear dependency, with the first column corresponding to the most linearly independent measurement point.

$$\widetilde{\Phi}^T = \Phi^T P \tag{21}$$

The first p columns are then used to pick the optimum sensor positions p of $\widetilde{\Phi}^T$. The number of m diagonal elements in R is thus limited to p. If p is greater than m, the required sensor positions p exceeds the target mode shape m, the first m columns of $\widetilde{\Phi}^T$ are picked and the associated Φ^{T} columns in are set to zero. The new Φ^{T} is then subjected to QR decomposition, and the remaining $p - m$ sensor positions are chosen from the new.

iv. Genetic algorithm

Genetic algorithms are a type of evolutionary algorithm that uses processes including inheritance, mutation, selection, and crossover that are inspired by evolutionary biology(38). A computer simulation is used to develop a customized genetic algorithm in which the genome improves toward better solutions. Each answer is referred to as an individual unique solution. In this approach, solutions are generated as binary strings of 0s and 1s. Evolution occurs in generations and begins with a population of randomly created individuals. Each generation assesses the fitness of each individual in the population. A predetermined number of individuals are randomly chosen from the present population depending on their fitness and maybe recombined (crossover) or even mutated to create a new population. The new population is subsequently employed in the algorithm's following iteration.

One potential set of sensor positions is expressed as an individual in a genetic algorithm. A binary vector is a simple approach to coding an individual so it provides a combination of possible sensor positions given with

$$s_p = [1 \quad 0 \quad 1 \quad 0 \quad 1 \quad \ldots \quad 0\,]. \tag{22}$$

The number of possible sensor locations n is equal to the length of s_p. A zero means that no sensor is set at this location, while a one indicates that a sensor is installed. As a result, the number of required sensors p is equal to the sum of the elements in sp. Anderson (39) proposes a different description. The length of the vector of potential sensor positions s_p is p, and it contains the position numbers of the sensors under evaluation. So, the s_p becomes

$$s_p = [1 \quad 3 \quad 5 \quad \cdots] \tag{23}$$

Upon the s_p coding of an individual, the corresponding fitness must be determined. Fitness determines the individual's survival to the following generation. The fitness index is defined in (40) by Stabb et al.

$$f = \frac{1}{1+E} \tag{24}$$

The index employs an error function E to indicate an individual's weakness. The definition of an individual's fitness is important to the genetic algorithm's effectiveness. As a reason, in this analysis, two different error functions E will be evaluated.

$$E_1 = max\left(\varphi_i^T \varphi_j - \varphi_{ind\,,i}^T \varphi_{ind\,,j}\right) \tag{25}$$

The first error function (E_1) calculates the maximum error between the correlation of φ_i, $\varphi_j$and $\varphi_{ind,i}$, $\varphi_{ind,j}$. The shapes $\varphi_{ind,i}$, $\varphi_{ind,j}$ are built up from the Φ with the sensor positioned elements.

$$E_2 = \sum_{i}^{m-1} \sum_{j=i+1}^{m} MAC_{ind}\,(i,j) e^{2-|i-j|} \tag{26}$$

The second error function (E_2) used for fitness index evaluates the weighted sum of the individual upper off-diagonal AutoMAC elements.

An initial population of stochastically selected individuals must be created before the genetic algorithm begins. The population is categorized in accordance with the equation 23, and each individual's fitness is assessed using equation 24. The natural selection criterion, which determines whether individuals move on to the next generation, must then be defined. According to Stabb et al (40), the 10 percent strongest individuals based on fitness should be immediately reproduced. Crossover reproduction, mutation, and direct reproduction will be used to generate the remaining 90% of the new generation. By combining two randomly selected individuals from the previous generation, crossover reproduction produces a new individual (child). A binary crossover vector is created at random for this combination. Crossover vector indicates where 1 indicates a child inherits a corresponding gene from 1st parent, otherwise from the second parent. Example with the p=10 sensors read;

$$parent_{\,1} = [75\ 77\ 79\ 81\ 99\ 101\ ], \tag{27}$$

$$parent_{\,2} = [11\ 15\ 69\ 81\ 91\ 101\ ], \tag{28}$$

$$crossover = [1\ 1\ 0\ 1\ 0\ 0\ ], \tag{29}$$

$$child = [75\ 77\ 69\ 81\ 91\ 101\ ] \tag{30}$$

The above crossover reproduction produces a majority of the individuals for the new generation. The other individuals are created through direct reproduction and mutation to maintain independence from the randomly created initial population. With direct reproduction, new individuals are developed randomly similar to the initial population. A child individual is created by modifying one random gene of a parent individual through mutation.

The preceding process is repeated until the termination criterion that determines the best sensor configuration is found. two termination criteria are used in this method. The first criterion is that the difference between maximum fitness and the average fitness of a generation should be smaller than a user defined $\varepsilon(41)$. The second criterion is to establish a large enough number c such that the algorithm ends if f_{max} has not changed in the through c iterations.

$$\left|f_{max} - f_{avg}\right| \leq \varepsilon, \tag{31}$$

User designed input data for the genetic algorithm is presented in the Table 5 and 6

Input data	Sensors []	Direct reproduction [%]	Natural Selection [%]	Crossover [%]	Mutation from crossover child [%]	Generation size []
Value	10	10	10	80	10	100

Table 5: User designed input data of generation of sensor placements with genetic algorithm

Input data	ε [ ]	c [ ]
Value	$1 * 10^{-5}$	150

Table 6: User defined data of two criteria of stopping iteration for genetic algorithm

4.5. Result and Discussion

i. Damage localization

Damage localization of the tower model is performed on MATLAB using mode-shape based vibration-based damage detection by Kim et.al(20).

The four cases of damage are induced in the system by decreasing stiffness of the certain elements. As shown in figure 11, damage algorithm successfully locates the damages in the

system, which are member 5 for case 1, member 36 for case 2 and 3, and member 5 and 43 for case 4.

	Case 1	Case 2	Case 3	Case 4
Member(s)	5	36	36	5, 43
Damage	-25%	-10%	-25%	-25%, -25%

Table 7: Damage scenarios of the model

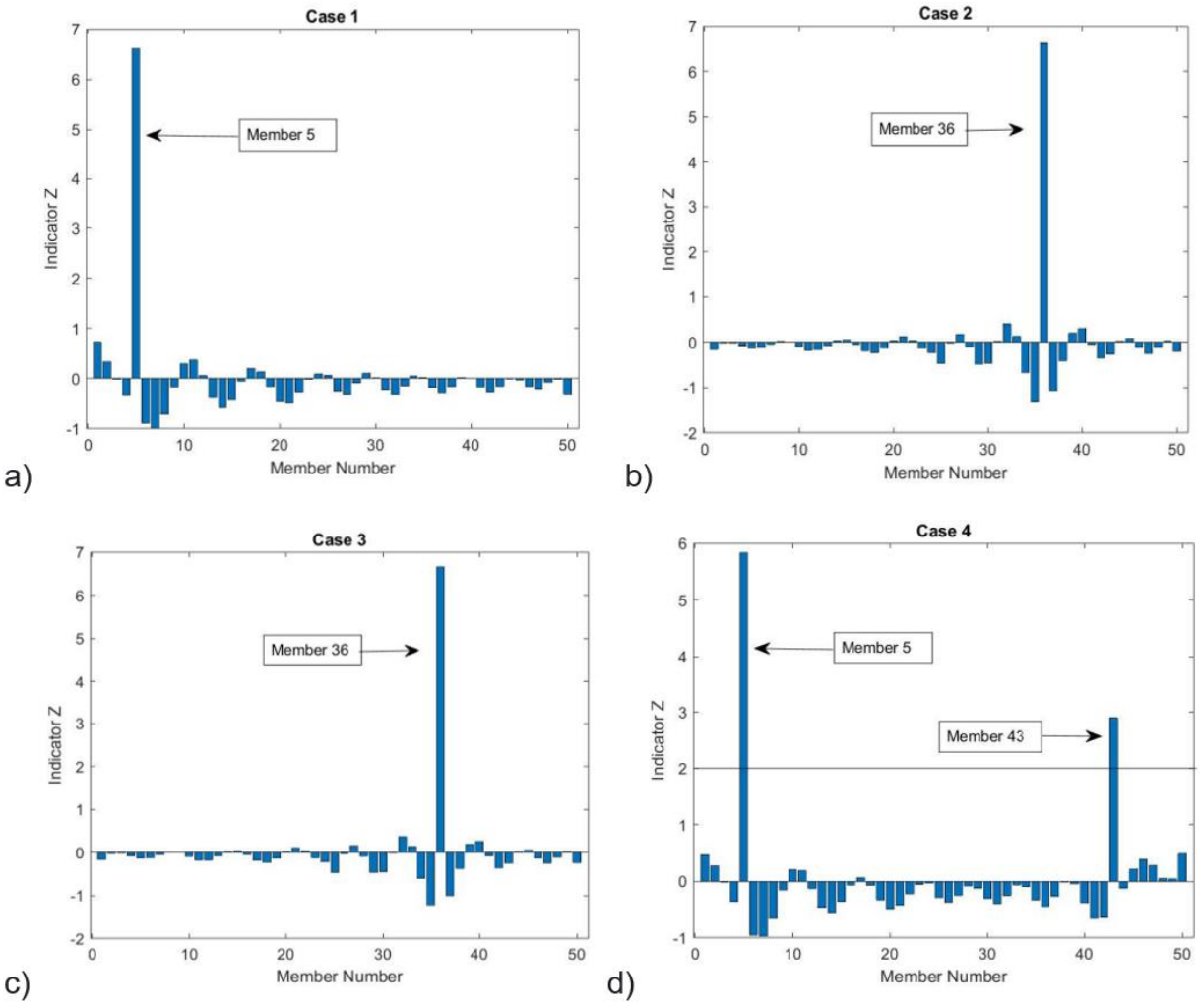

Figure 7: Damage indication for different damage case events; a) Case 1, b) Case 2, c) Case 3 and d) Case 4

The severity of the damaged is computed using equation 15. Damage severity of the localized damage by the algorithm shown in the table. Predicted severity estimation increase significantly when the number of target mode of the system is more. It is assumed that the target mode should be far less than number of degrees of freedom. For this table, number of target mode considered to be 10. The ratios of the predicted and simulated magnitudes of the

damages are relatively low and these indicates that the algorithm is underestimating the severity when the target mode shape is fewer.

Damage case	Damage location/ member	Simulated magnitude	Predicted magnitude	Ratio
1	5	-0.25	-0.10	0.4
2	36	-0.10	-0.04	0.4
3	36	-0.25	-0.13	0.52
4	5, 43	-0.25, 0.25	-0.10, -0.11	0.4, 0.44

Table 8: Severity estimation of simulated and predicted structure

ii. Sensor placement optimization

In order to find an optimal set of sensor placements modal data were applied to the sensor placement optimization scheme described above. Besides $\boldsymbol{\Phi}$, the only input data required for minimizing of the weighted off-diagonal scheme as well as for the QR decomposition is given number of sensors, here m=6 and m=10. The genetic algorithm on the other hand required additional input data listed in the Table 7 and 8. The result of the sensor placements algorithm with 6 and 10 sensors are illustrated in Figure 12. As genetic algorithm generates random population for every iteration, the 100 iterations are placed in the algorithm and mode of the population are then found out for every sensor placement. As shown in the Figure 12, for the minimization weighted off-diagonal element, QR decomposition the primary sensor placements are located mostly between the two third and top of the tower (Figure 12a), secondary most focused placements are located between bottom and one third of the tower (Figure 12b). However, genetic algorithm with maximum error criteria generates placements as mainly focused on between the two third and the top of the tower (Figure 12a, b). The placements generated by genetic algorithm with weighted off-diagonal criteria are mainly in the lower half of the tower.

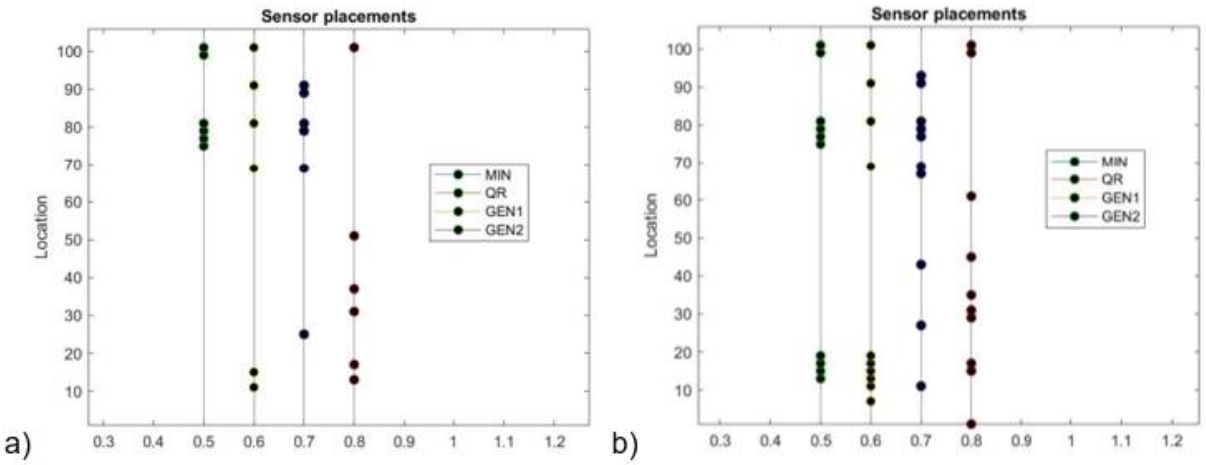

Figure 8: Sensor placement for MIN weighted off diagonal elements (MIN), QR decomposition (QR), Genetic algorithm with maximum error criteria (GEN1) and Genetic algorithm with weighted off diagonal criteria (GEN2) with a) 6 sensors and b) 10 sensors

Methods	**Case 1 (6 sensors)**	**Case 2 (10 sensors)**
Minimum weighted off diagonal elements	[75 77 79 81 99 101]	[13 15 17 19 75 77 79 81 99 101]
QR decomposition	[11 15 69 81 91 101]	[7 11 13 15 17 19 69 81 91 101]
Genetic algorithm with maximum error criteria	[25 69 79 81 89 91]	[11 27 43 67 69 77 79 81 91 93]
Genetic algorithm with weighted off diagonal criteria	[13 17 31 37 51 101]	[1 15 17 29 31 35 45 61 99 101]

Table 9: Sensor positions obtained by different methods with cases of 6 and 10 sensors

The AutoMAC matrices, presented in Figures 14-21, are used to evaluate the sensor sets that each method produces. The weaker the dependence between the sensor mode shapes, and thus the greater the measurement quality, the smaller the off-diagonal AutoMAC elements are. To assess the MAC matrixes of different methods, MAC value of 0.4 is compared to the matrix elements.

Case for 6 sensors

Figure 13 shows the AutoMAC matrix obtained by the minimizing off-diagonal elements scheme. Overall, the resulting AutoMAC matrix indicates certain low of linear dependency between some of the target mode shapes. The mode shape pairs 2-6 indicated with MAC value

of 0.4507. The mode shapes 1 and 3 could be clearly identified during a measurement with the proposed sensor configuration.

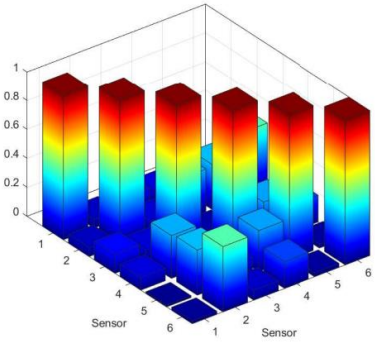

Figure 9: AutoMAC matrix generated by minimizing weighted off-diagonal elements

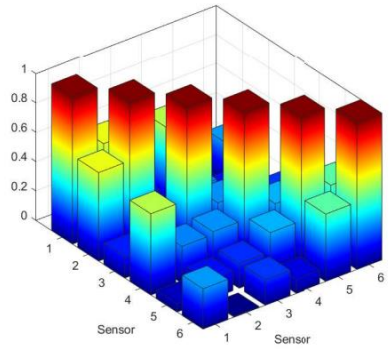

Figure 10: AutoMAC matrix generated by QR decomposition

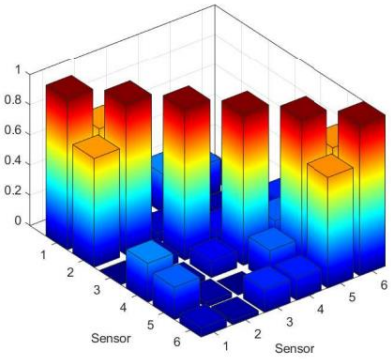

Figure 11: AutoMAC matrix generated by genetic algorithm with maximum error criteria

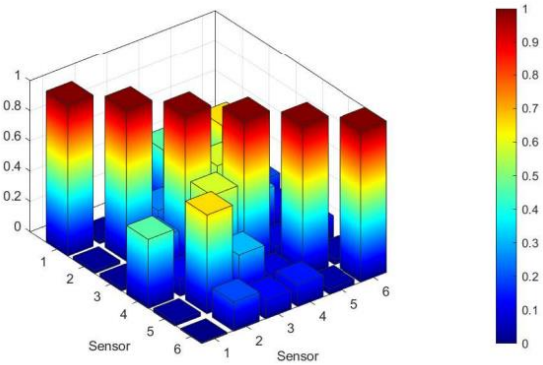

Figure 12: AutoMAC matrix generated by genetic algorithm with weighted off-diagonal criteria

The AutoMAC matrix obtained with the QR-decomposition is shown in Figure 14. Here the off-diagonal elements indicate a sensor configuration of lower quality than the one proposed by the first scheme. The Matrix indicates strong linear dependency between mode shape pairs 1-2, 1-4 and 5-6 with values over 0.4. Only mode shape 3 could be identified since their MAC values show weak correlation with the other mode shapes. The AutoMAC in Figure 15 is calculated based on the genetic algorithm with the maximum error criterion. The Matrix indicates strong linear dependency between mode shape pairs 1-2 and 5-6 with MAC values of 0.7210 and 0.7332. The mode shapes 3 and 4 could be clearly identified. The matrix in Figure 16 is obtained with the weighted off-diagonal criterion. The Matrix indicates strong linear dependency between mode shape pairs 1-4, 2-5 and 3-4 with MAC values over 0.4. Only the mode shape 6 could be identified.

Case for 10 sensors

In Figure 17, the resulting AutoMAC matrix indicates certain level of linear dependency between some of the target mode shapes. The mode shape pairs 4-5 with MAC value of 0.4124. In Figure 18, the resulting AutoMAC matrix indicates certain level of linear dependency

between some of the target mode shapes. The mode shape pairs 5-6 with MAC value of 0.4082. In Figure 19, the Matrix indicates certain linear dependency between mode shape pairs 1-2, 1-4, 1-8 and 2-4 with values over 0.5.

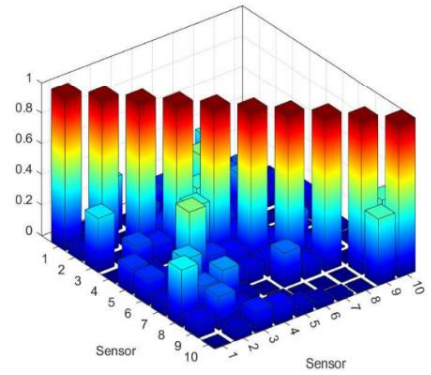

Figure 13: AutoMAC matrix generated by minimizing weighted off-diagonal elements

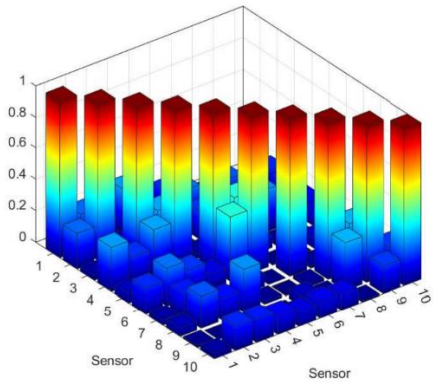

Figure 14: AutoMAC matrix generated by QR decomposition

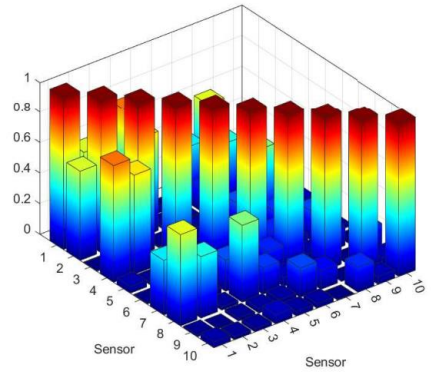

Figure 15: AutoMAC matrix generated by genetic algorithm with maximum error criteria

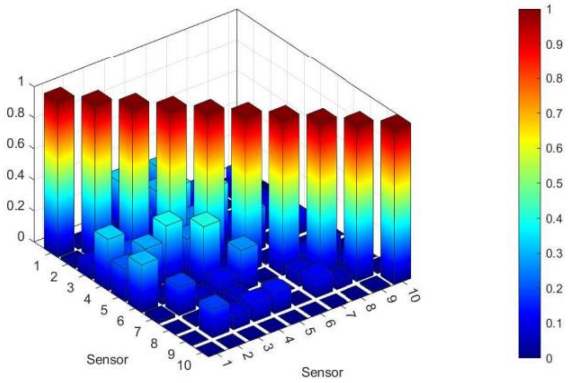

Figure 16: AutoMAC matrix generated by genetic algorithm with weighted off-diagonal criteria

The AutoMAC in Figure 20 is calculated based on the genetic algorithm with the weighted off diagonal criteria. The Matrix indicates relatively low linear dependency between mode shape pairs with the MAC values smaller than 0.4. Most of the mode shapes could be clearly identified compared to the other methods.

For the case of 6 sensor configuration, minimization of weighted off-diagonal element method generates smaller linear independency between modes compared to the other methods. However, for the case of 10 sensors, AutoMAC matrix obtained by genetic algorithm with weighted off-diagonal criteria has smaller off-diagonal elements, hence, it produces greater measurement quality.

5. Conclusion

In this thesis, damage types of the wind turbine tower and sensor types, which used in structural health monitoring for the case of vibration-based damage, are studied. The mode shape-based approach for damage detection is shown to be successful for localizing damage of the element with the reduced stiffness in the system. The estimated severity of the damage was underestimated due to the smaller number of target modes.

The most of the wind tower damages generally start at the lower part of the tower. The closer the sensor locations between two neighboring sensors near the simulated damage location, the more likely it is to detect the local effect. As the most of the tower collapses primarily starts at the lower one third part of the tower, the sensor location should be deployed on those locations first. From the sensor placements results, only genetic algorithm with weighted off-diagonal criteria generated the sensor placements, which were concentrated on the lower part with 6 sensors configuration performed better than the other methods. With the 10 sensors, all of the method located the sensors throughout the tower including damage susceptible lower part.

The AutoMAC matrices yielded different measurement quality for 6 and 10 sensor placements. In the case of 6 sensor configuration, sensor placements generated by minimization of off-diagonal element method produced lower linear dependency of mode shapes, as it indicates higher quality sensor method than others. However, when the number of sensors is 10, the genetic algorithm with the weighted off-diagonal criteria showed more accurate measurement results than other methods.

For those reasons, the genetic algorithm with weighted off-diagonal criteria is considered as a suitable method for optimal sensor placement of the tower model considering only one translation motion in the x direction for degree of freedom.

6. Future recommendation

The interconnection between vibration-based damage detection and the optimal sensor optimizations could be studied further. Moreover, other types of degrees of freedom might be considered in the future study.

References

1. Karbhari VM, Ansari Farhad. Structural health monitoring of civil infrastructure systems. 2009; 528.
2. Farrar CR, Sohn HY. Condition and Damage Monitoring Methodologies. In: The Consortium of Organizations for Strong Motion Observation Systems (COSMOS) Workshop [Internet]. Everyville, CA; 2001 [cited 2022 May 15]. Available from: https://www.osti.gov/biblio/975887
3. Law SS, Shi ZY, Zhang LM. Structural Damage Detection from Incomplete and Noisy Modal Test Data. Journal of Engineering Mechanics [Internet]. 1998 Nov 1 [cited 2022 May 15];124(11):1280–8. Available from: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-9399%281998%29124%3A11%281280%29
4. Stubbs N, Park S, Sikorsky C, Choi S. A global non-destructive damage assessment methodology for civil engineering structures. https://doi.org/101080/00207720050197758 [Internet]. 2010 [cited 2022 May 15];31(11):1361–73. Available from: https://www.tandfonline.com/doi/abs/10.1080/00207720050197758
5. Escobar JA, Sosa JJ, Gómez R. Damage detection in framed buildings. Canadian Journal of Civil Engineering. 2001;28(1):35–47.
6. Javier Valentin-Sivico L, Rao VS, Koval LR. Health monitoring of bridge like structures using state variable models. https://doi.org/101117/12274639 [Internet]. 1997 May 23 [cited 2022 May 15];3043(23):156–68. Available from: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/3043/0000/Health-monitoring-of-bridgelike-structures-using-state-variable-models/10.1117/12.274639.full
7. Park KC, Reich GW, Alvin KF. Structural Damage Detection Using Localized Flexibilities: http://dx.doi.org/101177/1045389X9800901107 [Internet]. 2016 Jul 27 [cited 2022 May 15];9(11):911–9. Available from: https://journals.sagepub.com/doi/abs/10.1177/1045389x9800901107
8. Machine Learning - Neural Networks, Genetic Algorithms and Fuzzy Systems. Kybernetes. 1999 Apr 1;28(3):317–8.

9. Zang C, Imregun M. Combined neural network and reduced FRF techniques for slight damage detection using measured response data. Archive of Applied Mechanics 2001 71:8 [Internet]. 2001 Aug [cited 2022 May 15];71(8):525–36. Available from: https://link.springer.com/article/10.1007/s004190100154

10. Hung SL, Kao CY. Structural damage detection using the optimal weights of the approximating artificial neural networks. Earthquake Engineering & Structural Dynamics [Internet]. 2002 Feb 1 [cited 2022 May 15];31(2):217–34. Available from: https://onlinelibrary.wiley.com/doi/full/10.1002/eqe.106

11. Chou JH, Ghaboussi J. Genetic algorithm in structural damage detection. Computers & Structures. 2001 Jun 1;79(14):1335–53.

12. Hao H, Xia Y. Vibration-based Damage Detection of Structures by Genetic Algorithm. Journal of Computing in Civil Engineering [Internet]. 2002 Jul 1 [cited 2022 May 15];16(3):222–9. Available from: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290887-3801%282002%2916%3A3%28222%29

13. Kim JT, Stubbs N. <title>Damage localization accuracy as a function of model uncertainty in the I-40 bridge over the Rio Grande</title>. In: Smart Structures and Materials 1995: Smart Systems for Bridges, Structures, and Highways. SPIE; 1995. p. 193–203.

14. Lanier MW. LWST Phase I Project Conceptual Design Study: Evaluation of Design and Construction Approaches for Economical Hybrid Steel/Concrete Wind Turbine Towers; June 28, 2002 -- July 31, 2004 [Internet]. 2005. Available from: http://www.osti.gov/bridge

15. Wei FS. Mass and stiffness interaction effects in analytical model modification. https://doi.org/102514/325269 [Internet]. 2012 May 17 [cited 2022 May 16];28(9):1686–8. Available from: https://arc.aiaa.org/doi/pdf/10.2514/3.25269

16. Kim JT, Stubbs N. <title>Damage localization accuracy as a function of model uncertainty in the I-40 bridge over the Rio Grande</title>. In: Smart Structures and Materials 1995: Smart Systems for Bridges, Structures, and Highways. SPIE; 1995. p. 193–203.

17. Aenlle ML, Brincker R, Fernández Canteli A. Some Methods to Determine Scaled Mode Shapes in Natural Input Modal Analysis.

18. Ma Y, Martinez-Vazquez P, Baniotopoulos C. Wind turbine tower collapse cases: a historical overview. https://doi.org/101680/jstbu1700167 [Internet]. 2019 Jul 22 [cited 2022 May 16];172(8):547–55. Available from: https://www.icevirtuallibrary.com/doi/abs/10.1680/jstbu.17.00167

19. Chou JS, Ou YC, Lin KY. Collapse mechanism and risk management of wind turbine tower in strong wind. Journal of Wind Engineering and Industrial Aerodynamics. 2019 Oct 1;193:103962.

20. Damage Detection In Offshore Jacket Structures From Limited Modal Information | International Journal of Offshore and Polar Engineering | OnePetro [Internet]. [cited 2022 May 16]. Available from: https://onepetro.org/IJOPE/article-abstract/26815/Damage-Detection-In-Offshore-Jacket-Structures

21. Das S, Saha P. A review of some advanced sensors used for health diagnosis of civil engineering structures. Measurement. 2018 Dec 1;129:68–90.

22. Li H, Ou J. Structural Health Monitoring: From Sensing Technology Stepping to Health Diagnosis. Procedia Engineering. 2011 Jan 1;14:753–60.

23. Barski M, Kedziora P, Muc A, Romanowicz P. Structural health monitoring (SHM) methods in machine design and operation. Archive of Mechanical Engineering. 2014 Dec 1;Vol. LXI(nr 4):653–77.

24. de Castro BA, Baptista FG, Ciampa F. New signal processing approach for structural health monitoring in noisy environments based on impedance measurements. Measurement. 2019 Apr 1;137:155–67.

25. Chandarana N, Lansiaux H, Gresil M. Characterisation of Damaged Tubular Composites by Acoustic Emission, Thermal Diffusivity Mapping and TSR-RGB Projection Technique. Applied Composite Materials [Internet]. 2017 Apr 1 [cited 2022 May 16];24(2):525–51. Available from: https://link.springer.com/article/10.1007/s10443-016-9538-8

26. Yeasin M, Roshan Joseph V, Yeasin Bhuiyan M, Giurgiutiu V, Joseph R. Acoustic emission source modeling in a plate using buried moment tensors. https://doi.org/101117/122260167 [Internet]. 2017 Apr 28 [cited 2022 May 16];10170:464–75. Available from: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10170/1017028/Acoustic-emission-source-modeling-in-a-plate-using-buried-moment/10.1117/12.2260167.full

27. In-Fiber Integrated Sensor Array With Embedded Weakly Reflective Joint Surface [Internet]. [cited 2022 May 16]. Available from: https://opg.optica.org/jlt/abstract.cfm?uri=jlt-36-23-5663

28. A Novel White Light Interference Based AFM Head [Internet]. [cited 2022 May 16]. Available from: https://opg.optica.org/jlt/abstract.cfm?uri=jlt-35-16-3604

29. de Araújo M, Robert Schmitt E, König N, Manfrin de Araújo E, Schmitt R. Surface profile analysis using a fiber optic low-coherence interferometer. https://doi.org/101117/12827823 [Internet]. 2009 Jun 17 [cited 2022 May 16];7389(17):372–9. Available from: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/7389/738914/Surface-profile-analysis-using-a-fiber-optic-low-coherence-interferometer/10.1117/12.827823.full

30. FBG Accelerometer | Safibra, s.r.o. [Internet]. [cited 2022 May 16]. Available from: http://www.safibra.cz/en/FBG-Accelerometer

31. Das S, Saha P. A review of some advanced sensors used for health diagnosis of civil engineering structures. Measurement. 2018 Dec 1;129:68–90.

32. Kinet D, Mégret P, Goossen KW, Qiu L, Heider D, Caucheteur C. Fiber Bragg Grating Sensors toward Structural Health Monitoring in Composite Materials: Challenges and Solutions. Sensors 2014, Vol 14, Pages 7394-7419 [Internet]. 2014 Apr 23 [cited 2022 May 16];14(4):7394–419. Available from: https://www.mdpi.com/1424-8220/14/4/7394/htm

33. Drissi-Habti M, Raman V, Khadour A, Timorian S. Fiber Optic Sensor Embedment Study for Multi-Parameter Strain Sensing. Sensors 2017, Vol 17, Page 667 [Internet]. 2017 Mar 23 [cited 2022 May 16];17(4):667. Available from: https://www.mdpi.com/1424-8220/17/4/667/htm

34. Wondra B, Malek S, Botz M, Glaser SD, Grosse CU. Wireless High-Resolution Acceleration Measurements for Structural Health Monitoring of Wind Turbine Towers. Data-Enabled Discovery and Applications 2019 3:1 [Internet]. 2019 Jan 14 [cited 2022 May 16];3(1):1–16. Available from: https://link.springer.com/article/10.1007/s41688-018-0029-y

35. Schulze A, Zierath J, Rosenow SE, Bockhahn R, Rachholz R, Woernle C. Optimal sensor placement for modal testing on wind turbines. In: Journal of Physics: Conference Series. Institute of Physics Publishing; 2016.

36. A Method for Identification of a Set of Optimal Measurement Points for Experimental Modal Analysis - NASA/ADS [Internet]. [cited 2022 May 16]. Available from: https://ui.adsabs.harvard.edu/abs/1995SPIE.2460.1029B/abstract

37. Li D. Herausgeber: Claus-Peter Fritzen Sensor Placement Methods and Evaluation Criteria in Structural Health Monitoring.

38. Coley DA. An Introduction to Genetic Algorithms for Scientists and Engineers. An Introduction to Genetic Algorithms for Scientists and Engineers. 1999 Jan;

39. 30682235-MIT.

40. A Genetic Algorithm for Optimally Selecting Accelerometer Locations - NASA/ADS [Internet]. [cited 2022 May 16]. Available from: https://ui.adsabs.harvard.edu/abs/1995SPIE.2460.1530S/abstract

41. Guo HY, Zhang L, Zhang LL, Zhou JX. Optimal placement of sensors for structural health monitoring using improved genetic algorithms. Smart Materials and Structures [Internet]. 2004 Apr 19 [cited 2022 May 16];13(3):528. Available from: https://iopscience.iop.org/article/10.1088/0964-1726/13/3/011

Appendix

Tables 10-17 shows AutoMAC values of different sensor placement algorithms. The values greater than 0.4 are highlighted in yellow.

1.0000	0.0607	0.1288	0.0863	0.0122	0.0211
	1.0000	0.0806	0.2950	0.3107	0.4507
		1.0000	0.0369	0.1150	0.0673
			1.0000	0.2826	0.1845
				1.0000	0.0266
sym					1.0000

Table 10: AutoMAC matrix generated by minimizing weighted off-diagonal elements for 6 sensors

1.0000	0.6027	0.1460	0.5538	0.0555	0.2743
	1.0000	0.0064	0.2501	0.1342	0.0076
		1.0000	0.2688	0.1674	0.1739
			1.0000	0.2974	0.0933
				1.0000	0.4584
sym					1.0000

Table 11: AutoMAC matrix generated by QR decomposition for 6 sensors

1.0000	0.7210	0.0029	0.2569	0.2142	0.0603
	1.0000	0.0651	0.0279	0.0324	0.0289
		1.0000	0.1058	0.0009	0.1538
			1.0000	0.2143	0.1448
				1.0000	0.7332
sym					1.0000

Table 12: AutoMAC matrix generated by genetic algorithm with maximum error criteria for 6 sensors

1.0000	0.0210	0.0234	0.4572	0.0252	0.0075
	1.0000	0.2537	0.1114	0.6535	0.1933
		1.0000	0.5847	0.3068	0.1255
			1.0000	0.0847	0.1439
				1.0000	0.0358
sym					1.0000

Table 13: AutoMAC matrix generated by genetic algorithm with weighted off-diagonal criteria for 6 sensors

1.0000	0.0579	0.3525	0.0006	0.1689	0.1675	0.0792	0.3721	0.1024	0.0000
	1.0000	0.0009	0.1896	0.0718	0.1170	0.3173	0.2100	0.2184	0.0637
		1.0000	0.1454	0.0053	0.5036	0.1212	0.2657	0.0891	0.1426
			1.0000	0.4124	0.1619	0.1280	0.0425	0.0002	0.1226
				1.0000	0.1262	0.0921	0.0049	0.0190	0.0685
					1.0000	0.1089	0.2244	0.0044	0.0428
						1.0000	0.0973	0.0003	0.0168
							1.0000	0.0037	0.0043
								1.0000	0.4343
sym									1.0000

Table 14: AutoMAC matrix generated by minimizing weighted off-diagonal elements for 10 sensors

1.0000	0.2252	0.0302	0.2787	0.0590	0.1630	0.0316	0.0511	0.0133	0.0345
	1.0000	0.0452	0.1682	0.0027	0.2371	0.1460	0.2033	0.0004	0.1575
		1.0000	0.2874	0.0011	0.1756	0.0151	0.1345	0.0071	0.1469
			1.0000	0.0301	0.0943	0.0016	0.2692	0.0155	0.1199
				1.0000	0.4082	0.0806	0.0174	0.0007	0.0914
					1.0000	0.0002	0.0036	0.0621	0.1042
						1.0000	0.0186	0.0169	0.0947
							1.0000	0.3025	0.0101
								1.0000	0.1762
sym									1.0000

Table 15: AutoMAC matrix generated by QR decomposition for 10 sensors

1.0000	0.5825	0.0035	0.7616	0.0430	0.0057	0.3567	0.5958	0.0133	0.0380
	1.0000	0.2507	0.6479	0.0157	0.0089	0.1483	0.4005	0.0012	0.0214
		1.0000	0.0069	0.0212	0.0237	0.0354	0.0204	0.0183	0.0226
			1.0000	0.0614	0.0006	0.2024	0.5015	0.0289	0.0615
				1.0000	0.0277	0.0786	0.1724	0.0350	0.0310
					1.0000	0.1562	0.0567	0.1906	0.0296
						1.0000	0.0354	0.1215	0.0030
							1.0000	0.0011	0.1633
								1.0000	0.0500

sym									1.0000

Table 16: AutoMAC matrix generated by genetic algorithm with maximum error criteria for 10 sensors

1.0000	0.0180	0.0902	0.3282	0.2098	0.3073	0.0188	0.0004	0.0009	0.0000
	1.0000	0.0228	0.1827	0.2870	0.0356	0.1820	0.0138	0.1833	0.0000
		1.0000	0.0165	0.3843	0.0317	0.0566	0.0068	0.0834	0.0000
			1.0000	0.2461	0.3984	0.1113	0.0689	0.1185	0.0000
				1.0000	0.0462	0.2650	0.0037	0.1334	0.0000
					1.0000	0.0065	0.0005	0.0016	0.0000
						1.0000	0.0792	0.1234	0.0000
							1.0000	0.0500	0.0000
								1.0000	0.0113
sym									1.0000

Table 17: AutoMAC matrix generated by genetic algorithm with weighted off-diagonal criteria for 10 sensors

MATLAB code

```
%% Undamaged case computing
% damage detection algorithm
% modal stiffness
clc
clear
nelm=50;
ndof= 2*nelm+2;
M(ndof,ndof)=0.0;
C(ndof,ndof)=0.0;
roh=8700;%kg/m3              %structural design optimiztion of wind
towers
E=2.1E11; %(N/m^2)
l=100/nelm;
secoftower=2;
sss=100/secoftower;
% A=0.076058; %m^2
% I=0.0557243; %(m^4)
d_D1=(8.839-6.706)/sss;
t_D1=(0.0365-0.0349)/sss;
d_D2=(6.706-4.572)/sss;
t_D2=(0.0349-0.0222)/sss;
D1=8.839; t1=0.0365;
D2=6.706; t2=0.0349;
% D3=4.572; t3=0.0222;
for i=1:(nelm/2)
 A(i)=pi*(((D1-i*d_D1)/2)^2-((D1-i*d_D1)/2-(t1-i*t_D1))^2);
 I(i)=pi/64*((D1-i*d_D1)^4-((D1-i*d_D1)-2*(t1-i*t_D1))^4);
Ce(:,:,i)=(E*I(i)/l^3)*[12   6*l     (-12)     6*l;
          6*l   4*l^2   (-6*l)   (2*l^2);
          (-12) (-6*l)  12        (-6*l);
          6*l   (2*l^2) (-6*l)  (4*l^2)];
Me(:,:,i)= (roh*A(i)*l/420)*[156   22*l      54   -13*l;
               22*l   4*l^2     13*l   -3*l^2;
               54     13*l      156     -22*l;
              -13*l  -3*l^2    -22*l   4*l^2];
end
for i=(nelm/2):nelm
 A(i)=pi*(((D2-i*d_D2)/2)^2-((D2-i*d_D2)/2-(t2-i*t_D2))^2);
 I(i)=pi/64*((D2-i*d_D2)^4-((D2-i*d_D2)-2*(t2-i*t_D2))^4);
Ce(:,:,i)=(E*I(i)/l^3)*[12   6*l     (-12)     6*l;
          6*l   4*l^2   (-6*l)   (2*l^2);
          (-12) (-6*l)  12        (-6*l);
          6*l   (2*l^2) (-6*l)  (4*l^2)];
Me(:,:,i)= (roh*A(i)*l/420)*[156   22*l      54   -13*l;
               22*l   4*l^2     13*l   -3*l^2;
               54     13*l      156     -22*l;
              -13*l  -3*l^2    -22*l   4*l^2];
End
```

```
%%
p=0;
q=0;
for ne=1:nelm
for ii=1:4
    for jj=1:4
        M(p+ii,q+jj)=M(p+ii,q+jj)+ Me(ii,jj,ne);
        C(p+ii,q+jj)=C(p+ii,q+jj)+ Ce(ii,jj,ne);
    end
end
p=p+2;
q=q+2;
end
C(1,1)=1E15;
C(2,2)=1E15;
%
M(ndof-1,ndof-1)=M(ndof-1,ndof-1)+480076;
M(ndof,ndof)=M(ndof,ndof)+480076;
%*******************************
[evec,eval]=eig(C,M);
fhz(ndof,ndof)=0.0;
for i=1:ndof
  eval(i,i)=sqrt(eval(i,i));
  fhz(i,i)= eval(i,i)/(2*pi);
end
%% Consider only Vertical DOF
Pho = zeros(ndof);
Pho = Pho + evec;
%Checking mass normalised mode shape
a=zeros(1, ndof);
for i=1:ndof
a(i) = Pho(:,i).' * M * Pho(:,i);
if a(i)==1.00
else
   Pho(:,i) = 1/sqrt(a(i)) * Pho(:,i);
end
end
%i th modal stiffness K_i  %K_i=Pho_i'T * C * Pho_i
Ki=zeros(1,ndof);
for i=1:ndof
Ki(i)= Pho(:,i).' * C * Pho(:,i);
end
%contribution of the j th member to the i th modal stiffness K_ij
%K_ij=Pho_i'T * Cj * Pho_i
% making Cj matrix
Cj=zeros(ndof,ndof,nelm);
for i=1:nelm
b=2*i-1;
Cj(b:(b+3),b:(b+3),i)=Ce(:,:,i);
end
%
Kij=zeros(1,1,nelm);
for i= 1:ndof
for j=1:nelm
```

```
    Kij(:,i,j)= Pho(:,i).' * Cj(:,:,j) * Pho(:,i);
end
end
%F_ij=K_ij/K_i
% fraction of modal energy for i th mode that is concentrated in the
% j th member(the sensitivity of the j th member to the i th mode)
Fij(1,ndof,nelm)=0.0;
for i=1:ndof
for j=1:nelm
Fij(:,i,j)=Kij(:,i,j)/Ki(:,i);
end
end
%% plotting mode shapes
plot_Pho=zeros(20,3);
for j=1:3
  jko=2*j-1;
for i=1:20
  ok=2*i-1;
  plot_Pho(i,j)=Pho(ok,jko);
end
end
figure(1)
x=linspace(0,100,20);
plot(plot_Pho(:,1),x,'-r.')
hold on
plot(plot_Pho(:,2),x,'-b.')
hold on
plot(plot_Pho(:,3),x,'-m.')
title('First three mode shapes - undamaged case')
legend({'Mode 1','Mode 2','Mode 3'},'Location','southeast')
xlabel('Modal Amplitude')
ylabel('Location')
hold off
```

```
%% Damaged case computing
%Condition:
dmgmem=[5; 0.25];
dmn(1:2, 1:nelm)=0.0;
```

```
for i=1:nelm
  for o=1:length(dmgmem(1,:))
      if i == dmgmem(1,o)
        dmn(1,i)=1;
        dmn(2,i)=dmgmem(2,o);
      end
  end
end
M_d = M;  % damaged mass matrix, same
C_d(ndof,ndof)=0.0;   % damaged system stiffness matrix
p=0;
q=0;
for ne_d=1:nelm
if dmn(1, ne_d) == 1
    for ii=1:4
        for jj=1:4
          C_d(p+ii,q+jj)=C_d(p+ii,q+jj)+ Ce(ii,jj,ne_d) * (1-
dmn(2,ne_d));     % inserting damaged member
        end
    end
else
    for ii=1:4
        for jj=1:4
              C_d(p+ii,q+jj)=C_d(p+ii,q+jj)+ Ce(ii,jj,ne_d);
        end
    end
end
p=p+2;
q=q+2;
end
C_d(1,1)=1E15;
C_d(2,2)=1E15;
%*******************************
% ei=eig(K,M);
[evec_d,eval_d]=eig(C_d,M_d);
fhz_d(ndof,ndof)=0.0;
for i=1:ndof
eval_d(i,i)=sqrt(eval_d(i,i));
fhz_d(i,i)= eval_d(i,i)/(2*pi);
end
Pho_d = zeros(ndof);
Pho_d = Pho_d + evec_d;
%Checking mass normalized mode shape
a_d=zeros(1, ndof);
for i=1:ndof
a_d(i) = Pho_d(:,i).' * M_d * Pho_d(:,i);
if a_d(i)==1.00
else
    Pho_d(:,i) = 1/sqrt(a_d(i)) * Pho_d(:,i);
end
end
%i th modal stiffness K_i_d        %K_i_d=Pho_i_d'T * C_d * Pho_i_d
Ki_d=zeros(1,ndof);
for i=1:ndof
```

```
Ki_d(i)= Pho_d(:,i).' * C_d * Pho_d(:,i);
end
%contribution of the j th member to the i th modal stiffness K_ij_d
%K_ij_d=Pho_i_d'T * Cj_d * Pho_i_d
Cj_d=Cj;
for i=1:nelm
if dmn(i) == 1
  Cj_d(:,:,i)=(1-dmn(2,i)) * Cj(:,:,i);
end
end
%**************************
Kij_d=zeros(1,ndof,nelm);
for i= 1:ndof
for j=1:nelm
  Kij_d(:,i,j)= Pho_d(:,i).' * Cj_d(:,:,j) * Pho_d(:,i);
end
end
%F_ij_d=K_ij_d/K_i_d
% fraction of modal energy for i th mode that is concentrated in the
% j th member(the sensitivity of the j th member to the i th mode)
Fij_d(1,ndof,nelm)=0.0;
for i=1:ndof
for j=1:nelm
  Fij_d(:,i,j)=Kij_d(:,i,j)/Ki_d(:,i);
end
end
%Ej represents material stiffness property
Ej=zeros(1,nelm)+E;
%Cjo involves only geometric quantities
Cjo(ndof,ndof,nelm)=0.0;
for j=1:nelm
%   Co(j)=C(:,:,j)/Ej;
Cjo(:,:,j)=Cj(:,:,j)/Ej(j);
end
% plotting mode shapes_ damaged
plot_Pho_d=zeros(20,3);
for j=1:3
  jko=2*j-1;
for i=1:20
  ok=2*i-1;
  plot_Pho_d(i,j)=Pho_d(ok,jko);
end
end
figure(2)
subplot(1,3,1)
x=linspace(0,100,20);
plot(plot_Pho(:,1),x,'-r.')
hold on
plot(plot_Pho_d(:,1),x,'-b.')
title('Mode 1')
legend({'Undamaged','Damaged'},'Location','southeast')
xlabel('Modal Amplitude')
ylabel('Location')
hold off
```

```
%
subplot(1,3,2)
x=linspace(0,100,20);
plot(plot_Pho(:,2),x,'-r.')
hold on
plot(plot_Pho_d(:,2),x,'-b.')
title('Mode 2')
legend({'Undamaged','Damaged'},'Location','southeast')
xlabel('Modal Amplitude')
ylabel('Location')
hold off
%
subplot(1,3,3)
x=linspace(0,100,20);
plot(plot_Pho(:,3),x,'-r.')
hold on
plot(plot_Pho_d(:,3),x,'-b.')
title('Mode 3')
legend({'Undamaged','Damaged'},'Location','southeast')
xlabel('Modal Amplitude')
ylabel('Location')
hold off
%%
t = flip(0:0.1:10);
r = 1/2.1335*t+2.286;
[X,Y,Z] = cylinder(r);
h=100;
Z=Z*h;
figure(3)
surf(X,Y,Z)
daspect([1 1 1])
arrow = mArrow3([-10 10 dmgmem(1,1)],[-5 5 dmgmem(1,1)],
'facealpha', 1, 'color', 'red', 'stemWidth', 0.1);
title('Damage indicated on the tower model')

%% Damage localization indicator
%Ej_d represnts damaged material stiffnes property
Ej_d=Ej;
for i=1:nelm
if dmn(1,i) == 1
    Ej_d(i)=Ej_d(i)* (1-dmn(2,i));
end
end
c(ndof,nelm)=0.0;
d(ndof,nelm)=0.0;
e(ndof,nelm)=0.0;
sumofc(1,ndof)=0.0;
c_d(ndof,nelm)=0.0;
d_d(ndof,nelm)=0.0;
e_d(ndof,nelm)=0.0;
sumofc_d(1,ndof)=0.0;
Kie_d=zeros(1,ndof);
Kie=zeros(1,ndof);
Bij_top(ndof,nelm)=0.0;
```

```
Bij_bot(ndof,nelm)=0.0;
Bj_top(1,nelm)=0.0;
Bj_bot(1,nelm)=0.0;
Bj(1,nelm)=0.0;
for j=1:nelm
     for i=1:10
        for k=1:nelm
            c_d(i,k)=Pho_d(:,i).' * Cjo(:,:,k) * Pho_d(:,i);  %
damaged
            c(i,k)=Pho(:,i).' * Cjo(:,:,k) * Pho(:,i);
        end
        for o=1:ndof
            sumofc_d(o)=sum(c_d(o,:));
            sumofc(o)=sum(c(o,:));
        end
    d_d(i,j)=Pho_d(:,i).' * Cjo(:,:,j) * Pho_d(:,i);
    d(i,j)=Pho(:,i).' * Cjo(:,:,j) * Pho(:,i);
    e_d(i,j)=d_d(i,j)+sumofc_d(1,i);
    e(i,j)=d(i,j)+sumofc(1,i);
    Kie(i)= Pho(:,i).' * C * Pho(:,i);
    Kie_d(i)= Pho_d(:,i).' * C * Pho_d(:,i);
    Bij_top(i,j)=e_d(i,j)*Kie(i);
    Bij_bot(i,j)=e(i,j)*Kie_d(i);
    end
Bj_top(j)=sum(Bij_top(:,j));
Bj_bot(j)=sum(Bij_bot(:,j));
Bj(j)=Bj_top(j)/Bj_bot(j);
end
% for i=1:nelm
% Bj(i)=sum(Bij(:,i));
% end
B_mean= mean(Bj);
B_std= std(Bj);
% normalization of B value
Zj(1,nelm)=0.0;
for i=1:nelm
Zj(i)=(Bj(i)-B_mean)/B_std;
end
H(1,nelm)=0.0;
kof=2; % k=2, level of significance is 0.024
for i=1:nelm
if Zj(i)>kof
    H(i)=1;
else
    H(i)=0;
end
end
%%
figure(4)
bar(Zj)
title('Case1')
xlabel('Member Number')
ylabel('Indicator Z')
%% Damage severity estimation
```

```
alpha(1:10,1:nelm)=0.0; % the fractional change in the stiffness
for j=1:nelm
   for i=1:10
%     alpha(i,j)=(Pho(:,i).' * Cjo(:,:,j) * Pho(:,i))/(Pho_d(:,i).'
* Cjo(:,:,j) * Pho_d(:,i))*(Kie_d(i))/(Kie(i))-1;
     alpha(i,j)=Bij_bot(i,j)/Bij_top(i,j)-1;
   end
end
sumofalpha(1,1:nelm)=0.0;
for i=1:nelm
sumofalpha(i)=sum(alpha(:,i));
end
%% modal assurance critirean
m=5;
mac_dmg(m)=0.0;
for i=1:m
for j=1:m
    mac_dmg(i,j)=(Pho(:,i)' * Pho(:,j))^2/((Pho(:,i)' *
Pho(:,i))*(Pho(:,j)' * Pho(:,j)));
end
end
figure(5)
barMAC = bar3(mac_dmg);
for k = 1:length(barMAC)
zdata = barMAC(k).ZData;
barMAC(k).CData = zdata;
barMAC(k).FaceColor = 'interp';
end
colormap(jet);
colorbar
xlabel('Mode','fontsize',15)
ylabel('Mode','fontsize',15)
title('Modal Assurance Criterion, m=15','fontsize',15)
box on
%% MIn weighted off diagonal elements
format long
m=5; % target mode shapes
p=10;   % number of sensors
%reduced ndof=ndof/2 ===k? every possible measurement point   ==n
kdof=ndof/2; % k=redndof;
k=kdof;
diagvec(1,1:k)=1; diagmat=diag(diagvec);
B(1:k,1:k,1:k)=0.0;
for i=1:k
B(:,:,i)=diagmat;
B(i,i,i)=0;
end
ttPho(1:k,1:k)=0.0; %nxm
for iii=1:k
aa=2*iii-1;
for jjj=1:k
    bb=2*jjj-1;
    ttPho(iii,jjj)=Pho(aa,bb);
end
```

```
end
tPho=ttPho(:,1:m);
kMAC(m,m,k)=0.0;
for kk=1:k
for i=1:m
    for j=1:m
        kMAC(i,j,kk)=((B(:,:,kk) * tPho(:,i)).' * (B(:,:,kk) *
tPho(:,j)))^2 / (((B(:,:,kk) * tPho(:,i)).' * (B(:,:,kk) *
tPho(:,i))) * ((B(:,:,kk) * tPho(:,j)).' * (B(:,:,kk) *
tPho(:,j))));
    end
end
end
Z=[];
Zk=[];
Zkk=[];
for kk=1:k
for i=1:(m-1)
    for j=(i+1):m
        Zkk(i,j)=kMAC(i,j,kk)*abs(i-j);
    end
    Zk(kk,i)=sum(Zkk(i,:));
end
Z(kk)=sum(Zk(kk,:));
end
%% removing max Z from the list and iteration
%Z(2,:)=1:k;
s_val=[];
s_loc=[];
senloc=[];
for i=1:p
[s_val, s_loc]=max(Z);
Z(s_loc)=0.0;
senloc(i)=s_loc;
end
senlocc=sort(senloc);
sppl=senlocc*2-1;
figure(6)
plot(0.5,sppl,'-o','MarkerEdgeColor','g', 'MarkerFaceColor','k')
hold on
xline([0.5 0.6 0.7 0.8])
title('Sensor placements')
ylabel('Location')
axis([0 1 0 105]);
hold on
%*******************************************
%%         QR decomposition method
t_tPho=tPho.';
[qq, rr, qr_pp] = qr(t_tPho,0);
qrplotto=sort(qr_pp(1:p));
qrplot=qrplotto*2-1;
figure(6)
plot(0.6,qrplot,'-o','MarkerEdgeColor','y', 'MarkerFaceColor','k')
hold on
```

```
%% gen
gen_locs_final1=[15 29 47 69 73 75 79 85 89 93];%[27   69    79    79
     89    91];
gen_locs_final2=[1 15 21 27 33 39 43 59 97 101];%[13   17    31    45
     51    101];
figure(6)
plot(0.7,gen_locs_final1,'-
o','MarkerEdgeColor','b','MarkerFaceColor','k')
hold on
plot(0.8,gen_locs_final2,'-o','MarkerEdgeColor','r',
'MarkerFaceColor','k')
title('Sensor placements')
legend({'MIN','QR','GEN1','GEN2'})
hold off
%% modal assurance critirean min
mac_min(1:p,1:p)=0.0;
min_Pho(1:p,1:p)=0.0;
for h=1:p
  for oo=1:p
    min_Pho(h,oo)=ttPho(senlocc(h),senlocc(oo));
  end
end
for i=1:p
for j=1:p
    mac_min(i,j)=(min_Pho(:,i)' * min_Pho(:,j))^2/((min_Pho(:,i)' *
min_Pho(:,i))*(min_Pho(:,j)' * min_Pho(:,j)));
end
end
figure(7)
barMAC = bar3(mac_min);
for k = 1:length(barMAC)
zdata = barMAC(k).ZData;
barMAC(k).CData = zdata;
barMAC(k).FaceColor = 'interp';
end
colormap(jet);
colorbar
xlabel('Sensor','fontsize',10)
ylabel('Sensor','fontsize',10)
title('AutoMAC matrix generated by minimizing weighted off-diagonal
elements','fontsize',12)
box on
%% modal assurance critirean qr
mac_qr(1:p,1:p)=0.0;
qr_Pho(1:p,1:p)=0.0;
for h=1:p
  for oo=1:p
    qr_Pho(h,oo)=ttPho(qrplotto(h),qrplotto(oo));
  end
end
for i=1:p
for j=1:p
    mac_qr(i,j)=(qr_Pho(:,i)' * qr_Pho(:,j))^2/((qr_Pho(:,i)' *
qr_Pho(:,i))*(qr_Pho(:,j)' * qr_Pho(:,j)));
```

```
end
end
figure(8)
barMAC = bar3(mac_qr);
for k = 1:length(barMAC)
zdata = barMAC(k).ZData;
barMAC(k).CData = zdata;
barMAC(k).FaceColor = 'interp';
end
colormap(jet);
colorbar
xlabel('Sensor','fontsize',10)
ylabel('Sensor','fontsize',10)
title('AutoMAC matrix generated by QR-decomposition.','fontsize',12)
box on

%*********************************************
%         Genetic algorithm
tic
sens_ite=100;
for jkk=1:sens_ite
% function a=geneticalalgorithm()
f_max=[];
f_avg=[];
eep=[];
ite=130; %criteria 1
N=100;
gen_sen_locs(1:N,1:p,1:ite)=0.0;
sorted_gen_sen_locs(1:N,1:p,1:ite)=0.0;
for ll=1:ite
%% nulling matrices
in_pop(1:N,1:p)=0.0;
sp=[];
ind_tPho(1:k,1:m)=0.0;
E11=[];
E1=[];
ind_MAC=[];
E222=[];
E22=[];
E2=[];
ind_fit=[];
nats=[];
parent1=[];
parent2=[];
child=[];
ind_parent=[];
soo=[];
crit_E11=[];
crit_E1=[];
ind_MAC(m,m)=0.0;
crit_ind_MAC=[];
crit_E222=[];
crit_E22=[];
crit_E2=[];
```

```
crit_ind_fit=[];
%% natural selection
for i=1:N
k=kdof;
in_pop(i,:)=randperm(k,p);
spp(i,:)=sort(in_pop(i,:));
sp=zeros(i,k);
for j=1:k
  for o=1:p
      if j == spp(i,o)
         sp(i,j)=1;
      end
  end
end
ind_tPho(1:k,1:m)=0.0;
for h=1:k
if sp(i,h) == 1
ind_tPho(h,:)=tPho(h,:);
end
end
for g=1:m
for l=1:m
E11(g,l,i)=(tPho(:,g).' * tPho(:,l) - ind_tPho(:,g).' *
ind_tPho(:,l));
end
end
E1(i)=max(max(E11(:,:,i)));
% ind_MAC(m,m)=0.0;
for z=1:m
for x=1:m
ind_MAC(z,x)=(ind_tPho(:,z)' * ind_tPho(:,x))^2/((ind_tPho(:,z)' *
ind_tPho(:,z))*(ind_tPho(:,x)' * ind_tPho(:,x)));
end
end
% E22(1,m)=0.0;
% E222(m,m,N)=0.0;
for r=1:(m-1)
for t=(r+1):m
E222(r,t,i)=ind_MAC(r,t)*exp(2-abs(r-t));
end
E22(r,i)=sum(E222(r,:,i));
end
E2(i)=sum(E22(:,i));
ind_fit(1,i)=1/(1+E1(i));
ind_fit(2,i)=1/(1+E2(i));
end
indtest=ind_fit;
nat_sel=10;
nats(1,nat_sel)=0.0;
for j=1:2
for i=1:nat_sel
[nat_val(j), nat_loc(j)]=max(ind_fit(j,:));
ind_fit(j,nat_loc(j))=0.0;
nats(j,i)=nat_loc(j);
```

```
end
end
for i=1:nat_sel
gen_sen_locs(i,:,ll)=in_pop(nats(1,i),:);
end
%% crossover function
cf=zeros(1,p);
for i=1:(round(p/2)-1)
cf(i)=1;
end
cf(round(p/2)+1)=1;
cross_sel=80;
% creating parents
for i=1:cross_sel
parent1(i,:)=randperm(k,p);
parent2(i,:)=randperm(k,p);
for j=1:p
if cf(j) == 1
child(i,j)=parent1(i,j);
else
  for ko=1:j
      if parent2(i,j) == child(i,ko)
         child(i,j)=parent1(i,j);
      else
         child(i,j)=parent2(i,j);
      end
  end
end
end
end
gen_sen_locs((nat_sel+1):(nat_sel+cross_sel),:,ll)=child(:,:);
%% mutation from an individual parent
mut_sel=10;
for i=1:mut_sel
ind_parent(i,:)=randperm(k,p);
end
% ind_parren=ind_parent;
for i=1:mut_sel
ind1 = round(1 + (p-1) * rand(1,1));
ind2 = round(1 + (k-1) * rand(1,1));
for j=1:p
if ind2 ==ind_parent(i,j)
  ind2 = round(1 + k * rand(1,1));
       for jj=1:p
        if ind2 ==ind_parent(i,j)
           ind2 = round(1 + k * rand(1,1));
        end
       end
end
end
ind_parent(i,ind1)=ind2;
end
gen_sen_locs((nat_sel+cross_sel+1):(nat_sel+cross_sel+mut_sel),:,ll)
=ind_parent(:,:);
```

```
%% criterian 1
% population==gen_sen_locs
sorted_gen_sen_locs=sort(gen_sen_locs,2);
soo(1:N,1:k)=0.0;
for pp=1:N
  for j=1:k
      for o=1:p
          if j == sorted_gen_sen_locs(pp,o,ll)
             soo(pp,j)=1;
          end
      end
  end
ind_tPho(1:k,1:m)=0.0;
for h=1:k
  if soo(pp,h) == 1
    ind_tPho(h,:)=tPho(h,:);
  end
end
for g=1:m
for l=1:m
crit_E11(g,l,pp)=(tPho(:,g).' * tPho(:,l) - ind_tPho(:,g).' *
ind_tPho(:,l));
end
end
crit_E1(pp)=max(max(crit_E11(:,:,pp)));
% ind_MAC(m,m)=0.0;
for z=1:m
for x=1:m
crit_ind_MAC(z,x)=(ind_tPho(:,z)' *
ind_tPho(:,x))^2/((ind_tPho(:,z)' * ind_tPho(:,z))*(ind_tPho(:,x)' *
ind_tPho(:,x)));
end
end
% E22(1,m)=0.0;
% E222(m,m,N)=0.0;
for r=1:(m-1)
for t=(r+1):m
crit_E222(r,t,pp)=crit_ind_MAC(r,t)*exp(2-abs(r-t));
end
crit_E22(r,pp)=sum(crit_E222(r,:,pp));
end
crit_E2(pp)=sum(crit_E22(:,pp));
crit_ind_fit(1,pp)=1/(1+crit_E1(pp));
crit_ind_fit(2,pp)=1/(1+crit_E2(pp));
end
%finding max f
[f_max(1,ll), loc_f_max(1,ll)]=max(crit_ind_fit(1,:));
f_avg(1,ll)=mean(crit_ind_fit(1,:));
[f_max(2,ll), loc_f_max(2,ll)]=max(crit_ind_fit(2,:));
f_avg(2,ll)=mean(crit_ind_fit(2,:));
eep(1,ll)=abs(f_max(1,ll)-f_avg(1,ll));
eep(2,ll)=abs(f_max(2,ll)-f_avg(2,ll));
end
```

```
% extracting  sensor  locations aka parents with maximum fit from
ite=500
% iteration this populations can be used for sensor placement to
check
% optimal sensor is doing good.
mode1_vals_freq=[];
for oo=1:ll
max_fit_parents1(oo,:)=sorted_gen_sen_locs(loc_f_max(1,oo),:,oo);
max_fit_parents2(oo,:)=sorted_gen_sen_locs(loc_f_max(2,oo),:,oo);
end
% finding most frequently occured sensor locations from above
population
% this will extract optimal placement from number of iterations.
for hj=1:p
[mode1_vals_freq1(1,hj),
mode1__vals_freq1(2,hj)]=mode(max_fit_parents1(:,hj),1);
[mode1_vals_freq2(1,hj),
mode1__vals_freq2(2,hj)]=mode(max_fit_parents2(:,hj),1);
end
% removing davtagdsan location from moded placements
mode2_vals1(jkk,:)=mode1_vals_freq1(1,:)*2-1;
mode2_vals2(jkk,:)=mode1_vals_freq2(1,:)*2-1;
end
gen_locs_final1=mode(mode2_vals1,1);
gen_locs_final2=mode(mode2_vals2,1);
figure(6)
plot(0.7,gen_locs_final1,'-o')
hold on
plot(0.8,gen_locs_final2,'-o')
title('Sensor placements')
legends({'MIN','QR','GEN1','GEN2'})
hold off
toc



%% modal assurance critirean genetic
mac_gen1(1:p,1:p)=0.0;
gen_Pho1(1:p,1:p)=0.0;
for h=1:p
  for oo=1:p
    gen_Pho1(h,oo)=ttPho((gen_locs_final1(h)+1)/2,(gen_locs_final1(o
o)+1)/2);
  end
end
for i=1:p
for j=1:p
    mac_gen1(i,j)=(gen_Pho1(:,i)' *
gen_Pho1(:,j))^2/((gen_Pho1(:,i)' * gen_Pho1(:,i))*(gen_Pho1(:,j)' *
gen_Pho1(:,j)));
end
end
figure(9)
```

```
barMAC = bar3(mac_gen1);
for k = 1:length(barMAC)
zdata = barMAC(k).ZData;
barMAC(k).CData = zdata;
barMAC(k).FaceColor = 'interp';
end
colormap(jet);
colorbar
xlabel('Sensor','fontsize',10)
ylabel('Sensor','fontsize',10)
title('AutoMAC matrix generated by genetic algorithm with maximum
error criteria','fontsize',12)
box on
%%
mac_gen2(1:p,1:p)=0.0;
gen_Pho2(1:p,1:p)=0.0;
for h=1:p
  for oo=1:p
    gen_Pho2(h,oo)=ttPho((gen_locs_final2(h)+1)/2,(gen_locs_final2(o
o)+1)/2);
  end
end
for i=1:p
for j=1:p
    mac_gen2(i,j)=(gen_Pho2(:,i)' *
gen_Pho2(:,j))^2/((gen_Pho2(:,i)' * gen_Pho2(:,i))*(gen_Pho2(:,j)' *
gen_Pho2(:,j)));
end
end
figure(10)
barMAC = bar3(mac_gen2);
for k = 1:length(barMAC)
zdata = barMAC(k).ZData;
barMAC(k).CData = zdata;
barMAC(k).FaceColor = 'interp';
end
colormap(jet);
colorbar
xlabel('Sensor','fontsize',10)
ylabel('Sensor','fontsize',10)
title('AutoMAC matrix generated by genetic algorithm with weighted
off-diagonal criteria','fontsize',12)
box on
```