Reliable spurious mode rejection using self learning algorithms

Ivan Goethals, Bart Vanluyten, Bart De Moor,
K.U.Leuven, Department of Electrical Engineering,
Kasteelpark Arenberg 10, B-3001, Leuven, Belgium,
e-mail: {**ivan.goethals/bart.vanluyten/bart.demoor**}**@esat.kuleuven.ac.be**.

Abstract

In this paper, we introduce a new technique for the separation of physical and spurious modes based on an initial clustering in frequency-damping space, followed by a self-learning classification algorithm. For the classification, Least Squares Support Vector Machines are used, a Least Squares version of the theory of Support Vector Machines which maps the classification problem to a high-dimensional feature space were the data points are linearly separable.

1 Introduction

Separation of physical and spurious modes is a long-standing problem in the modal analysis community [10]. It is widely known that in order to extract all relevant information from ambiently or directly excited vibrating structures, a high model order should be chosen during the identification process. However, high model orders will almost inevitably lead to the introduction of so-called spurious modes into the model, i.e., modes which are found in the model but have little to none physical relevance.

Various techniques have been suggested over the last few years to remove such spurious modes from a model. The most popular approach is without any doubt the creation of a so-called stabilization diagram [10], which is typically generated from a set of identified models with increasing model order. Modes which appear in most of these models with consistent frequency and damping are considered physical. Modes which only appear in some models are considered spurious.

Alternatively, in recent years, techniques have been proposed to remove spurious modes without generation of a stabilization diagrams. Examples include the analysis of pole/zero cancellations [15] and various methods to assess the energy content of vibrational modes [1, 2]. Although these methods have been shown to be quite succesful in many cases, the use of stabilization diagrams is still considered by many to be the most reliable technique for mode classification. A major disadvantage of stabilization diagrams, however, is that they involve a large amount of user interaction. Hence, analysis of stabilization diagram is a costly and time-consuming process, and certainly unsuited for online applications, such as the monitoring of vibrating airplanes in flight [3]. In recent years, automatic procedures for the analysis of stabilization diagrams have been proposed which to a large extend try to mimic the decision making process of a human operator. Examples include the use of simple heuristic rules [7], fuzzy logics and clustering. A major drawback in all these algorithms, however, is the fact that many thresholds and parameters need to be specified in advance.

In this paper, we introduce an automatic interpretation algorithm for stabilization diagrams which is based on a simple clustering technique followed by a self-learning classification algorithm based on the theory of Least Squares Support Vector Machines (LS-SVMs) [8, 9]. Due to the self-learning nature of the algorithm, all thresholds and parameters are automatically tuned to the kind of measurement data that is available. Hence, once the LS-SVMs are trained, the proposed method can be considered as fully automatic.

The outline of this paper is as follows: In Section 2, the main aspects of the algorithm are described and a

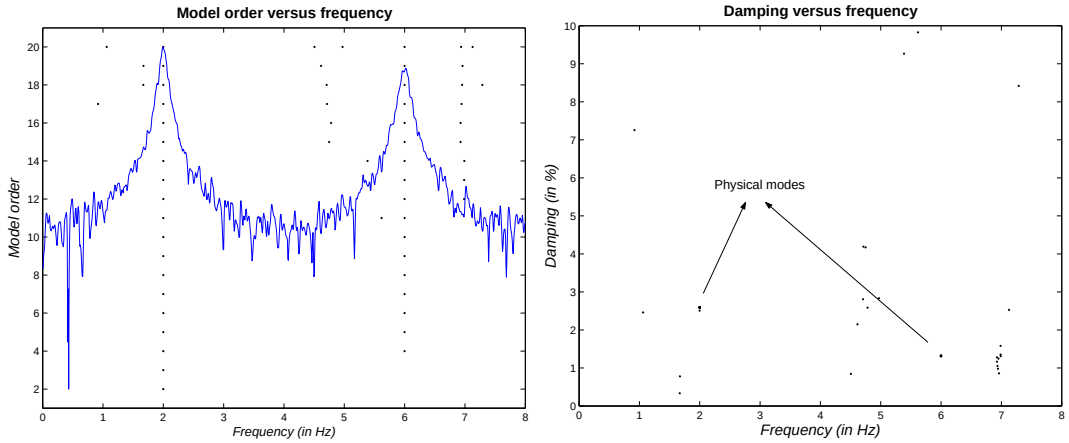

Figure 1: Stabilization diagrams of a simple system with modes at 2 Hz and 6 Hz. The picture at the left displays a classical order vs. frequency diagram. The picture at the right a damping vs. frequency diagram. Note that the physical modes form two dense clusters in the picture at the right.

concrete algorithm for the initial clustering step is proposed. In Section 3, LS-SVMs are introduced and used for the classification of the clusters obtained in Section 2. In Section 4, finally, the performance of the new technique is analyzed by means of a couple of examples.

2 Main building blocks of the algorithm

2.1 A stabilization diagram as a damping vs. frequency plot

Assume discrete input-output measurements $\{u_k, y_k\}, k = 0, \ldots, N-1$, with $u_k \in \mathbb{R}^{n_u}, y_k \in \mathbb{R}^{n_y}$, are available and a set of linear models

$$\mathcal{M}_i,\ i = n_l, \ldots, n_h,\ n_h > n_l > 0 \tag{1}$$

has been obtained using system identification on the input-output measurements with orders $i = n_l, \ldots, n_h$. Assume further, that from each model $\mathcal{M}_i$ a set of modes $p_{i,j}$ are obtained characterized by a frequency, damping value and corresponding modal vector, denoted as $f_{i,j} \in \mathbb{R}^+, d_{i,j} \in \mathbb{R}$ and $v_{i,j} \in \mathbb{R}^{n_y}$ respectively, with j ranging from 1 to the number of available modes in $\mathcal{M}_i$, which we shall conveniently denote as $n_{\mathcal{M}_i}$.

Two common ways to depict these modes are displayed in Figure 1 for a simple system involving two physical modes at frequencies $2Hz$ and $6Hz$. The picture at the left displays a classical stabilization diagram with on its vertical axis the model orders, and on its horizontal axis the frequencies $f_{i,j}, j = 1, \ldots, n_{\mathcal{M}_i}$ that are obtained for each model order i. The picture at the right displays the same information in a frequency-damping plot, with $d_{i,j}$ plotted versus $f_{i,j}$ for each i and j such that $n_l <= i <= n_h$ and $1 <= j <= n_{\mathcal{M}_i}$.

From the picture at the right, it is clear that the estimated modes corresponding to a physical mode form dense clusters which could in principle be retrieved using classical clustering approaches such as the K-means algorithm [4]. However, most of these algorithms assume that the number of clusters to find is well known in advance, which is most often not the case. In this paper, we therefore propose to overestimate the number of clusters and use a clustering algorithm which does not rely on an initial estimate of the number of clusters to find. In a second step a self-learning classification algorithm trained on a set of cluster properties such as the number of members, and the variances in their frequencies, dampings and modal vectors can be used to distinguish between clusters representing physical modes and those representing spurious modes. Hence, the main purpose of the clustering algorithm will be to perform some initial preprocessing on the

stabilization diagram. The final decision on the nature of a cluster and the modes within is taken by the learning algorithm. We discuss a possible clustering algorithm in the following subsection.

2.2 Clustering based on the distance between individual datapoints

Probably the most straightforward way to cluster a set of data points, where the data points are in this case the set of modes $p_{i,j}$, is to start with one initial data point, and append all data points which are closer to this point than a certain threshold τ. This procedure is repeated for the newly acquired data points to include all points which are mutually connected by a finite number of hops with distance smaller than τ. Once a first cluster is constructed in this way, a data point which does not belong to the first cluster is chosen and the procedure is repeated to form the second cluster. The procedure is continued until all data points belong to exactly one cluster.

For the distance measure between two datapoints, in our case two modes p_{i_1,j_1} and p_{i_2,j_2}, we use:

$$\text{dist}(p_{i_1,j_1}, p_{i_2,j_2}) = \sqrt{\frac{1}{\delta_f}(f_{i_1,j_1} - f_{i_2,j_2})^2 + \frac{1}{\delta_d}(d_{i_1,j_1} - d_{i_2,j_2})^2}, \tag{2}$$

where δ_f and δ_d are two tuneable parameters which can be seen as a measure of what is considered 'far' in terms of distances between frequencies and dampings. Note the absence of the modal vectors in the distance function. The modal vectors will be used in a later stage. As a threshold for the clustering algorithm we choose $\tau = 1$. The most apparent disadvantages of this approach are:

- The clustering algorithm relies on the specification of two tuneable parameters δ_f and δ_d. A reliable technique to extract δ_f and δ_d from training data is given in Section 3.
- The parameters δ_f and δ_d are preferably chosen rather high to avoid breaking up groups of modes which correspond to one physical mode (which would result in a physical mode being classified as a spurious one, which is in general considered worse than the opposite). However, because of this policy, a set of physical modes which are very densely packed can accidentally be grouped in one single cluster, even if the modes are fundamentally different. A dedicated procedure to deal with this situation is given in the next subsection.

As biggest advantages we have:

- A very fast implementation of this algorithm can be made as the decision process is fairly simple.
- The algorithm is non-recursive and there is no dependency on initial datapoints as is for instance the case in K-means algorithms.

Results of the described clustering algorithm are shown on measurements obtained from a steel beam that is part of a cellular phone network [5, 6]. 9 accelerometers were placed on the beam and their signals used as outputs in a stochastic subspace identification algorithm [11, 12, 13]. As the beam was sufficiently excited by the wind, no further artificial excitation was used. A stabilization diagram of the beam is shown in Figure 2 together with the results of an initial clustering algorithm with $\delta_f = 0.04$ Hz and $\delta_d = 0.5\%$.

2.3 Dealing with closely spaced modes

A well known problem in the field of modal analysis is that of closely spaced modes. As the identification order increases, modes in the stabilization diagram tend to split, and this for spurious as well as for physical modes. Especially in the latter case it is often difficult to determine whether the splitting of a physical mode corresponds to a real double mode, or is just an artefact of the artificially high model orders involved. An

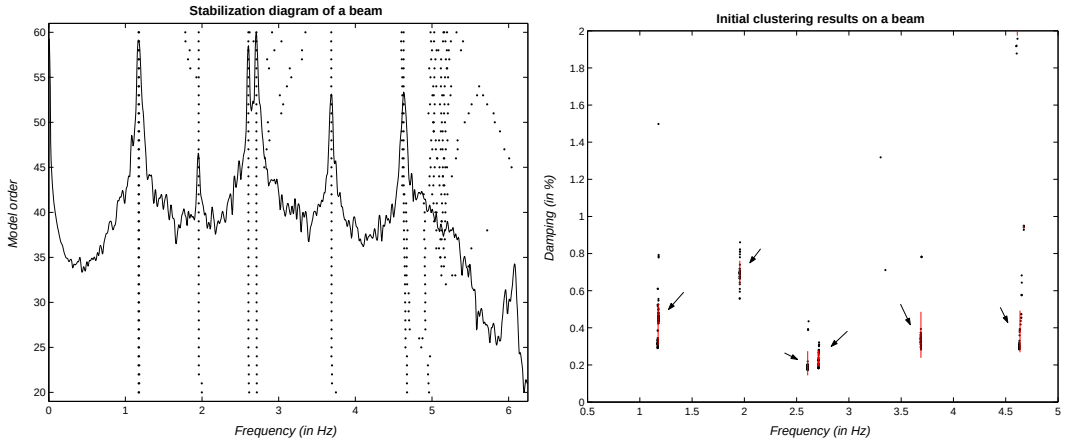

Figure 2: Stabilization diagrams (left) and initial clustering results (right) on a steel beam. Clusters are indicated with arrows and were obtained setting $\delta_f = 0.04$ Hz and $\delta_d = 0.5\%$. Their centers and standard deviations in frequency and damping are displayed using red crosses.

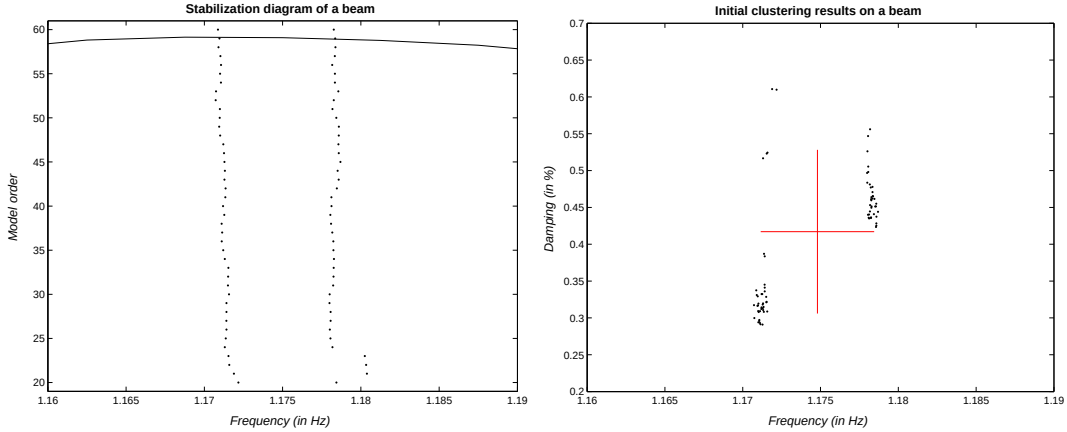

Figure 3: Closer look at the area around 1.175 Hz of the stabilization diagrams (left) and initial clustering results (right) displayed in Figure 2. Clusters are indicated using red crosses. Note how a double mode is represented by one single cluster.

example of such a double mode is displayed in Figure 3 which gives a closer look at the frequency range around 1.175 Hz in Figure 2. Note how the double mode was identified as a single cluster, although it is well known to be a real double mode (see the experiments in [5, 6]).

The presence of two or more different modes in a cluster can easily be detected from the fact that for most model orders, two or more poles identified at this model order are found within the cluster. As an algorithmic rule we propose for a given cluster, to count for each model order n such that $n_l \leq n \leq n_h$ the number of modes $p_{i,j}$ present in this cluster so that $i = n$. The rounded average of this number over all model orders in the stabilization diagram is taken as a measure for the number of modes in the cluster.

The modes themselves are separated by looking at their MAC-values, which is a measure for the angle between their respective modal vectors. For two modal vectors v_{i_1,j_1} and v_{i_2,j_2} we have:

$$\text{MAC}(v_{i_1,j_1}, v_{i_2,j_2}) = \frac{|v_{i_1,j_1}^T v_{i_2,j_2}|}{\|v_{i_1,j_1}\| \|v_{i_2,j_2}\|}$$

Using this as a distance measure, a K-means clustering algorithm with the above mentioned number of modes

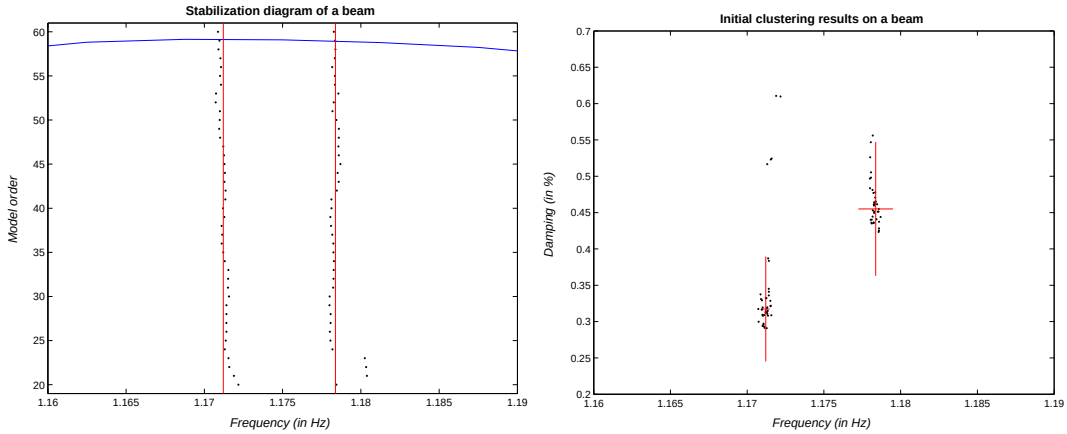

Figure 4: Closer look at the area around 1.175 Hz of the stabilization diagrams (left) and initial clustering results (right) displayed in Figure 2, after a K-means algorithm was applied to separate the double mode using a MAC-distance measure. Clusters are indicated using red lines and crosses

given is used to redistribute the poles in the original cluster over several new smaller ones.

Alternatively, and especially if the number of outputs is low, so that the extra information obtained in the modal vectors is limited, the K-means algorithm can be run using the distance measure (2) based on frequencies and dampings.

Applying the algorithm as explained above, the mode around 1.175 Hz is correctly split, as seen in Figure 4.

3 Learning and generalization

3.1 The need for learning techniques

In Section 2, an algorithm was introduced to divide the set of modes, obtained from identification on a vibrating structure, in a finite number of clusters. In principle, one could now continue by ordering these clusters according to decreasing number of members, set a threshold (based on a gap in the ordered member-count), and declare modes inside clusters with a high number of members as physical, and the other ones as spurious. Several objections to this strategy can however be noted:

1. The outcome of the algorithm might heavily depend on the choice of the tuning parameters δ_f and δ_d in (2).

2. It is not guaranteed that a clear gap in the ordered member-count will appear. In this case, one needs to define a threshold manually, which would constitute yet another tuning parameter.

3. A member-count as such is not a reliable method to distinguish between spurious and physical modes. It is widely known that in stabilization diagrams, spurious modes can show stabilizing behavior as well, to a certain extend, which could lead to them being classified as physical.

As an example of the stabilization of spurious modes, Figure 5 displays a more detailed view on the region around 4.6 Hz in Figure 2, for the measurements on a transmitter beam. Only one mode around 4.6 Hz is known to be physical, namely the second one from the left. Yet, several other spurious modes are seen to form clusters with a considerable amount of members. However, one can clearly notice that the physical

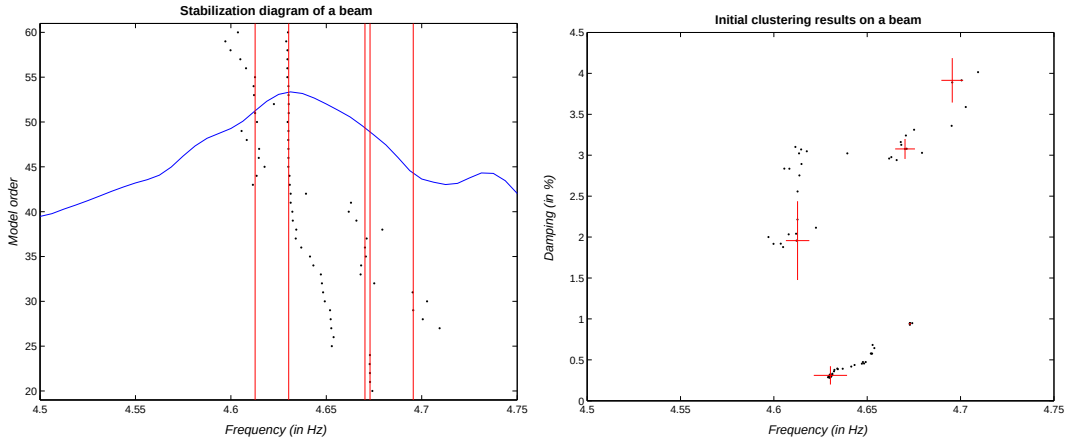

Figure 5: Closer look at the area around 4.6 Hz of the stabilization diagrams (left) and initial clustering results (right) displayed in Figure 2, after a K-means algorithm was applied to separate double mode using a MAC-distance measure. Clusters are indicated using red lines and crosses

mode has a much better stabilization for the orders 40 to 60 than the spurious ones. These properties will be exploited in a learning classification algorithm to make a distinction between physical and spurious modes. First of all, learning techniques will allow us to make a good choice for the tuning parameters δ_f and δ_d from a so-called training data set. Secondly, learning classification algorithms can be trained on several properties of clusters, such as the variance in frequencies, dampings and modal vectors of the modes within (which are a measure for the quality of the stabilization). This will allow for a much better discrimination between physical and spurious modes than what would be obtained using a member-count only.

3.2 Construction of a training dataset

Learning algorithms need training data. In this subsection, we briefly discuss the generation of such data. Assume a vibrating structure is given with a set of known physical modes. The frequencies, dampings and modal vectors of these modes will be denoted as follows:

- Frequencies: $f_k^0, k = 1, \ldots, n_{\mathcal{M}}$
- Damping values: $d_k^0, k = 1, \ldots, n_{\mathcal{M}}$
- Modal vectors: $v_k^0, k = 1, \ldots, n_{\mathcal{M}}$

where $\mathcal{M}$ is used to denote the true system, and $n_{\mathcal{M}}$ symbolises the total number of known physical modes in this system.

Suppose, based on a sequence of input-output or output-only data measured on the system described above, identified models $\mathcal{M}_i$ are obtained for $n_l \leq i \leq n_h$. We again denote the set of extracted frequencies, damping values and modal vectors as $f_{i,j}, d_{i,j}, v_{i,j}$, in accordance with Section 2.

The idea of a learning algorithm is now to learn to extract the true modal parameters from the set of estimated ones. In a typical industrial situation, the known physical modes would be extracted by a skilled engineer. In this case, training the learning algorithm is equivalent with teaching it to mimic the decision making process of the engineer. As outlined before, in a first step δ_f and δ_d will have to be extracted from the training data. In a second step, clustering will be performed on the extracted modal parameters. Finally, an LS-SVM classification algorithm will be trained to make a distinction between clusters containing physical modes and clusters containing spurious modes.

3.3 Extraction of δ_f and δ_d

The extraction of good choices for δ_f and δ_d from one or more training data sets can be done using a simple procedure. Given a physical mode with frequency f_k^0 and damping d_k^0, the following algorithm is used:

1. set t as the closest integer to $\frac{n_h - n_l}{2}$
2. find a set of $\{j_i\}, i = n_l, \ldots, n_h$ such that $j_i = \arg\min_x \|f_{i,x} - f_k^0\|$
3. Sort the obtained sequence $\{f_{i,j_i}\}, i = n_l, \ldots, n_h$ according to increasing distance to the target frequency f_k^0. Put the result in a vector $\mathcal{F}$.
4. Calculate mean ν and standard deviation σ of the first t elements in $\mathcal{F}$, assuming that their distribution is approximately Gaussian.
5. If $\mathcal{F}(t+1)$ is within a 3σ-bound of the Gaussian distribution calculated above, increase t with one and continue with step 4.
6. $\delta_f = 3\sigma$.

This procedure is repeated for all physical modes and all available training data sets, whereafter the maximal value over all obtained δ_f is chosen. A similar procedure can be used to make a choice for δ_d. Note that the so obtained choices for δ_f and δ_d will be rather conservative (given the wide 3σ bound). This, as mentioned earlier, mostly to avoid false negatives.

Another measure that could turn out to be usefull later is the expected deviation of estimated modal vectors from the true modal vector. Using the procedure described above, a parameter δ_v, giving a threshold for the MAC value between two modal vectors which are essentially describing the same physical mode, can easily be constructed. A typical choice in the modal analysis community is to set δ_v to 0.9 or 0.95.

3.4 Parametrization of the clusters and introduction to LS-SVMs

With the choices of δ_f,δ_d, and δ_v obtained in the former subsection, the clustering algorithm outlined in Section 2 can be started, resulting in a set of clusters which can all be considered as physical mode candidates. However, as noted before, due to the fact that the number of clusters will in general be higher than the number of physical modes, many of the obtained clusters will have to be removed in a second step. In order to do this, for each cluster, a set of parameters indicating the relevance of the cluster are constructed. For the examples included in this paper, we chose the following set of parameters:

1. Number of modes in the cluster
2. Quality of stabilization: A mode p_{i_1,j_1} is considered stable if a mode p_{i_1-1,j_2} is found in the same cluster such that

   $$|f_{i_1,j_1} - f_{i_1-1,j_2}| < \delta_f,\ |d_{i_1,j_1} - d_{i_1-1,j_2}| < \delta_d,\ MAC(v_{i_1,j_1}, v_{i_1-1,j_2}) > \delta_v$$

   For each stable mode, a quality parameter q_i is calculated as follows:

   $$q_i = G_{\delta_f}(f_{i_1,j_1} - f_{i_1-1,j_2}) + G_{\delta_d}(d_{i_1,j_1} - d_{i_1-1,j_2}) + G_{1-\delta_v}(|1 - MAC(v_{i_1,j_1}, v_{i_1-1,j_2})|),$$

   with G_x the Gaussian density function with zero mean and standard deviation x. The sum of all quality parameters is used as a measure for the quality of the stabilization.
3. Standard deviation of all the frequencies

4. Standard deviation of all the damping values
5. Standard deviation of all the frequencies devided by the average frequency
6. Standard deviation of all the damping values devided by the average damping value

The idea behind a classification algorithm is now to construct a mapping $f : \mathbb{R}^6 \to \{-1, 1\}$ classifying each cluster as physical (1) or spurious (-1). In a self-learning classification algorithm, this mapping f is derived from a set of so-called training data points for which the classes are known.

Many self-learning classification algorithms exist, of which various flavours of cellular neural networks are probably the most widely known. However, in this paper, we chose an algorithm based on the theory of Least Squares Support Vector Machines [8, 9], a least squares implementation of the theory of Support Vector Machines introduced by Vapnik [14]. The main advantages of working with SVMs can be summarized as follows:

1. Classifications for test data points are expressed as a function of those for the training data points (the co-called primal dual framework). Hence, by construction, SVM classification algorithms zoom in on those regions of the input-space were the training data is located. Assuming that training and test data are of the same nature, this leads to greatly improved behavior over more classical classification algorithms which spread their effort over the entire input space.
2. SVM techniques are more robust against the so-called curse of dimensionality. From the definition of the 6 cluster parameters above, we might expect a lot of correlation between them. Hence, all data points can be expected to lie on a manifold in $\mathbb{R}^6$ of which the dimension might in fact be lower than 6. Since SVM algorithms zoom in on those regions of the input-space were the training-data is located, the effective dimension of the search space is often much lower for SVM algorithms than for most classical classification algorithms, resulting in vastly superior performance.

A more rigorous introduction to the LS-SVM classification algorithm is given in the following subsection. In Subsection 3.6, we apply the LS-SVM classification algorithm to the cluster analysis problem.

3.5 Derivation of the LS-SVM classification algorithm

The idea behind the LS-SVM classification algorithm, is to map the set of data points $x_t \in \mathbb{R}^6, t = 1, \ldots, T$, to a high, possibly infinite dimensional, space where the two classes ($y_t = -1$ and $y_t = 1$) become linearly separable. This mapping is performed using the so-called feature map:

$$\varphi : \mathbb{R}^6 \to \mathbb{R}^{n_f}$$

which is only implicitly defined through its inner-product, called the kernel:

$$K(x_k, x_l) = \varphi(x_k)^T \varphi(x_l). \tag{3}$$

Many choices for the kernel exist of which a linear kernel and a Radial Basis Function kernel are the most widely known.

The LS-SVM training algorithm can be summarized as follows:

$$\min_{w,b,e} J(w, b, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{t=1}^{T} e_t^2 \tag{4}$$

$$\text{such that} \qquad y_t[w^T \varphi(x_t) + b] = 1 - e_t, \; t = 1, \ldots, T \tag{5}$$

where the constraint can also be written as $y_t = \text{sign}[w^T\varphi(x) + b]$. The Lagrangian for the problem (4) is

$$\mathcal{L}(w, b, e; \alpha) = J(w, b, e) - \sum_{t=1}^{T} \alpha_t \{y_t[w^T\varphi(x_t) + b] - 1 + e_t\}$$

where the α_t values are the Lagrange multipliers. The conditions for optimality are:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow \quad w = \sum_{t=1}^{T} \alpha_t y_t \varphi(x_t), \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow \quad \sum_{t=1}^{T} \alpha_t y_t = 0, \\ \frac{\partial \mathcal{L}}{\partial e_t} = 0 & \rightarrow \quad \alpha_t = \gamma e_t, \;\; t = 1, \ldots, T, \\ \frac{\partial \mathcal{L}}{\partial \alpha_t} = 0 & \rightarrow \quad y_t[w^T\varphi(x_t) + b] - 1 + e_t = 0, \;\; t = 1, \ldots, T. \end{cases}$$

Defining $Z = \begin{bmatrix} y_1\varphi(x_1) & y_2\varphi(x_2) & \ldots & y_T\varphi(x_T) \end{bmatrix}^T$, $y = \begin{bmatrix} y_1 & y_2 & \ldots & y_T \end{bmatrix}^T$, $1_v = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}^T$, $e = \begin{bmatrix} e_1 & e_2 & \ldots & e_T \end{bmatrix}^T$, $\alpha = \begin{bmatrix} \alpha_1, \alpha_2, \ldots, \alpha_T \end{bmatrix}^T$ and eliminating w and e, one obtains estimates for α and b from the following Karush-Kuhn-Tucker (KKT) system:

$$\left[\begin{array}{c|c} 0 & y^T \\ \hline y & \Omega + I/\gamma \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1_v \end{array}\right], \tag{6}$$

where $\Omega = Z^T Z$ which can be calculated using the definition of the kernel 3. The classifier, finally, takes the form:

$$y(x) = \text{sign}\left[\sum_{t=1}^{T} \alpha_t y_t K(x, x_t) + b\right]. \tag{7}$$

The Kernel which will be used for the examples in this paper is the Radial Basis Function (RBF) kernel:

$$K(x_k, x_l) = \varphi(x_k)^T\varphi(x_l) = \exp(-\|x_k - x_l\|_2^2/\sigma^2).$$

3.6 Applying the LS-SVM classification algorithm to the spurious mode rejection problem

Given a set of clusters and their relevance parameters as defined in subsection (3.4) for a training data set, training input vectors $x_t \in \mathbb{R}^6$ and output vectors $y_t \in \mathbb{R}$ can be constructed. Hereby, we use the convention that $y_t = 1$ is used for a cluster containing physical modes and $y_t = -1$ is used for a cluster containing spurious modes. α and b are readily extracted from (6), whereby two hyper-parameters σ and γ are typically tuned using validation techniques such as 10-fold cross validation.

Once α and b are obtained, equation (7), the so-called generalization, can be used to assess the nature of any new cluster. The generalization performance of this type of classification algorithms is evaluated in the next Section.

4 Performance

4.1 A simulated example

40 output-only datasets sampled at 2π Hz were generated, each with 9 outputs, and 5 to 13 physical modes randomly distributed within the frequency range $0, \ldots, \pi$ Hz and damping range $0, \ldots, 5\%$. 10% of output noise was added.

All datasets were identified using stochastic subspace identification and stabilization diagrams were generated for orders 20 ranging to 60. The clustering algorithm as introduced in section 2 was applied and relevance parameter vectors $x_t \in \mathbb{R}^6$ generated for every cluster. A random selection of 496 physical and

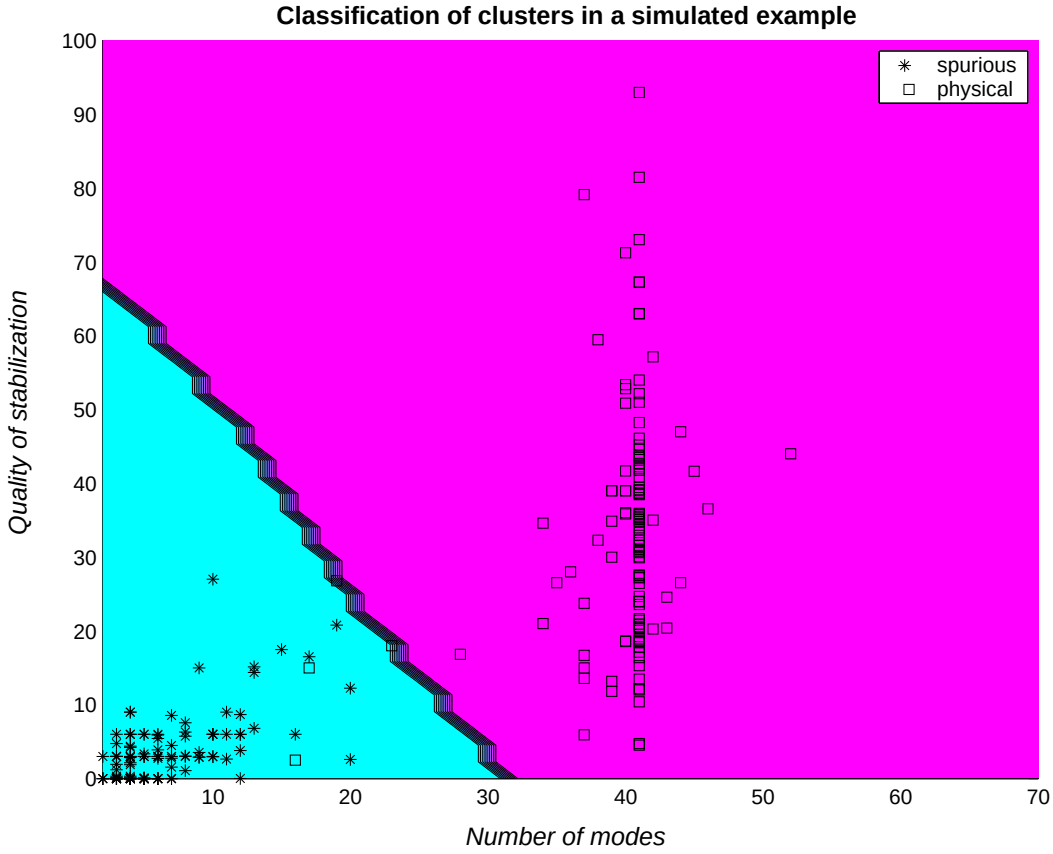

Figure 6: Classification of clusters based on their number of modes and a 'quality of stabilization'-measure. The dark line on the figure represents the boundary between the physical and the spurious class as set by the learning classification algorithm on simulated training data. The symbols denote the cluster properties for simulated test data. 3 clusters in the test set are incorrectly classfied by the learning algorithm.

spurious clusters was made, of whom half were used as training data, and half as test data to study the performance of the self learning algorithm introduced in Section 3.

In a first test, only two relevance parameter for every cluster were used by the classification algorithm, namely the number of poles and the quality of stabilization. This because working in two dimensions allows us to show some figures, something which is not possible when working with higher dimensions. Figure 6, for instance displays the results of a classification algorithm based on these two parameters, using the training data. The 'physical' and 'spurious' region as found by the classification algorithm are clearly visible in the figure. Scattered within these regions are symbols denoting the relevance parameters of the clusters in the test set. A total of 98.8% of clusters in the test set were correctly classified by the algorithm.

As can be seen in the figure, some clusters in the test set can not be attributed to the right class based on the two relevance parameters used above. When using all 6 parameters, on the other hand, the performance of the test set increases to 100% on the test set.

Note from Figure 6 that in this simulated example, a mere distinction based on the number of modes in a cluster would already have performed well. As a more realistic example, in the next subsection, the learning algorithm is tested on data from a flight flutter test.

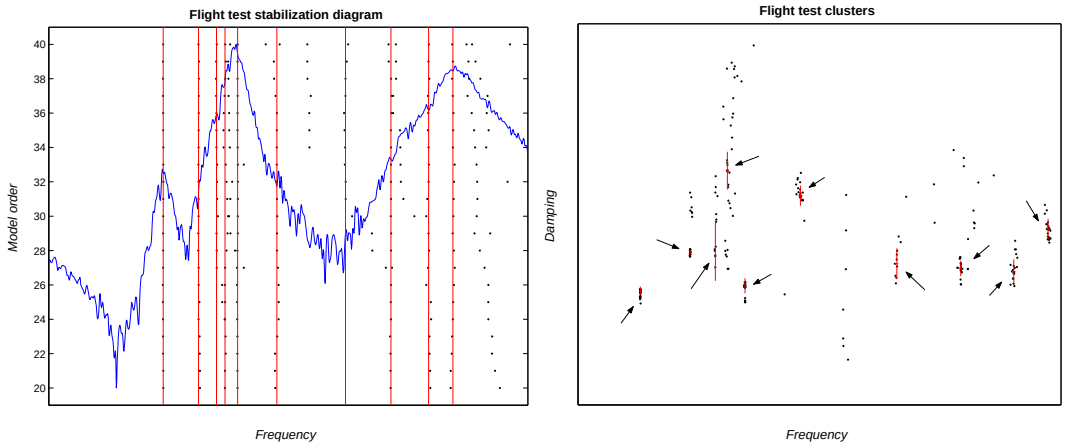

Figure 7: Stabilization diagram as an order vs. frequency and a damping vs. frequency plot for a fly by wire aircraft. Modes identified as physical following a clustering and self learning classification algorithm are denoted by lines (left) and crosses and arrows (right). The classification algorithm was trained using datasets of the same airplane in different flight conditions.

4.2 A real-life example

As a real-life example, 5 datasets of an airplane in flight were used. Each dataset corresponds to a different altitude and speed, and hence, a different linear model. The airplane was excited using a single input in a fly by wire system. Vibrations were recorded using 12 accelerometers. The first 4 datasets were used to train the classification algorithm. The fifth dataset was used as a test set. A stabilization diagram of the fifth dataset is displayed in Figure 7, with the modes identified as physical by the classification algorithm indicated using lines, crosses and arrows (see the caption with the figure). The frequency and damping scale on this figure were not displayed for reasons of confidence. However, the recovered physical modes correspond nicely to the ones specified by the engineers of the airplane (who's judgement was also used as input when training the classification algorithm), as will be noted by the reader upon looking at the stabilization diagram.

5 Conclusions

In this paper, we have introduced an automatic technique for the analysis of stabilization diagram. The technique employs an initial clustering algorithm, followed by a learning classification algorithm to assess the nature of the obtained clusters. Using this two step procedure many problems that are classicaly encountered when analyzing stabilization diagrams, such as the need to choose a set of thresholds or have prior knowledge on the number of modes to find, are avoided. The new algorithm was seen to perform well on simulated and real-life datasets.

Acknowledgments

Acknowledgements. This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven. It is supported by grants from several funding agencies and sources: Research Council KU Leuven: Concerted Research Action GOA-Mefi sto 666 (Mathematical Engineering), IDO (IOTA Oncology, Genetic networks), several PhD/postdoc & fellow grants; Flemish Government: Fund for Scientifi c Research Flanders (several PhD/postdoc grants, projects G.0407.02 (support vector machines), G.0256.97 (subspace), G.0115.01 (bio-i and microarrays), G.0240.99 (multilinear algebra), G.0197.02 (power islands), G.0499.04

(robust statistics), research communities ICCoS, ANMMM), AWI (Bil. Int. Collaboration Hungary/ Poland), IWT (Soft4s (soft-sensors), STWW-Genprom (gene promotor prediction), GBOU-McKnow (Knowledge management algorithms), Eureka-Impact (MPC-control), Eureka-FLiTE (flutter modeling), several PhD grants); Belgian Federal Government: DWTC (IUAP IV-02 (1996-2001) and IUAP V-10-29 (2002-2006) (2002-2006): Dynamical Systems and Control: Computation, Identification & Modelling), Program Sustainable Development PODO-II (CP/40: Sustainibility effects of Traffic Management Systems); Direct contract research: Verhaert, Electrabel, Elia, Data4s, IPCOS. JS is an associate professor and BDM is a full professor at K.U.Leuven Belgium, respectively.

References

[1] I. Goethals, B. De Moor, "Model reduction and energy analysis as a tool to detect spurious modes," *Proceedings of the International Conference on Noise and Vibration Engineering (ISMA)*, Leuven, Belgium, Jun. 2002.

[2] I. Goethals, B. De Moor, "Subspace identification combined with new mode selection techniques for modal analysis of an airplane," *Proceedings of the 13th IFAC symposium on system identification (SYSID 2003)*, Rotterdam, the Nederlands, Sep. 2003.

[3] I. Goethals, L. Mevel, A. Benveniste, B. De Moor, "Recursive output-only subspace identification for in-flight flutter monitoring," *Proceedings of the 22nd International Modal Analysis Conference (IMAC-XXII)*, Dearborn, Michigan, Jan. 2004.

[4] J. A. Hartigan, M.A. Wong, "A K-means clustering algorithm," Applied Statistics 28, 1979, pp. 100-108.

[5] B. Peeters, Steel mast website, http://www.kuleuven.ac.be/bwm/macec/mast.htm, March 1999.

[6] B. Peeters, *System Identification and Damage Detection in Civil Engineering*, PhD thesis, K.U.Leuven, Leuven, Belgium, 2000. Available on http://www.kuleuven.ac.be/bwm/peet00a.pdf.

[7] M. Scionti, J. Lanslots, I. Goethals, A. Vecchio, H. Van der Auweraer, B. Peeters, B. De Moor, "Tools to improve detection of structural changes from in-flight flutter data,", *Proceedings of the Eigth International Conference on Recent Advances in Structural Dynamics (ISVR)*, Southampton, UK, Jul. 2003.

[8] J.A.K. Suykens, J. Vandewalle. "Least squares support vector machine classifiers," *Neural Processing Letters*, 9(3):293–300, 1999.

[9] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle. "Least Squares Support Vector Machines," *World Scientific*, 2002.

[10] H. Van der Auweraer, "Structural Dynamics Modeling using Modal Analysis: Applications, Trends and Challenges," *Proceedings of the 2001 IEEE Instrumentation and Measurement Technology Conference*, Budapest, Hungary, 2001.

[11] P. Van Overschee, B. De Moor, "Subspace algorithms for the stochastic identification problem," *Automatica*, vol. 29, no. 3, Mar. 1993, pp. 649-660.

[12] P. Van Overschee, B. De Moor, "A unifying theorem for three subspace system identification algorithms", *Automatica*, Special Issue on Trends in System Identification, vol. 31, no. 12, Dec. 1995, pp. 1853-1864.

[13] P. Van Overschee, B. De Moor, *Subspace Identification for Linear Systems, Theory, Implementation, Application*, Kluwer Academic Publishers, 1996.

[14] V.N. Vapnik. *Statistical Learning Theory*, John Wiley and Sons, 1998.

[15] P. Verboven, E. Parloo, P. Guillaume, M. Van Overmeire, “Autonomous Structural Health monitoring - part 1: Modal Parameter Estimation and Tracking,” *Mechanical Systems and Signal Processing*, 16, no. 4, 2002, pp. 637-657.