Saitta S., Kripakaran, P., Raphael, B. and Smith, I.F.C. "Improving System Identification Using Clustering" Journal of Computing in Civil Engineering, Vo1 22, No 5, 2008, pp 292-302. http://cedb.asce.org Copyright ASCE

Improving System Identification using Clustering

Sandro Saitta[1], Prakash Kripakaran[2], Benny Raphael[3], Ian F.C. Smith[4]

Abstract

System identification involves identification of a behavioral model that best explains the measured behavior of a structure. This research uses a strategy of generation and iterative filtering of multiple candidate models for system identification. The task of model filtering is supported by measurement cycles. During each measurement cycle, the location for subsequent measurement can be chosen using the predictions of current candidate models. In this paper, data mining techniques are proposed to support such measurement-interpretation cycles. Candidate models, representing possible states of a structure, are clustered using a technique that combines principal component analysis and K-means clustering. Representative models of each cluster are used to place sensors for subsequent measurement on the basis of the entropy of their predictions. Models are filtered from candidate model sets using new measurements. Results show that clustering is necessary to identify the different groups of candidate models. The entropy of predictions is found to be a valid stopping criterion for iterative sensor addition. While measurement-interpretation cycles can lead to a unique model for structures with low levels of complexity, engineers may be left with large numbers of models for structures with higher levels of uncertainty. In those situations, clustering is a powerful tool to classify models and thus provide much fewer representative models to engineers for further decision making.

1 Introduction

Recently, the use of sensors for structural health monitoring (Brownjohn, 2007) increased exponentially. Large numbers of sensors lead to enormous amounts of data. Often, data are either redundant or meaningless, thereby complicating data management. It is thus important to select and place sensors so that maximum useful information is obtained. This process, which stands upstream from data interpretation, is known as sensor placement. In civil engineering, and in other engineering domains, this process is iterative. After placing an initial set of sensors, measurements are taken and sensors can be added afterward. To support this iterative process, data mining techniques such as clustering are helpful.

Sensors are increasingly used worldwide for tasks such as fault diagnosis (Camelio et al., 2005) and automatic control (Culler and Hong, 2004). The field of sensor configuration has emerged recently and research concerning sensor networks is now emerging in parallel.

[1]Grad. Res. Assist. in Comp. Sc., IMAC, Struct. Eng. Inst., Station 18, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland. E-Mail: sandro.saitta@epfl.ch.

[2]Post Doc. Res. in Civil Eng., IMAC, Struct. Eng. Inst., Station 18, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland. E-Mail: prakash.kripakaran@epfl.ch.

[3]Assist. Prof. of Civil Eng., Department of Building, National University of Singapore, 117566, Singapore. E-Mail: bdgbr@nus.edu.sg.

[4]Prof. of Civil Eng., F. ASCE, IMAC, Struct. Eng. Inst., Station 18, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland. E-Mail: ian.smith@epfl.ch.

Examples of the interest in this field are the special issue of Communications of the ACM on wireless sensor networks in 2004 and the publication of a new journal, ACM Transactions on Sensor Networks, in 2005. Moreover, research evolves in managing these sensor networks mainly to satisfy the always growing user needs (Mullen et al., 2006). Work on sensors is carried out in areas such as multi-sensor management (Xiong and Svensson, 2002), reliability (Bagajewicz and Sanchez, 2000) and uncertainty (Guratzsch and Mahadevan, 2006).

One of the most concerned fields is civil engineering. Applications areas in this field include fault detection (Worden and Burrows, 2001), water networks (Robert-Nicoud et al., 2005c) and health monitoring (Meo and Zumpano, 2005). Installation of sensors and measurement campaigns are time-consuming tasks. This motivates the use of a framework for automating the sensor placement process. Li et al. (2006) use norm based techniques to place sensors. Parker et al. (2006) propose experimental validation of their genetic algorithm strategy for sensor placement. In Schulte et al. (2006), a forward-backward selection algorithm is envisaged for optimal sensor placement. Minimization of an information entropy criterion is used in Papadimitriou et al. (2000). All of these studies involved structural dynamics contents and have yet to be evaluated with static measurement data.

One of the most important reasons for making measurements is system identification (Ljung, 1999), where the idea is to understand the behavior of a structure. In this case, the challenge is to determine the true state of the structure according to measurements. System identification can be model-based. In this case, goals are to find models and estimate the model parameters that best match measurements. Part of this task is known as parameter estimation or model updating. Existing work in model-based system identification involves matching observations (measurements) with hypotheses (models). For such a task, the use of an optimization technique for minimizing the error between measurements and models is needed. In recent work (Robert-Nicoud et al., 2000), the idea of working with several models in system identification rather than one (albeit with parameters that can be varied) has emerged.

Recently, sensor placement strategies regarding multiple models have been studied (Robert-Nicoud et al., 2005b,a). In Saitta et al. (2006), greedy and global search approaches have been compared for initial sensor placement. Although successful in some situations, global search is not ideal for iterative sensor addition due to higher computation costs. Therefore, the above mentioned references are limited in the way of supporting iterative sensor placement. The sensor placement methodology using multiple models is divided in two parts. In the first part, upstream model generation, there is the initial sensor placement which consist of finding the number and locations of sensors. The next part, is to iteratively add new sensors using measurement data from existing sensors. This iterative process is needed to achieve the final objective of identifying and monitoring the state of the structure. Data mining techniques, such as clustering, can support engineers in this process.

Data mining (Tan et al., 2006; Witten and Frank, 2005) is a field of research concerned with finding patterns in data for both understanding and predicting purposes. Data mining algorithms are especially useful when dealing with amounts of data that are so considerable, human processing is infeasible. Data mining methods have already been successfully applied in research areas such as gene classification, speech processing, image recognition and web mining. More applications can be found in Pal and Mitra (2004). Data mining has also been applied in engineering (Hua et al., 2007; Soibelman and Kim, 2002). Examples of applications

include structural reliability analysis (Deng and Gu, 2005), system identification (Tang et al., 2007) and composite connection behavior (Shirazi Kia et al., 2005). However, all of these contributions use data mining to make predictions. There are engineering tasks in which it is more appropriate to use data mining to extract knowledge from the data (Saitta et al., 2005).

Iterative sensor placement is an example of a task where clustering can be used to support engineers in system identification. The goal of clustering (Webb, 2002; Tan et al., 2006) is to group data points that are similar according to a given similarity metric (by default Euclidean distance is used). Clustering usually aims at finding compact and clearly separated clusters. Clustering techniques have been applied in domains such as sensory time series (Yin and Yang, 2005) and text mining (SanJuan and Ibekwe-SanJuan, 2006).

This paper presents an iterative methodology for supporting system identification using clustering. The objective of clustering is to group together models that are similar. In each iteration, a new measurement at an appropriate location should eliminate the maximum number of models. An algorithm that finds such a location for subsequent measurement based on cluster information is presented. Section 2 contains a description of concepts behind multiple-model system identification and clustering. The proposed methodology for iterative sensor addition is described in Section 3. Results of applying the methodology on an existing bridge are shown in Section 4. Finally, conclusions drawn from this work are presented in the last Section.

2 Multiple Model System Identification

Traditionally, system identification is treated as an optimization problem in which the difference between model predictions and measurements is minimized. Values of model parameters for which model responses best match measured data are determined by this approach. However, this approach is not reliable because different types of modeling and measurement errors are present (Banan et al., 1994; Sanayei et al., 1997; Catbas et al., 2007). Moreover, they can compensate each other such that the global minimum may be far away from the correct state of the system (Robert-Nicoud et al., 2005c). Therefore, instead of optimizing one model, a set of candidate models is identified in our approach such that their prediction errors lie below a certain threshold value. For this paper, a model is defined as values for a set of parameters. The threshold is computed using an estimate of the upper bound of errors due to modeling assumptions as well as measurements. The set of candidate models is iteratively filtered using subsequent measurements for system identification. This approach could generate an unique model for the structure or a set of models which are equally capable of representing the structure. This depends on parameters chosen for the identification problem and errors.

Modeling assumptions define the parameters for the identification problem. The set of model parameters may consist of quantities such as elastic modulus, connection stiffness and moment of inertia. Each set of values for the model parameters corresponds to a model of the structure. An objective function is used to evaluate the quality of candidate models. The objective function E is defined as follows:

$$E = \begin{cases} \varepsilon & \text{if } \varepsilon > \tau \\ 0 & \text{if } \varepsilon \leq \tau \end{cases} \quad \text{with} \quad \varepsilon = \sqrt{\sum (m_i - p_i)^2} \tag{1}$$

ε is the error which is calculated as the difference between predictions p_i and measurements m_i. τ is a threshold value evaluated from measurement and modeling errors in the identification process. The set of models that have $E = 0$ form the set of candidate models for the structure.

The need for a strategy of generation and iterative filtering of multiple models is demonstrated with a simple truss example. The structure is made of ten bars each with a cross-sectional area of 16 cm^2. Figure 1 shows the truss. Only the displacement at location A is measured using a sensor. The structure has a vertical displacement of 10.5 mm at position A when subject to a vertical load F of 40 kN at the same position. The objective is to detect damage in the truss. Three distinct candidate models are given in Table 1. All of them have predictions at A that lie within 5% of measurement (at point A) and will be part of a candidate model set for this identification problem. The uncertainty in identifying the model that represents correctly the structure is due to errors and lack of sufficient measurements. Including more measurements such as having strain gauges on certain members can filter models from the candidate model set. However, minimizing the difference between errors and measurements can lead to the wrong model. Consequently, multiple models are needed to correctly accommodate system identification. The concept of multiple models significantly affects measurement system design since sensor placement has to be undertaken accounting for several models instead of one.

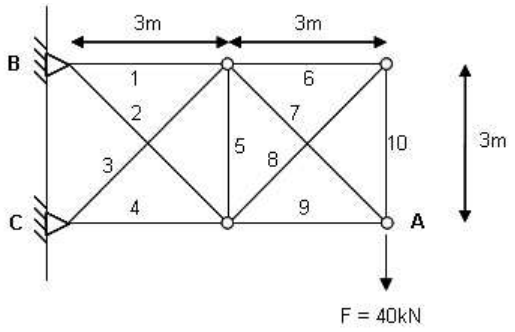

Figure 1: Schema of the truss structure used to justify the need of a multiple model approach for system identification.

A schematic diagram of the software system for system identification is given in Figure 2. It has three modules. The focus of this paper is on the data mining module and its interaction with the measurement system design module. Given sensor measurements and the parameters for the identification problem, the model generation module uses stochastic search to generate the set of candidate models. The set of candidate models are then analyzed using data mining techniques. The result from data mining is used to determine locations for further measurements. Models in the candidate set for which $E \neq 0$ when considering the new measurement are filtered. When this process is carried over cycles, ideally the candidate model set gradually reduces to the model that represents the behavior of the structure.

Case	Damage scenario	Description	Displacement
Model 1	Element 2 damaged	87% area reduction	10.3 mm
Model 2	Element 6 and 7 damaged	69% area reduction	10.1 mm
Model 3	Support B damaged	displacement	11.0 mm

Table 1: Details of three models that can explain the behavior of the truss structure in Figure 1. For each model, the damaged element(s) and the modified area(s) are given. All other elements have an area of $16cm^2$.

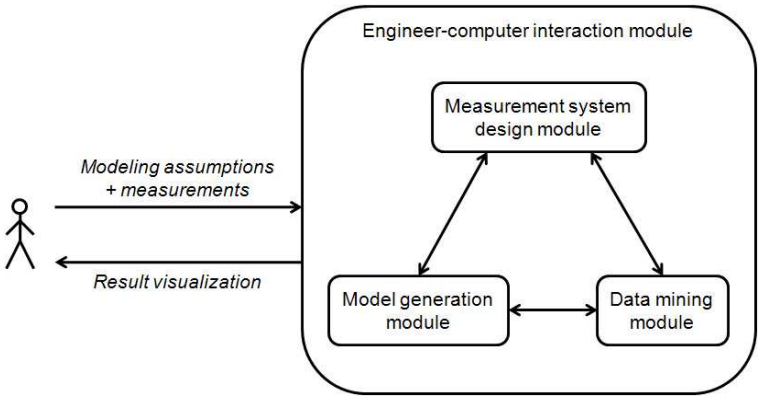

Figure 2: Decision support using multiple-model system identification

2.1 Clustering Multiple Models

In system identification the process goes from measurements (consequences) to a possible model (causes). This is an abductive task. The unreliability of abductive tasks, and the presence of compensating errors, are the motivations for multiple-model system identification. The correct model for the structure should be contained in the model sets given by model generation module. Clustering techniques aid in eliminating incorrect models from these model sets and thus rapidly converge to the correct model. Visualizing distributions of models in multi-dimensional parameter spaces is difficult for engineers without suitable computing tools. The use of a data mining method such as clustering can give engineers an idea of the topology of the candidate model space. This section presents the clustering strategy and then describes the index used to correctly estimate the number of clusters among the models.

Clustering Algorithm

The methodology for grouping models into clusters combines PCA and K-means in order to improve visualization of results. After normalization, the PCA procedure is applied to the models. Using all the principal components, the complete set of models is transformed into the feature space. After that, the number of clusters is estimated using a score function. More details about this step are given in the last paragraphs of this Section. Once the

number of clusters is known, K-means algorithm is applied to the data in the feature space. Table 2 presents the pseudo-code of the methodology used.

Clustering procedure
1. Normalize the data.
2. Transform the data using PCA.
3. Choose the number k of clusters (Section 2.1).
4. **Loop** i from 1 to t
5. Run K-means with k clusters.
6. Evaluate results (Section 2.1).
7. **End**
8. Select cluster i with best results

Table 2: Pseudo-code of the clustering procedure combining PCA and K-means to separate models into clusters. k is the number of clusters and t the number of times K-means is run.

Principal Component Analysis: When a clustering technique such as K-means is applied to data in more than three dimensions, the solution space becomes difficult to represent. PCA is a method for linearly transforming the data to a new and uncorrelated feature space (Jolliffe, 2002). Ultimately, PCA finds a set of principal components (PC) that are sorted such that the first few components explain most of the variability of the data. The first step to obtain the principal components of a data set is to construct the covariance matrix S. Each element of the covariance matrix is given by Equation 2:

$$cov(x, y) = \sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y}) \tag{2}$$

where n is the number of samples. The particular case of $cov(x, x)$ refers to the variance of variable x. The next step is to write the covariance matrix as the product which realizes the eigen decomposition. It is given by Equation 3:

$$S = VLV^T \tag{3}$$

where L is a diagonal matrix that contains the eigenvalues of the covariance matrix S. The columns of V are made by eigenvectors. Each eigenvector is directly related to its eigenvalue. The principal components are the eigenvectors sorted in decreasing order of their eigenvalues. Each sample can then be transformed into the feature space using selected principal components. In the machine learning community, PCA is usually used as a preprocessing technique, for example before a supervised learning algorithm. In this research, PCA is used for visualization purposes. By plotting the two first PCs instead of two randomly chosen parameters, the clusters obtained are easier to visualize.

K-means: The K-means clustering algorithm (Webb, 2002) is widely used in practice. Although it is simple to understand and implement, it is effective only if applied and interpreted correctly. The K-means algorithm divides the data into k clusters according to a given distance measure. Although the Euclidean distance is usually chosen, other metrics may be

more appropriate. More precisely, K-means is a procedure that iterates over k clusters in order to minimize their intra-cluster distances, shown as the measure J in Equation 4

$$J = \sum_{j=1}^{k} \sum_{x_i \in c_j} ||x_i - z_j||^2 \tag{4}$$

where k is the number of clusters, x_i the i^{th} data point and z_j the centroid of cluster c_j. The k starting centroids are chosen randomly among all data points. The data set is then partitioned according to the minimum squared distance. The cluster centers are iteratively updated by computing the mean of the points belonging to the clusters. The process of partitioning and updating is repeated until either the cluster centers or J do not significantly change over two consecutive iterations.

The standard K-means algorithm has two main drawbacks. First, the number of clusters has to be specified by the user a-priori. The next section describes a function to estimate the number of clusters in a data set. Second, the k initial centroids are chosen randomly at the beginning of the K-means procedure. Therefore, running the algorithm two times may result in two different clustering results for the same data. To limit such a problem, K-means is run $t = 20$ times and the best result according to a score function is chosen. This score function is described next.

Optimal Number of Clusters

As stated in the previous Section, the number of clusters is an input to the K-means algorithm and is not known in advance. Moreover, the number of clusters obviously has a crucial impact on the clustering results and therefore on the sensor placement process. If this number is not correctly chosen, K-means will produce clusters of bad quality. These clusters would be of no use to the engineer performing system identification. In this paper, we use a score function derived from Saitta et al. (2007) to: i) estimate the number of clusters and ii) evaluate the quality of the clustering results.

The score function is a function of the combination of two terms: the distance between clusters and the distance inside a cluster. The first notion is defined as the between class distance (*bcd*) whereas the second is the within class distance (*wcd*) . In this research, the *bcd* is defined by Equation 5:

$$bcd = \frac{1}{nk} \sum_{i=1}^{k} d(z_i, z_{tot})^2 \cdot n_i \tag{5}$$

where n is the number of models, k the number of clusters, z_i the centroid of c_i, z_{tot} the centroid of all clusters and n_i the number of models in c_i. The function $dist(x, y)$ is the Euclidean distance between x and y. In this work, the *bcd* indicates how different the k situations are. The *wcd* is given through Equation 6:

$$wcd = \frac{1}{k} \sum_{i=1}^{k} \sqrt{\frac{1}{n_i} \sum_{x \in c_i} d(x, z_i)^2} \tag{6}$$

where the same notation as for Equation 5 stands. The *wcd* gives an overview of the spread of groups of models. For the score function to be effective, it should i) maximize the *bcd*, ii) minimize the *wcd* and iii) be bounded. Maximizing Equation 7 satisfies the above conditions:

$$SF = 1 - \frac{1}{e^{e^{(bcd-wcd)}}} \tag{7}$$

This double exponential reciprocal function has the advantages that the higher the value of the SF, the more suitable the number of clusters. Therefore, with the proposed SF, it is now possible to estimate the number of clusters (groups of models) for a given set of models. The procedure to determine the best number of clusters is to evaluate the SF value for different number of clusters from k_{min} to k_{max}. As for the previous Section, the randomness of K-means, through its starting centroids, has to be taken into consideration. For this, the algorithm is run t times and the maximum value for the score function is chosen. The procedure is described in Table 3. More details can be found in Saitta et al. (2007).

Score Function Procedure
1. **Loop** i from 1 to t
2. **Loop** j from k_{min} to k_{max}
3. Run K-means with j clusters.
4. Calculate score function (SF).
5. **End**
6. **End**
7. Select results corresponding to maximum SF.

Table 3: Procedure to estimate the number of clusters in a data set. t is the number of time K-means is run. k_{min} and k_max are the bound for the number of clusters.

2.2 Sensor Placement using Entropy

The concept of entropy-based sensor placement is explained earlier in this paper to describe how the results from clustering are used for subsequent sensor addition. In the field of model-based system identification, configuring a measurement system can be defined as finding optimal positions for sensors in order to best separate model predictions[1]. Different methods can be used to measure the separation between predictions. For example, variance was compared to entropy as a measure of model separability and entropy was found to be better. Therefore, following Robert-Nicoud et al. (2005b), the notion of entropy is used. The expression used to calculate entropy is the Shannon's entropy function (Shannon and Weaver, 1949) which comes from the field of information theory. Shannon's entropy function represents the disorder within a set. In the present work, a set is an ensemble of predictions for a particular system identification task. The entropy or disorder is maximum when predictions show wide dispersion.

[1]The term *predictions* will be used in place of *model predictions* for readability.

Since the goal is to use information to the maximum, positions with maximum prediction disorder are the most interesting. In other words, the best measurement location is the one with maximum entropy (model predictions have maximum variations). For a random variable X, the entropy $H(X)$ is given by Equation 8:

$$H(X) = -\sum_{i=1}^{|X|} p_i \cdot log(p_i) \tag{8}$$

where p_i are the probabilities of the $|X|$ different possible values of X. For practical purposes, $0 \cdot log(0)$ is taken to be zero. When a variable takes $|X|$ discrete values, the entropy is maximum when all values have the same probability $log(X)$. Thus entropy is a measure of homogeneity in a distribution. A completely homogeneous distribution has maximum entropy.

In the present study, the entropy for a given sensor location is calculated from the histogram of predictions. Given a set of candidate models (Robert-Nicoud et al., 2005b; Raphael and Smith, 2003), the finite element method is used to compute predictions at all possible sensor locations. These predictions can be seen as a matrix in which each row corresponds to predictions for a model and each column is a specific sensor location. At each possible sensor location, a histogram containing predictions is built. Each bar in the histogram represents those models whose predictions lie within that interval. Note that intervals are defined by the accuracy of the measurement devices. At each iteration, the sensor location corresponding to maximum entropy of predictions is chosen. Sensors are therefore sorted in ascending order according to their efficiency in separating model predictions. The probability p_i of an interval is the ratio of the number of predictions r_i in the interval by the total number of predictions r_{tot} (see Figure 3). Therefore, for S possible sensor locations, S histograms are evaluated according to the entropy measure.

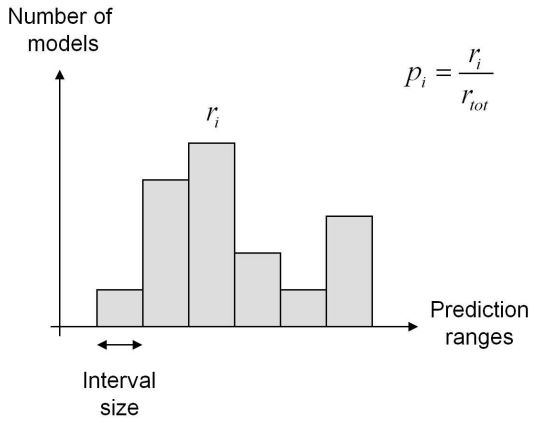

Figure 3: Histogram for a specific sensor position. The x-axis is the sensor prediction range. The y-axis is the number of models. The vertical size of each bar corresponds to the number of predictions lying in the interval. The probability p_i is the ratio of the number r_i of predictions in an interval by the total number of predictions r_{tot}.

3 Methodology

The overall objective of this study is to improve a measurement system - by correctly adding new sensors - in order to support system identification. To achieve this goal, the following methodology combines techniques such as global search, entropy and clustering. A schema of the overall methodology is given in Figure 4 and details about it are given below.

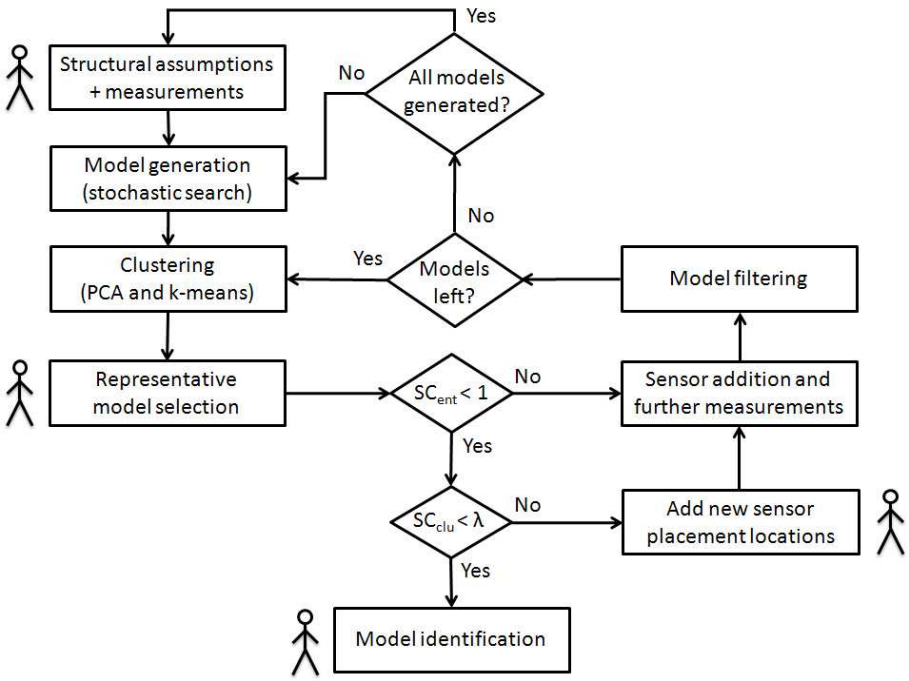

Figure 4: Overall schema showing the methodology for iterative sensor placement using multiple models. The stick person indicates where human-computer interaction is needed.

Structural assumptions and measurements: Assumptions define the parameters of the identification problem. Measurements could be from the initial measurement system or from a sensor that was added during the previous iteration. A method of designing initial measurement systems that is suitable for identification using multiple models is given in Saitta et al. (2006).

Model generation: The next step creates - using stochastic search - a set of candidate models that may represent the real state of the structure. Measurements, a set of model parameters and an objective function (Equation 1) that evaluates models are needed to generate the set of candidate models.

Clustering: Once the models have been generated, the clustering algorithm described in (Section 2.1) is used to group models. Models are grouped into clusters to i) facilitate visualization of the model space and ii) reduce the number of models given to the engineer (the centroid of the cluster is a possible representative model for the entire cluster). Visualization of clusters is improved through the use of principal components. As described earlier, PCA is first applied to models before the K-means algorithm is used (see Section 2.1).

Representative model selection: In the representative model selection step, a few models representing each cluster are selected. Only models which are close to the center of the cluster are selected. In this study, 5% of the total number of models in each cluster are taken to be representative models (with a minimum of 10 models). This number has been chosen after experimental testing. Then, Shannon entropy is used as a measure of prediction separability to identify the next measurement location (see Equation 8). If model sets have high values of entropy, more candidate models can be filtered.

The first stopping criterion, sc_{ent} is using the entropy of remaining sensors. If the entropy of predictions is not significant (below 1) at every sensor location, then $sc_{ent} < 1$. If this is not the case, the next step is *sensor addition and further measurements*. If this is the case, it is then checked if there are multiple clusters using the $sc_{clu} < \lambda$ stopping criterion. sc_{clu} is defined as the maximum distance between all the remaining models and the mean (i.e. center of cluster) of all the models. If $sc_{clu} < \lambda$, where λ is a user-defined constant, a unique cluster is possible. Such a condition may mean that the current set of measurement locations is incapable of further filtering models. The engineer has to provide other measurement locations to the algorithm in order to find the correct model (*add new sensor placement locations* step). If there is only one cluster and the entropy is zero, the center of all remaining models is given to the engineer as the correct model for the structure (*model identification* step).

Sensor addition and further measurements: During this step, entropies of selected representative models are used to find the position of the next sensor. The location with the highest entropy is chosen as the best position for the next measurement. Then, the measurement is taken on the structure.

Model filtering: In this step, sensor measurements at the new location are compared for every candidate model. Candidate models that do not predict the measurement are eliminated from the current set of models.

If there are models left, then the next step is *clustering*. However, if no model is left, then it is likely that all models were not generated by the *model generation* step. While it may be possible to generate all models for a simple problem, it is practically impossible to generate all possible models in a complex structure. In that case, the *model generation* phase is revisited. On the other hand, if all models have been generated, then some assumptions related to modeling the structure are incorrect. Therefore, structure assumptions have to be checked and modified by the engineer (*structure assumptions and measurements* step).

4 Results

4.1 Case study: the Schwandbach Bridge

To demonstrate the methodology for sensor addition, the Schwandbach bridge (designed by Maillart in 1933) is taken as a case study (Figure 5).

This structure is inspected periodically and has been the subject of many verifications as codes have improved, for example Salvo (2006). The Schwandbach bridge is now a pedestrian bridge, although it could be reopened for traffic. Deflection measurements have not been carried out since the 1930s and while the bridge shows no visible evidence of deterioration, the

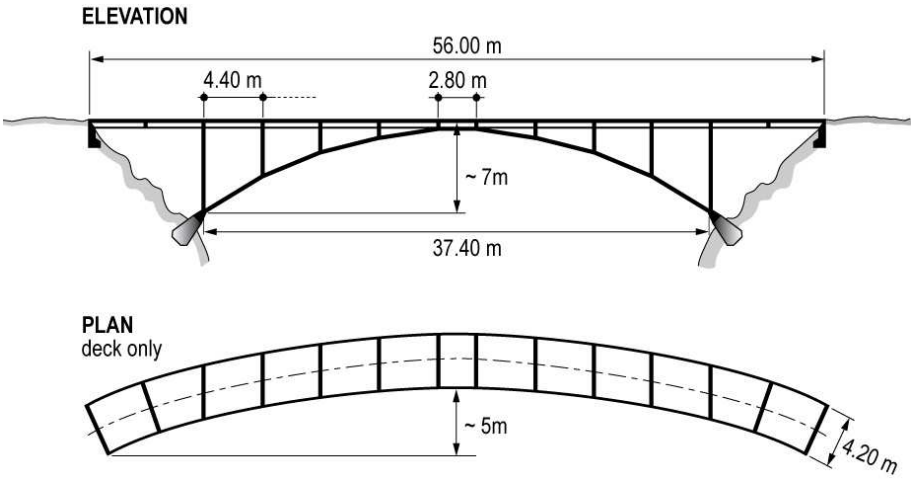

Figure 5: Schema of the Schwandbach bridge used to illustrate the proposed methodology for iterative sensor placement.

question of taking measurements arises periodically. In Switzerland, bridges are traditionally measured for changes in deflection at mid-span during load tests. A single model (usually the design model) is used with the deflection measurement and the loading to determine values for parameters that have some uncertainty, such as the elastic modulus multiplied by the moment of inertia, EI. However, this bridge is too complex for such rudimentary model-calibration strategies.

Details of the analysis at the design stage can be found in Smith and Saitta (2007). While many assumptions are acceptable at the design stage for achieving safety and serviceability, they are not appropriate for interpreting measurements. For example, there is no physical hinge at the extremities of the vertical spandrel elements. These connections cannot be assumed to be fixed either since even small amounts of cracking reduce connection stiffness. Furthermore, not all connections are expected to have the same stiffness due to factors such as relative slenderness and varying locations on the structure. The Schwandbach bridge has 20 such connections. They are shown in Figure 6 using open circles. In this paper, the system identification methodology (see Section 3) is used to determine the behavior of the structure.

In the case of the Schwandbach bridge, the number of permutations and combinations of modeling assumptions - connection stiffnesses - results in several tens of thousands of possible models. Although this case has important technical and historical attributes, these conclusions are equally valid for most ordinary structures of moderate complexity. Rather than "stab" at one model and hope for the best, this paper proposes explicit treatment of multiple models and iterative sensor placement using the methodology described in Section 3.

Bridges are often tested periodically using static loads to check for strength degradation. The response of the bridge for trucks positioned on the bridge is measured using sensors. Engineers estimate the stiffness of the bridge from measured responses and compare those with results from previous tests. In this paper, such a scenario is simulated for the Schwand-

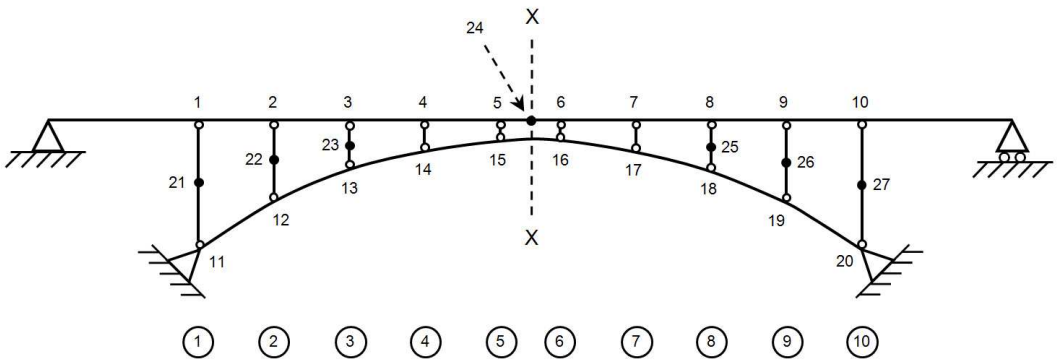

Figure 6: Schematic view of the bridge showing the 20 connections (1-20), the 17 possible sensor locations (1-10, 21-27) and the 10 vertical walls (1-10 circles).

bach bridge. It is schematically represented in Figure 5. For simulation, a three dimensional finite element model of the complete bridge is created. The vertical slab-girder connections and the vertical slab-arch connections are modeled using rotational springs. In this paper, a load test is simulated that involves two trucks. The details of the load test are given in Table 4.

Information	**Value**
Position of rear axle of left truck from left abutment	15 [m]
Distance between trucks	3.7 [m]
Distance front-rear axle	2.6 [m]
Front axle load	17 [kN]
Rear axle load	44 [kN]
Spacing between front wheels	1.8 [m]

Table 4: Details of the two trucks and their positions.

Measurements at different sensor locations (see each example of Section 4.2) are given as input to the model generation module. The parameters of the models generated, however, are the logarithms of the stiffness. In this paper, only inclinometers are used. Sensor precision are $9.5\mu rad$ (micro radian), τ (see Section 2) is taken to be the sum of τ_{meas} ($3\mu rad$) and τ_{pred} ($8\mu rad$).

4.2 Application of the Methodology

Example 1

This example illustrates the ability of the proposed methodology to iteratively add sensors to uniquely identify the system. The bridge has 10 vertical walls and therefore 10 wall-girder

connections and 10 wall-arch connections. For this example, it is assumed that the stiffnesses of the connections in walls 1, 2, 9 and 10 are the same. Other assumptions are (a) symmetry about axis X-X, (b) the stiffness values of the top and bottom connections are equal for each wall and (c) the stiffness values of these connections lie between 10^6 and 10^{12} Nm/rad . Thus there are three parameters in this example. p_1 represents the stiffness of the connections of walls 3 and 8, p_2 for walls 4 and 7 and p_3 for walls 5 and 6. p_1, p_2 and p_3 are permitted to vary between 6 and 12.

For simulation, a model representing the real structure is required. The correct model for this example is given in Table 6. The predictions given by this model are taken as the measurements. The starting measurement system is assumed to consist of inclinometers measuring the rotation at the following locations: 1, 10 and 24 (Figure 6). Since there are only three parameters, models can be directly visualized in three-dimension plots.

1000 candidate models are generated for this example. At the first iteration, only sensor locations on the deck can be chosen. This decision follows from the fact that it is easier to place sensors on the deck of the bridge. When the entropy for sensors on the deck is below 1 ($sc_{ent} < 1$), then other sensor locations are also included. Table 5 shows the number of models remaining and the selected sensors.

Iterations	**0**	**1**	**2**	**3**	**4**
Number of models	1000	926	907	906	10
Selected sensor	4	6	5	23	

Table 5: Evolution of the number of models at each iteration for example 1. The selected sensors are given as well.

The first observation concerns the sensors on the deck. They filter fewer candidate models compared to the sensor on the vertical wall. In fact, a measurement system with just one sensor at 23 can uniquely identify the system. After four iterations, the entropy values at the remaining sensor locations are close to zero. Therefore, there is no need to add more than four sensors. At iteration 0, the sc_{clu} (see Section 3) is 3.59. After four iterations it drops to 1.20. If the precision required in each parameter is 1.0, then this set of models is interpreted as a single cluster by the engineer. Consequently, the mean of this cluster is calculated, and the model closest to this mean is given to the engineer. A plot of the models in the original parameter space at iteration 0 and 4 are given in Figure 7. The model found as well as the correct model (which is known for this problem) are given in Table 6.

Parameters	p_1	p_2	p_3
Correct model	8.0	8.0	8.0
Model found	8.2	7.4	8.1

Table 6: Model found and correct model in the case of example one (in log scale).

Figure 7 shows how the candidate model space decreases from iteration 0 to 4. From

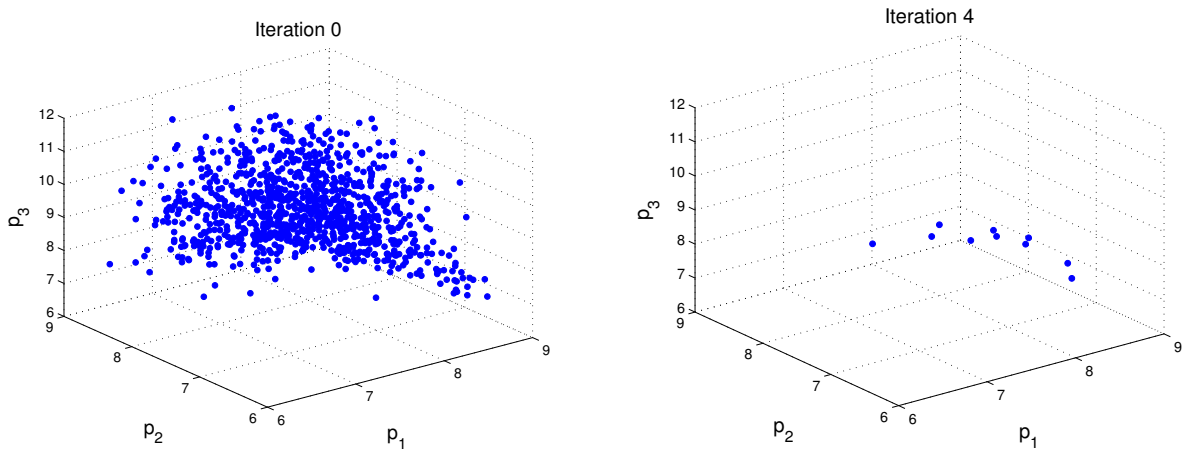

Figure 7: Models in the original parameter space at iteration 0 (left) and 4 (right).

Table 6 it is noted that the model found is very close to the correct model for this example. This is especially true for parameters p_1 and p_3. This illustrates the ability of the proposed methodology to uniquely identify the system. This example has only three parameters and a unique cluster of models. A more complex example is shown below.

Example 2

In practical situations, the identification problem may involve dozens of parameters. In such cases, it is impossible to visualize the model space as was done for the previous example for reasons of high dimensionality. The identification methodology is illustrated for such an example. The Schwandbach bridge is again considered, however, with more elaborate modeling assumptions. Symmetry about X-X (see Figure 6) is assumed. This example models 10 parameters. Each parameter corresponds to two connections, one on either side of X-X. Here, the starting measurement system consists of inclinometers at the following locations: 1, 7, 11, 23 and 25 (Figure 6). The stiffness values (K) of each connection vary between 10^2 and 10^{12} Nm/rad. 1719 candidate models are generated for this example. Input data for the PCA part of the methodology are the stiffness values of 10 sets of connections.

The number of clusters is estimated using the score function. The procedure in Table 2 is thus executed. The starting point for PCA is a matrix where each row is a different model and each column contains values of a parameter. Figure 8 shows the curve of the score function from $k_{min} = 2$ to $k_{max} = 10$ clusters at the very first iteration.

The first observation from Figure 8 is regarding the global maximum achieved for $k = 6$. This number has to be interpreted carefully since values for $k = 5$, $k = 7$ or even $k = 10$ are very close to the global maximum. This result has to be combined with the PCA plot of the models (Figure 9). The role of the engineer here is to carefully interepret these results. This is generally required of the user in any data mining task. According to the results of Figure 8, the number of clusters is chosen to be six for this case. The clustering results after applying Table 3 procedure are given in Figure 9.

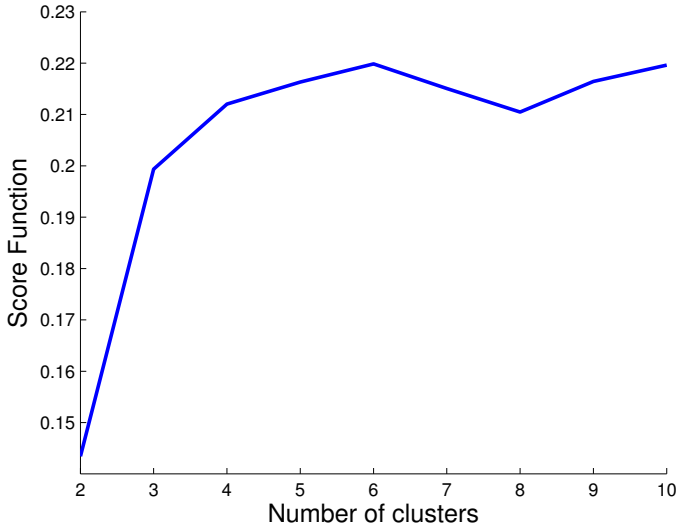

Figure 8: Curve of the score function from $k_{min} = 2$ to $k_{max} = 10$ clusters. The best value is taken over $t = 20$ runs.

In Figure 9, every point represents a model. Although all principal components are used in the K-means algorithm, only the two first components are used for visualization. The reader must be aware of the fact that other dimensions (i.e. other principal components) explain these data. Even if not well defined, clusters are already visible. In addition, clusters also contain outliers. This is not an issue since the score function is using the cluster size as a weight in Equation 5 and 6. Again this plot taken alone is not enough to estimate the correct number of clusters. This is mainly due to the dimensionality of the data set and the overlapping between clusters. Combined with Figure 8, it can help the engineer to estimate the most reliable number of clusters. The centroid of each cluster defines a possible state of the structure. Instead of having to examine 1719 models, the engineer can examine the six groups of models, each represented by its center. Indeed, the center of each cluster represents a bridge with a particular set of stiffness values for the connections.

The next step is to iteratively add sensors to reduce the total number of models. Representative models are selected in each cluster for evaluating entropy. Representative models are chosen around each cluster centroid. This way, only models that *represent* the cluster are taken into account. The selected set of representative models is 5% of the total number of remaining models. This set is proportionate to the cluster size (i.e. the number of models inside the cluster). Therefore, bigger clusters have more influence on the selection of the next sensor. Entropy is calculated at every remaining sensor location for the representative model predictions and a sensor is added at the location with highest entropy . The entropy value is found to be a valid stopping criteria (sc_{ent}) for the methodology. Once the new sensor is known, a new measurement is taken. All models whose predictions do not match the new measurement are eliminated. Models with a high error are filtered for the next iteration. This is repeated until the entropy of model predictions is zero for every sensor location. At

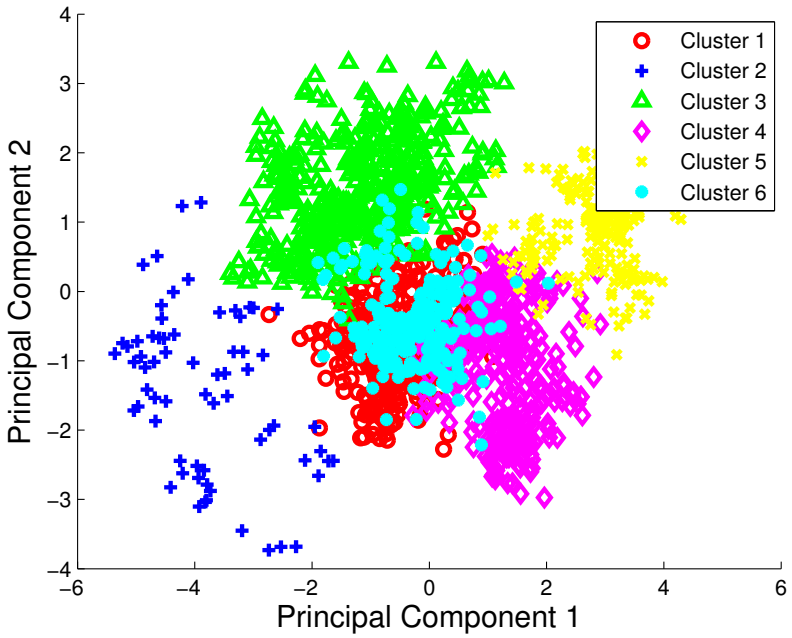

Figure 9: Clustering results at the very first iteration. Every point represents a model using the two first principal components (out of 10).

each iteration, the number of models is either reduced or the same.

Iterations	**0**	**1**	**2**	**3**
Number of models	1719	923	243	71
Selected sensor	8	21	26	

Table 7: Evolution of the number of models at each iteration for example 2. The selected sensors are given as well.

In this case, the methodology is unable to converge to the unique model for the bridge. At iteration 3, multiple clusters are still present. Indeed, sc_{clu} in iterations zero and three are respectively 9.68 and 5.85. This indicates that the remaining sensor locations are incapable of further reducing the number of candidate models. At this juncture, the engineer can consider adding more load cases, including other sensor types and augmenting the set of sensor locations. The engineer could also opt to look at a representative model (cluster centroid) from each cluster.

Table 7 shows that sensors on the deck are useful for reducing the number of candidate models in this example. This was not the case in the previous example. The choice of sensor locations is dependant on the parameter set.

Table 8 shows the entropy of each sensor for iteration 0 to 2 (all entropy values are 0 at

iteration 3). From Table 8, it is observed that locations on the vertical walls have a higher entropy and are thus better than locations on the deck to identify the system. In iterations 0 and 1, all locations on the deck have an entropy that is smaller than entropies for locations on the walls. The table also shows that the best location for a particular iteration is dependent on the locations chosen in the previous iteration. At iteration $i+1$ the entropy for a given sensor is not the same that at iteration i. After each iteration, models are filtered, and therefore the entropy of each remaining sensor may be different. In this example no unique model is found, rather the model closest to the mean of every cluster is given to the engineer. The proposed models as well as the correct model are given in Table 9.

Iteration 0		**Iteration 1**		**Iteration 2**	
Sensor	Entropy	Sensor	Entropy	Sensor	Entropy
26	3.58	**21**	2.47	**26**	1.49
21	3.45	27	1.93	22	1.31
27	3.12	26	1.88	2	0.00
22	3.12	22	1.64	3	0.00
8	2.46	3	0.86	4	0.00
3	2.30	7	0.67	5	0.00
4	2.19	2	0.00	6	0.00
2	2.04	4	0.00	7	0.00
7	1.96	5	0.00	9	0.00
9	1.86	6	0.00	27	0.00
6	1.46	9	0.00		
5	0.90				

Table 8: Selected sensors and entropy corresponding to every sensors. Values in bold represent the chosen sensors. After iteration 2, the entropy value is zero for every remaining sensor location.

Parameters	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
Correct solution	3.0	3.0	7.0	7.0	10.0	10.0	7.0	7.0	3.0	3.0
Solution 1	5.1	6.1	5.0	4.5	10.0	10.0	6.6	6.3	4.7	5.1
Solution 2	7.2	6.6	7.1	7.4	9.9	10.0	6.8	6.4	7.7	6.9
Solution 3	7.1	4.5	6.1	7.1	10.1	10.0	7.1	5.7	5.5	4.0
Solution 4	3.2	3.3	5.2	5.6	10.0	10.1	5.4	6.6	3.6	6.2
Solution 5	4.7	7.8	5.0	4.8	7.6	10.0	7.4	5.2	8.3	9.6
Soltuion 6	5.0	6.2	6.8	6.5	10.1	10.1	6.9	6.4	5.7	5.8

Table 9: Models found in the case of example 2 and correct solution of the problem (in log scale).

From Table 9 it is noted that more than one model is proposed as a correct model. Among

them, only solution 4 is closest to the correct model. The values for the different parameters show some common features among the solutions. Nearly all models have a value of 10 for both p_5 and p_6. Since the variation in these parameters is very small, they are likely have a much larger influence on predictions than the other parameters. The other parameters do not significantly affect the behavior of the bridge. In other words, the connections closer to the ends could be modeled as hinged or rigid and it would not generate changes in displacements that are detectable with inclinometers considered in this study. However, sensor technology is improving day-by-day and precision of sensors are gradually increasing. In the future, this will enable engineers to uniquely identify the model for even complex structures.

5 Conclusions

The study described in this paper results in the following conclusions:

- The use of K-means, for grouping models, and PCA for displaying them helps in visualizing the solution space. This support is needed since the methodology involves the use of several models for system identification.
- The score function is used to find the most reliable number of clusters in the model space, hence resolving the main issue of K-means concerning the user-defined number of clusters.
- The methodology helps engineers by providing cluster centers as possible models that explain the structural behavior. This is useful information for the engineer who can then, for example, adjust the focus of on-site inspection.
- The choice of sensor locations is dependent on the parameter set (example 1 and 2).
- The entropy value obtained at every sensor position is an iterative indication of the number of sensors needed on the structure. It is therefore used as a stopping criteria.

Several extensions to this work are in progress. Application of other clustering algorithms is under study. Work is in progress towards devising a standard way of estimating the number of representative models required from each cluster to identify subsequent measurement locations. The number of candidate models required for correct system identification is being treated probabilistically in ongoing work. Other data mining tools such as feature selection are being studied to extract information from parameter values. Finally, the search method for model generation is also being improved.

6 Acknowledgments

This work is funded by the Swiss National Science Foundation under grant no. 200020-109257. The authors would like to thank E. Bruehwiler, S. Ravindran and A. Salvo for their assistance with the Schwandbach Bridge case study and Dr. P. Lestuzzi for his comments on a preliminary version of this article.

References

Bagajewicz, M. and Sanchez, M. (2000). Cost-optimal design of reliable sensor networks. *Computers and Chemical Engineering*, 23(11):1757–1762.

Banan, M., Banan, M., and Hjelmstad, K. (1994). Parameter estimation of structures from static response. i. computational aspects. *Journal of Structural Engineering*, 120(11):3243–3258.

Brownjohn, J. (2007). Structural health monitoring of civil infrastructure. *Philosophical Transactions of the Royal Society A*, 365(1851):589–622.

Camelio, J., Hu, S., and Yim, H. (2005). Sensor placement for effective diagnosis of multiple faults in fixturing of compliant parts. *Transactions of the ASME*, 127:68–74.

Catbas, F., Ciloglu, S., Hasancebi, O., Grimmelsman, K., and Aktan, A. (2007). Limitations in structural identification of large constructed structures. *Journal of Structural Engineering*, 133(8):1051–1066.

Culler, D. and Hong, W. (2004). Wireless sensor networks. *Communications of the ACM*, 47(6):32–33.

Deng, J. and Gu, D. (2005). Radial basis function network approach to model the implicit performance function for raliability analysis. In *The Eighth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*. Civil-Comp Press. CDROM.

Guratzsch, R. and Mahadevan, S. (2006). Sensor placement design for shm under uncertainty. In *Third European Workshop on Structural Health Monitoring*, Granada, Spain.

Hua, X., Ni, Y., Ko, J., and Wong, K. (2007). Modeling of temperature-frequency correlation using combined principal component analysis and support vector regression technique. *Journal of Computing in Civil Engineering*, 21(2):122–135.

Jolliffe, I. (2002). *Principal Component Analysis*. Springer.

Li, D., Li, H., and Fritzen, C. (2006). On the physical significance of the norm based sensor placement method. In *Proceedings of the Third European Workshop on Structural Health Monitoring*, pages 1135–1143. DEStech publications, Inc.

Ljung, L. (1999). *System Identification - Theory For the User*. Prentice Hall.

Meo, M. and Zumpano, G. (2005). On the optimal sensor placement techniques for a bridge structure. *Engineering Structures*, 27:1288–1497.

Mullen, T., V., A., and D.L., H. (2006). Customer-driven sensor management. *IEEE Intelligent Systems*, 21(2):41–49.

Pal, S. and Mitra, P. (2004). *Pattern Recognition Algorithms for Data Mining*. CRC Press.

Papadimitriou, C., Beck, J., and Au, S. (2000). Entropy-based optimal sensor location for structural model updating. *Journal of Vibration and Control*, 6.

Parker, D., Frazier, W., Rinehart, H., and Cuevas, P. (2006). Experimental validation of optimal sensor placement algorithms for structural health monitoring. In *Proceedings of the Third European Workshop on Structural Health Monitoring*, pages 1144–1150. DEStech publications, Inc.

Raphael, B. and Smith, I. (2003). A direct stochastic algorithm for global search. *Journal of Applied Mathematics and Computation*, 146(2-3):729–758.

Robert-Nicoud, Y., Raphael, B., Burdet, O., and Smith, I. (2005a). Model identification of bridges using measurement data. *Computer-Aided Civil and Infrastructure Engineering*, 20(2):118–131.

Robert-Nicoud, Y., Raphael, B., and Smith, I. (2000). Decision support through multiple models and probabilistic search. In *Proceedings of Construction Information Technology*, pages 765–779.

Robert-Nicoud, Y., Raphael, B., and Smith, I. (2005b). Configuration of measurement systems using shannon's entropy function. *Computers and structures*, 83(8-9):599–612.

Robert-Nicoud, Y., Raphael, B., and Smith, I. (2005c). System identification through model composition and stochastic search. *Journal of Computing in Civil Engineering*, 19(3):239–247.

Saitta, S., Raphael, B., and Smith, I. (2005). Data mining techniques for improving the reliability of system identification. *Advanced Engineering Informatics*, 19(4):289–298.

Saitta, S., Raphael, B., and Smith, I. (2006). Rational design of measurement systems using information science. In *Proceedings of $IABSE$ Conference in Budapest*, volume $IABSE$ Report 92, page 118:119.

Saitta, S., Raphael, B., and Smith, I. (2007). A new bounded index for clustering. In Perner, P., editor, *Machine Learning and Data Mining in Pattern Recognition*, LNAI 4571, pages 174–187. Springer Verlag.

Salvo, A. (2006). Ponts de robert maillart. Technical report, EPFL-MCS, Lausanne, Switzerland.

Sanayei, M., Imbaro, G., McClain, J., and Brown, L. (1997). Structural model updating using experimental static measurements. *Journal of Structural Engineering*, 123(6):792–798.

SanJuan, E. and Ibekwe-SanJuan, F. (2006). Text mining without document context. *Inf. Process. Manage.*, 42(6):1532–1552.

Schulte, R., Bohle, K., Fritzen, C., and Schuhmacher, G. (2006). Optimal sensor placement for damage identification - an efficient forward-backward selection algorithm. In *Proceedings of the Third European Workshop on Structural Health Monitoring*, pages 1151–1159. DEStech publications, Inc.

Shannon, C. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press.

Shirazi Kia, S., Noroozi, S., Carse, B., and Vinney, J. (2005). Application of data mining techniques in predicting the behaviour of composite joints. In *The Eighth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*. Civil-Comp Press. CDROM.

Smith, I. and Saitta, S. (2007). Improving knowledge of structural system behavior through multiple models. *accepted for publication in Journal of Structural Engineering*.

Soibelman, L. and Kim, H. (2002). Data preparation process for construction knowledge generation through knowledge discovery in databases. *Journal of Computing in Civil Engineering*, 16(1):39–48.

Tan, P.-N., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Addison Wesley.

Tang, H., Xue, S., and Sato, T. (2007). H∞ filtering in neural network training and pruning with application to system identification. *Journal of Computing in Civil Engineering*, 21(1):47–58.

Webb, A. (2002). *Statistical Pattern Recognition*. Wiley.

Witten, I. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers.

Worden, K. and Burrows, A. (2001). Optimal sensor placement for fault detection. *Engineering Structures*, 23:885–901.

Xiong, N. and Svensson, P. (2002). Multi-sensor management for information fusion: issues and approaches. *Information Fusion*, 3(2):163–186.

Yin, J. and Yang, Q. (2005). Integrating hidden markov models and spectral analysis for sensory time series clustering. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 506–513, Washington, DC, USA. IEEE Computer Society.