Using support vector machines for the computationally efficient identification of acceptable design parameters in computer-aided engineering applications

Michael E. Cholette [a,*], Pietro Borghesani [a], Egidio Di Gialleonardo [b], Francesco Braghin [b]

[a] *Science and Engineering Faculty, Queensland University of Technology, Brisbane, Queensland Australia*
[b] *Dipartimento di Meccanica, Politecnico di Milano, Milan, Italy*

This paper addresses the problem of estimating continuous boundaries between acceptable and unac- ceptable engineering design parameters in complex engineering applications. In particular, a procedure is proposed to reduce the computational cost of finding and representing the boundary. The proposed methodology combines a low-discrepancy sequence (Sobol) and a support vector machine (SVM) in an active learning procedure able to efficiently and accurately estimate the boundary surface. The paper describes the approach and methodological choices resulting in the desired level of boundary surface refinement and the new algorithm is applied to both two highly-nonlinear test functions and a real-world train stability design problem. It is expected that the new method will provide designers with a tool for the evaluation of the acceptability of designs, particularly for engineering systems whose behaviour can only be determined through complex simulations.

Keywords:
Support vector machines
Zero finding
Design criteria evaluation
Train dynamics
Active learning

1. Introduction

Any engineering design process deals with the evaluation of a possible set of design solutions, each obtained as the combination of specific values of a set of *design parameters*. This evaluation is usually performed comparing a *design score* with a predetermined *limit score*, and often defining a safety factor to ensure compliance despite the uncertainty of the actual manufacturing process and operating conditions. Traditional applications of this methodology, such as the verification of static and fatigue criteria for beams and shafts, are based on simple analytic relationships between design and operation parameters (e.g. dimensions, material properties and loading) and design score (e.g. maximum von Mises equivalent tensile stress).

With a push for innovative design solutions, the designer is often following a time- and resource-intensive trial-and-error design process, including the proposal of creative new solutions and a long series of design score evaluations. In complex engineering systems (i.e. multiple design parameters and nonlinear functional relationships) the number of possible suitable solutions and the cost of computing each solution's design score increases dramatically. If the designer is unaware of the likely sets of design parameter combinations resulting in acceptable design scores, this process may become cumbersome.

Providing an estimate of the *limit-score boundary* separating acceptable and unacceptable design parameters (in terms of design score) would enable the designer to focus on areas of the solution space sufficiently far from the limit boundary. Moreover, once the solution is chosen, the assessment of the distance from the boundary would also allow evaluating its "safety margin", i.e. robustness to variations of design parameters.

This study aims at obtaining an estimate of the limit-score boundary using a limited number of computationally expensive design-score evaluations. The boundary search consists of finding all the combinations of design parameters in the space resulting in a fixed value of the design score (limit score). Without loss of generality, this task is equivalent to a zero-finding problem for an unknown multi-input single-output function.

To achieve this result, this paper proposes to (i) represent the zero finding task as a classification problem and estimate the level set as a decision boundary; (ii) ensure sufficient exploration of the design parameter space by using a space filling algorithm to select candidate samples; (iii) reduce the number of design-score evaluations needed to estimate the boundary by iteratively evaluating only samples likely close to the boundary.

* Corresponding author.
E-mail addresses: michael.cholette@qut.edu.au (M.E. Cholette), p.borghesani@qut.edu.au (P. Borghesani), egidio.digialleonardo@polimi.it (E.D. Gialleonardo), francesco.braghin@polimi.it (F. Braghin).

Published Journal Article available at: http://dx.doi.org/10.1016/j.eswa.2017.03.050

The proposed iterative procedure is based on:

1. a Sobol sequence for the generation of a space-uniform set of candidate design parameter combinations;
2. a Support Vector Machine (SVM), applied to a small subset of the candidate combinations from 1 to obtain a decision boundary used as a proxy (or surrogate) of the limit-score boundary;
3. the use of this computationally inexpensive SVM-surrogate to identify which of the design parameter combination of 1 is most likely close to the boundary (without running actual simulations);
4. the iterative (points 3 and 4) generation of refined SVM-based representations of the boundary by running targeted numerical simulations only for samples identified in 3.

This procedure allows producing a refined representation of the boundary without the need for an exhaustive evaluation of the design score throughout the multidimensional space of design parameters. This is done by using at each step the partial information on the location of the boundary to select only few and high-value design parameter combinations to be evaluated. The proposed methodology thus uses an SVM surrogate (or "meta-model") to identify an explicit representation of the boundary. While other meta-modelling approaches are available, e.g. Artificial Neural Neworks (Sundar & Shields, 2016), Response Surface methods (Goswami, Ghosh, & Chakraborty, 2016; Roussouly, Petitjean, & Salaun, 2012), and Kriging Methods (Sun, Wang, Li, & Tong, 2017), SVMs have important advantages for pursuing boundary re- finement (point 4) (Kremer, Steenstrup Pedersen, & Igel, 2014) and are therefore selected here.

The proposed estimation-and-refinement strategy (points 3 and 4) belongs to the framework of *active learning*, a branch of machine learning where training includes iteratively querying new training data points. Active learning is particularly beneficial when the labelling of new training points comes at a high computational cost. The idea is that by intelligently querying points, one can achieve a high-accuracy classifier with only a limited subset of training samples (Settles, 2009). Given a classifier trained on a (small) subset of the available data, the key question in active learning is how to select the most informative unlabelled samples. The most popular strategy is to select samples in regions where the classifier is the least confident, called *uncertainty sampling* or *simple query* strategies (Guyon, Cawley, Dror, & Lemaire, 2011; Ho, Tsai, & Lin, 2011; Kremer et al., 2014; Lewis & Gale, 1994; Settles, 2009). However, it is well known that this sampling approach can be problematic: it can over-emphasize regions of the feature space that are not representative of the data distribution and it assumes that the classifier accurately labels points that are far from the estimated decision boundary (Kremer et al., 2014). In other words, uncertainty sampling alone tends to stress "exploitation" while sacrificing "exploration" of new feature space regions (Guyon et al., 2011).

In active learning SVMs are commonly chosen as the algorithms to perform classification owing to their ability to clearly identify samples near the decision boundary (Kremer et al., 2014). SVMs have been traditionally used to classify experimental data in a wide variety of applications, including condition monitoring (Kim, Tan, Mathew, & Choi, 2012; Samanta, 2003; Widodo & Yang, 2007), face recognition (Huang, Shao, & Wechsler, 1998), medical diag- nosis (Chen, Yang, Liu, & Liu, 2011; Musselman & Djurdjanovic, 2012), and pattern recognition in control charts (Hachicha & Ghor- bel, 2012; Lu, Shao, & Li, 2011). In the traditional active learning applications, as for condition monitoring and diagnostics, the main focus has been the result of the classification (accuracy, recall, etc.) rather than the classification boundary itself, whose intrinsic value is generally disregarded because not physically meaningful.

A more closely related line of work can be found in literature on structural reliability where surrogates are used to decrease the computational effort required for Monte Carlo simulations about a design point. Two main approaches exist for developing surrogates for structural reliability analysis:

1) regression methods (e.g. Response Surface Methods) where the limit score function itself (not just the boundary) is approximated, often with a surrogate regression model such as Artificial Neural Networks (ANNs) or Gaussian Processes (Dai, Zhang, Wang, & Xue, 2012; Roussouly et al., 2012; Sundar & Shields, 2016; Viana, Haftka, & Watson, 2012);
2) classification methods that seek only to ascertain if a design is unacceptable or acceptable.

In structural reliability, regression approaches dominate, but a number of studies have been conducted in recent years using the classification approach (Alibrandi, Alani, & Ricciardi, 2015; Basudhar & Missoum, 2010; Bourinet, Deheeger, & Lemaire, 2011; Gorissen, Couckuyt, Demeester, Dhaene, & Crombecq, 2010; J.E. Hurtado & Alvarez, 2010; Lin, Qiu, Yao, & Wu, 2012; Song, Choi, Lee, Zhao, & Lamb, 2013; Van Der Herten, Couckuyt, Deschrijver, & Dhaene, 2016). Moreover, in a review of surrogate modelling tools, Hurtado (Hurtado, 2004) noted that classification methods are more naturally suited to identification of implicit limit score boundaries. The main justification is in the nature of the problem: one is only interested in the exceedance of score function limit and not in its exact value.

Most of the aforementioned classification studies employed SVMs as the surrogate classifier and some use different active learning strategies to refine boundary estimates. Alibrandi et al. (2015) developed a strategy that generated points in a cone between the nominal design values and the nearest point on the surface (found via a separate optimization problem). Bourinet et al. (2011) developed a subset sampling algorithm that employed an SVM-based active learning strategy which generated new samples close to the SVM boundary by clustering points that are close the boundary. Hurtado and Alvarez (2010) used Particle Swarm Optimisation to find local minima of the score function (i.e. samples that are close to the boundary).

Basudhar and Missoum developed an active learning strategy based on an SVM estimate of the decision function and an auxiliary optimization strategy (Basudhar & Missoum, 2010) aimed at select- ing new samples that provide the highest refinement of the bound- ary. Song et al. (2013) made use of the score function values (not just the sign) and augmented Basudhar and Missoum's method to include "virtual samples" based on a local regression. This most recent line of work is based on the active learning strategy proposed by Basudhar and Missoum (2008) combined with different complex methodologies to alleviate the "locking" phenomenon which results from a strong focus on selecting samples close to the limit score boundary.

In this work, a new SVM-based active learning strategy is developed with the aim of avoiding complex and *ad hoc* anti-locking strategies (Basudhar & Missoum, 2008,2010; Bourinet et al., 2011; Song et al., 2013). To strike a balance in the exploitation-exploration trade-off, an innovative combination of a space-filling strategy and active learning is proposed. In addition to its anti-locking properties, the new methodology enables an approximate *a priori* specification of the resolution error when the designer possesses some modest knowledge of the properties of the boundary.

The general SVM theory will be introduced and described conceptually in the first section of the paper, in combination with a recursive and efficient surface refinement methodology aimed at obtaining the best definition of the limit hyper-surface with the minimum number of numerical simulation runs. The next two sec- tions will present the practical implementation of the methodologies in an efficient algorithm for N-dimensional problems and the validation of the procedure with numerical tests on a-priori

known functions, therefore allowing the exact quantification of the difference between actual limit surface and SVM surrogate. Subsequently, the new methodology will be compared to Basudhar and Missoum (2010) on a benchmark function, showing no loss in performance (on the contrary, slightly improved) despite the significantly simpler sampling strategy. Finally, the algorithm will be tested on a railway engineering application, using ADTRES software (Bruni, Collina, Diana, & Vanolo, 2000) for the simulation of train dynamics.

2. Conceptual approach

2.1. Problem setting and terminology

The type of (design) problem described in the introduction is representable by a score function

$$v = f(\mathbf{x}) \text{ with } \mathbf{x} \in I^S \tag{1}$$

where $I^S = [0, 1]^S$ is the S-dimensional unit hyper-cube, $\mathbf{x}$ the normalised design and operation parameters[1] and v the design score or criterion. The score function v is usually characterised by a critical value v_0, defining the limit for acceptable design.

Therefore, the aim is to identify the boundary $\mathcal{B}$ within the design parameter space $\mathbf{x}$

$$\mathcal{B} = \left\{\mathbf{x} \in I^S : f(\mathbf{x}) = v_0\right\} \tag{2}$$

corresponding to the critical score value v_0.

In typical engineering problems, the subset $\mathcal{B}$ generally consists of a finite number of connected sets (i.e. each subset is a hyper-surface within in the I^S hyper-volume). This characteristic, common to most engineering problems, shows both advantages and challenges: the absence of isolated points in $\mathcal{B}$ suggests the possibility to explore and refine each hyper-surface progressively, once a point of the surface is found. However, the presence of multiple disconnected hyper-surfaces requires a good initial sampling of the space (exploration), to identify at least one point for each.

This study will consider the case where the function $f(\mathbf{x})$ is unknown and therefore the boundary $\mathcal{B}$ has to be estimated on the basis of finite training data $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}$, with $i = 1, \ldots, N$ and $(\mathbf{x}_i, f(\mathbf{x}_i)) \in I^S \times \mathbb{R}$. In the rest of this paper: $\mathbf{x}$ will be referred to as *function input*, f as *score function*, v as *score value* and $\mathcal{B}$ as *boundary set*.

2.2. Limit surface finding as a classification task

Given the binary nature of the problem (score values above and below v_0), estimating the limit surface from a training data set fits within the framework of binary classification problems. However, the purpose of this study is different: instead of seeking an accurate classification *result* we seek an accurate estimation of the *decision boundary* $\mathcal{B}$ that separates the two classes.

Under the classification framework, the boundary $\mathcal{B}$ defines the two subsets $\mathcal{C}_-$ and $\mathcal{C}_+$ of the space I^S, corresponding to points with score function values below and above v_0:

$$\mathcal{C}_- = \{\mathbf{x} \in I^s : f(\mathbf{x}) < v_0\} \text{ and } \mathcal{C}_+ = \{\mathbf{x} \in I^s : f(\mathbf{x}) > v_0\} \tag{3}$$

In typical engineering problems, the two subsets $\mathcal{C}_-$ and $\mathcal{C}_+$ each consist of a finite number of connected sets $\mathcal{C}_{-,k}$ and $\mathcal{C}_{+,\ell}$ (i.e. each subset is composed by a finite number of hyper-volumes within I^s), separated by the hyper-surfaces $\mathcal{B}_{k,\ell}$.

$$\mathcal{B} = \bigcup_{k,\ell} \mathcal{B}_{k,\ell} \tag{4}$$

[1] The normalisation of general real-valued design parameters is obtained by means of simple (linear) transformations.

Analogously, each point can be associated with a class label $c(\mathbf{x}) : I^s \mapsto [-1, 1]$, based on its membership in $\mathcal{C}_+$ or $\mathcal{C}_-$:

$$c(\mathbf{x}) = \begin{cases} 1 & x \in \mathcal{C}_+ \\ -1 & x \in \mathcal{C}_- \end{cases} \tag{5}$$

Following this approach, the training data $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}$ is mapped in a training set $\mathcal{D}$:

$$\mathcal{D} = \{(\mathbf{x}_i, c_i)\} \text{ with } c_i = c(\mathbf{x}_i),\ i = 1, \ldots, N \tag{6}$$

In this work an SVM classifier will be used to optimally sep- arate the two classes in (3) and consider this decision surface to be an estimate of the unknown limit surface $\mathcal{B}$. Since $f(\mathbf{x})$ is often nonlinear, it is natural to select a classifier that supports nonlinear decision boundaries, such as Artificial Neural Networks (ANNs), k-Nearest Neighbours (kNNs), or Support Vector Machines (SVMs). Amongst these approaches, only SVMs include explicit consideration of the decision surface (boundary) during learning: SVMs seek a decision hyper-surface that maximizes the distance (*margin*) from the nearest training points (*support vector*), minimizing the structural classification risk (Burges, 1998; Cortes & Vapnik, 1995). In addition, the SVM framework easily identifies points in $\mathcal{D}$ that are close to the decision boundary, a property that will prove useful in the refinement of the decision boundary (Section 4.5). This study will therefore utilise SVMs for the identification and estimation of the separating boundary $\mathcal{B}$ of the two classes.

2.3. Support vector machines

SVMs, introduced by Vapnik and his colleagues (Cortes & Vapnik, 1995), are a popular classification tool based on statistical learning theory. SVMs were originally formulated for hyper-plane decision surfaces but were extended to nonlinear surfaces via the "kernel trick" (Abe, 2010), i.e. transforming the original features into a new space via a kernel transform. While many variants of SVMs exist for multi-class classification, regression and other tasks, the original binary SVM suits this application for the separation of the classes in Eq. (3). Further details and extensions of SVMs can be found in the copious literature on the subject, e.g. (Abe, 2010; Burges, 1998; Chang & Lin, 2011; Cortes & Vapnik, 1995; Hearst, Dumais, Osuna, Platt, & Schölkopf, 1998).

SVM belongs to the family of maximum margin classifiers, which aim at defining a boundary hyper-surface with maximum "distance" from the separated classes. A traditional SVM boundary is defined by a vector $\mathbf{w} \in \mathbb{R}^q$ and an offset b, forming a hyperplane $(\mathbf{w}, b)$ in $\mathbb{R}^q$:

$$\mathbf{w}^T\mathbf{z} + b = 0 \quad \mathbf{z} \in \mathbb{R}^q \tag{7}$$

The hyper-plane $(\mathbf{w}, b)$ splits the space $\mathbb{R}^q$ in two regions $\hat{\mathcal{Z}}_-$ and $\hat{\mathcal{Z}}_+$:

$$\begin{aligned} \hat{\mathcal{Z}}_- &= \{\mathbf{z} \in \mathbb{R}^q : \mathbf{w}^T\mathbf{z} + b < 0) \\ \hat{\mathcal{Z}}_+ &= \{\mathbf{z} \in \mathbb{R}^q : \mathbf{w}^T\mathbf{z} + b > 0) \end{aligned} \tag{8}$$

This hyper-plane is however able to produce only plane boundaries, which are often insufficient in the representation of many highly nonlinear engineering problems $f(\mathbf{x}) = \mathbf{v_0}$. However, choosing an appropriate nonlinear transformation $\mathbf{z} = \phi(\mathbf{x}) : \mathbb{R}^s \to \mathbb{R}^q$ it is possible to use the same approach to define curved boundaries in $\mathbb{R}^s$. In this case the split of $\mathbb{R}^s$ is executed on the basis of the sign of the *decision value* $d(\mathbf{x})$:

$$d(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b \tag{9}$$

And the resulting predicted classes X_- and X_+ are obtained as:

$$\begin{aligned} \hat{\mathcal{C}}_- &= \left\{\mathbf{x} \in \mathbb{R}^q : \mathbf{w}^T\phi(\mathbf{x}) + b < 0\right\} \\ \hat{\mathcal{C}}_+ &= \left\{\mathbf{x} \in \mathbb{R}^q : \mathbf{w}^T\phi(\mathbf{x}) + b > 0\right\} \end{aligned} \tag{10}$$

The selection of suitable SVM parameters for a good approximation of the boundary $\mathcal{B}$ must therefore lead to the separation of the sets $\mathcal{C}_-, \mathcal{C}_+$ (actually defined by the sign of $f(\mathbf{x}) - v_0$) and $\hat{\mathcal{C}}_-, \hat{\mathcal{C}}_+$ (defined by the decision value $d(\mathbf{x})$). If this is ensured, the limit hyper-surface can be approximated by the SVM boundary $\mathcal{B}$:

$$\hat{\mathcal{B}} = \{\mathbf{z} : d(\mathbf{z}) = 0\} \tag{11}$$

In the classic definition of SVM (Abe, 2010; Chang & Lin, 2011; Cortes & Vapnik, 1995), the optimal values of $\mathbf{w}$ and b are found using the training set $\mathcal{D}_N$ defined in Eq. (6) and the optimization problem:

$$\begin{aligned} \arg\min_{\mathbf{w},b,\xi} &= \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} &= c_i\left(\mathbf{w}^T\phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i \\ &= \xi_i \geq 0,\ i = 1, 2, \ldots, N \end{aligned} \tag{12}$$

In this formulation, a minimum of $\mathbf{w}$ corresponds to the maximum distance between the boundary and the training points of the two classes (*classification margin*), while the combination of the other term with the first constraint represents a penalty for the misclassification of the training set ($\mathbf{x}_i \in \hat{\mathcal{C}}_-$ and $\mathbf{x}_i \notin \mathcal{C}_-$, or $\mathbf{x}_i \in \hat{\mathcal{C}}_+$ and $\mathbf{x}_i \notin \mathcal{C}_+$). Therefore Eq. (12) describes the conflicting objectives of maximizing the classification margin and minimizing the clas- sification errors (Cortes & Vapnik, 1995). The parameter C plays a key role in balancing the two objectives: a high value of C will increase the penalty for misclassification and therefore produce "*harder* " and lower-margin boundaries, whereas a low C will re- sult in lower penalties for misclassifications and therefore generate "*softer* "and higher-margin boundaries. The optimal value of $\mathbf{w}$ can be shown to be also dependent on the weighted sum of the transform of the training set $\phi(\mathbf{x}_i)$ (Cortes & Vapnik, 1995):

$$\mathbf{w} = \sum_{i=1}^{N} c_i\alpha_i\phi(\mathbf{x}_i) \tag{13}$$

The weighting coefficients of each summation term in Eq. (13) are defined by the sign of $c_i = \pm 1$ (the actual class of the training set) and by the magnitude of the Lagrange multipliers $\alpha_i \geq 0$ associated with the first constraint of Eq. (12). By combin- ing Eqs. (9) and (13) and defining the kernel function as $K(\mathbf{x}, \mathbf{y}) = \phi^T(\mathbf{x})\,\phi(\mathbf{y})$, the kernel-based formulation of SVM is obtained:

$$d(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i c_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{14}$$

Usually, $\alpha_i > 0$ only for a subset of the N training points, sufficiently close to the boundary. The points belonging to this subset $\mathcal{V} \subset \mathcal{D}$ are called *support vectors* and their definition limits the summation of Eq. (14) to less than N terms:

$$d(\mathbf{x}) = \sum_{(\mathbf{x}_i, c_i)\in\mathcal{V}} \alpha_i c_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{15}$$

The cardinality of the support vector set is strongly affected by the parameter C: a high value of C emphasises on correct classification and thus encourages the optimisation process to increase the number of support vectors enabling high-curvature portions in the boundary; a low value of C, on the contrary, relaxes the classification constraint, promoting "smoothness" in the surface and therefore few support vectors.

In this paper the radial basis function kernel (Scholkopf et al., 1997) is chosen for $K(\mathbf{x}, \mathbf{y})$:

$$K(\mathbf{x}, \mathbf{y}) = e^{-\gamma||\mathbf{x}-\mathbf{y}||^2} \tag{16}$$

This kernel does not allow an explicit definition of the transform $\phi(\mathbf{x})$ but has been shown effective in a large range of classification examples owing to its ability to reproduce complex boundary shapes, see for instance (Huang, Chen, & Wang, 2007; Kotsia & Pitas, 2007; Madeo, Peres, & Lima, 2016; Rojas & Nandi, 2006). This property is necessary in a highly nonlinear problem where the shape of the boundary can be highly irregular and even show a series of disconnected sets.

In this study, the parameters γ and C are set using a 5-fold cross-validation approach with a logarithmic scale for the parameter grid (Chang & Lin, 2011).[2]

2.4. Ensuring sufficient exploration

The capability of SVM of producing a good estimate of the boundary, as described in the previous section, clearly depends on the training data $\mathcal{D}$ on which it is based. The possibility of actually evaluating the score function $f(\mathbf{x})$ for any arbitrary design solution $\mathbf{x}$ offers, in this case, an additional opportunity compared to most traditional active learning studies where a finite training dataset is given *a priori*.

A good estimation of the boundary is obtained when the training set $\mathcal{D}$ includes sufficient points close to the surface and well-distributed on the whole surface. Without any prior knowledge of the score function $f(\mathbf{x})$ (and thus of the boundary $\mathcal{B}$), a homogeneous space filling represents the most reasonable choice.

In this study, the Sobol sequence $\mathcal{S}_N$ is chosen for this purpose. The Sobol sequence belongs to the family of quasi-random low-discrepancy sequences (Sobol', 1967). The discrepancy H of a set of N points $\{\mathbf{x}\}_N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, defined in the s-dimensional unit hyper-cube $I^s = [0, 1]^s$ (i.e. each point $\mathbf{x}_n = (x_{n,1}, \ldots, x_{n,s})$ with $0 \leq x_{n,s} \leq 1$), is expressed in analytical terms as (Kuipers & Niederreiter, 2012):

$$H(\{\mathbf{x}\}_N) = \max_{\mathbf{r}\in I^s}\left|\frac{P(G_\mathbf{r};\{\mathbf{x}\}_N)}{N} - \prod_{p=1}^{s} r_p\right| \tag{17}$$

where $\mathbf{r} = (r_1, \ldots, r_s) \in I^s$ and $P(G_\mathbf{r};\{\mathbf{x}\}_N)$ is the number of points of $\{\mathbf{x}\}_N$ contained in the hyper-rectangle $G_\mathbf{r} = [0, r_1) \times [0, r_2) \times \cdots \times [0, r_s)$ (i.e. the hyper-rectangle with vertex in $O = (0, \ldots, 0)$ and $\mathbf{r}$). Therefore, a low discrepancy consists of a uniform density of points represented in Eq. (17) by the proportionality between the hyper-volume $G_\mathbf{r}$ and the expected number of set points $\{\mathbf{x}\}_N$ within $G_\mathbf{r}$.

Another interesting property of Sobol sequences is the progressiveness of the low-discrepancy space filling. This is expressed as follows: for any integrable function $f(\mathbf{x})$ over I^S, the Sobol sequence $\mathcal{S}_N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ ensures that:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i) = \int_{I^s} f(\mathbf{x})dV \tag{18}$$

This means that, given a Sobol sequence $\mathcal{S}_N$, the first $N-1$ points constitute a Sobol sequence with the optimal space-filling property (low-discrepancy) for the $N-1$ points. In other words, the N-th point of the Sobol sequence $\mathcal{S}_N$ is placed in the "most empty" section of the space covered by the sequence $\mathcal{S}_{N-1}$. This sequential low-discrepancy property (progressive space-filling) will be exploited in the iterative algorithm described in the following section.

2.5. Refinement of the boundary based on uncertainty sampling

The hypothesis of complete ignorance of the boundary stated in the previous chapter is no longer valid once an estimate $\hat{B}_N$ is obtained by SVM using an initial number of training points N.

[2] The upper and lower bounds for C and γ were set sufficiently wide so that the optimal values were typically not on the bounds.

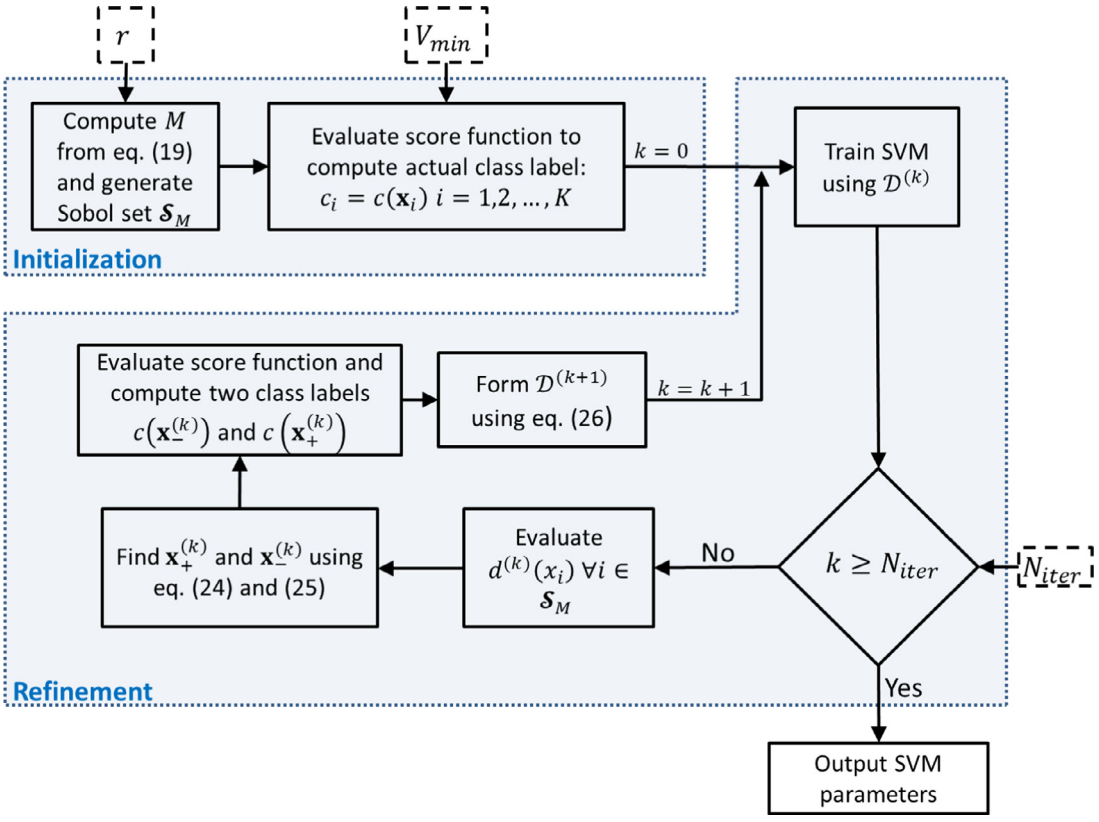

Fig. 1. Summary of the proposed algorithm. Dashed boxes represent the inputs to the algorithm: r is the desired 1D distance between points near the boundary, V_{min} is the minimum single-class volume, and N_{iter} is computed using Eq. (27).

This estimate can be used for further selection of training points likely close to the actual boundary $\mathcal{B}$ according to the active learning paradigm. In doing this, it is important to ensure that all the surface is sufficiently covered by the selected training points.

This is obtained by using a Sobol sequence to generate a pool of potential training points from which subsequent training samples are selected. The density of this sample pool, which represents a discretisation of the solution space, determines the maximum final resolution of the boundary estimation. This limit, imposed to the local refinement of the boundary, effectively limits exploitation already prior to the iterative refinement process. The first iteration of the boundary estimation process is based on a first small subset of the entire Sobol set (which is itself a Sobol set). The density of this subset is crucial in ensuring a full exploration of the space and identification of disconnected sections of the boundary.

Therefore, the proposed procedure is composed of an *initialization* and an iterative *refinement* phase (Fig. 1). Initialisation starts with the generation of a candidate set of points with sufficiently fine resolution in the unit hyper-cube I^s. These points serve as a discretisation of the space I^s. The candidate set is established using a Sobol sequence $\mathcal{S}_M = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ in the unit hyper-cube I^s. The number M of candidate points is selected according to the desired final resolution the grid of points which will define the surface at the end of the algorithm:

$$M \approx \frac{1}{r^s} \tag{19}$$

where r is the desired linear (1D) resolution of the final grid, i.e. the target average distance[3] between two points defining the boundary, and s is the dimensionality of the domain.

[3] This distance is defined in the normalised unit hyper-cube I^s ($r < 1$).

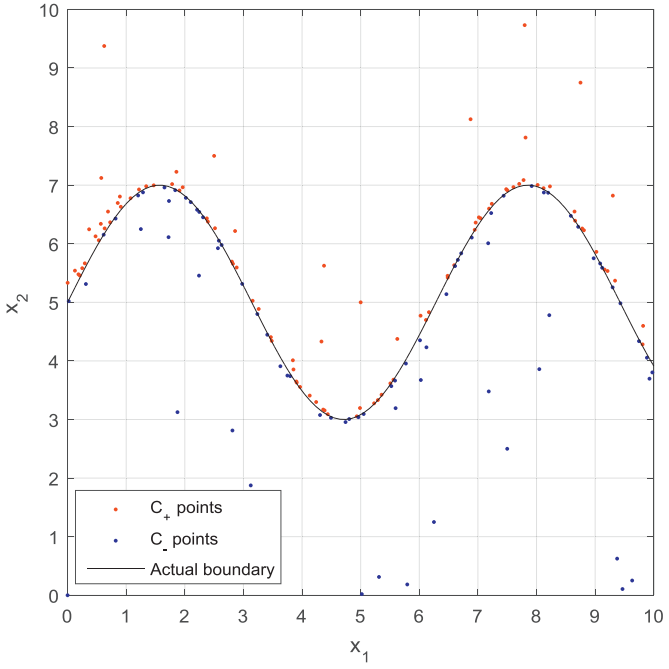

Fig. 2. Result of the SVM iterative refinement procedure applied to the test function.

A first subset of the candidate set $\mathcal{S}_M$ has to be selected for the generation of the first SVM-surrogate without any knowledge or estimate of the boundary $\mathcal{B}$, thus purely focussing on the exploration of the domain I^s. The first K points of $\mathcal{S}_M$ ensure the most uniform exploration of the space, given the property of *subsequent*

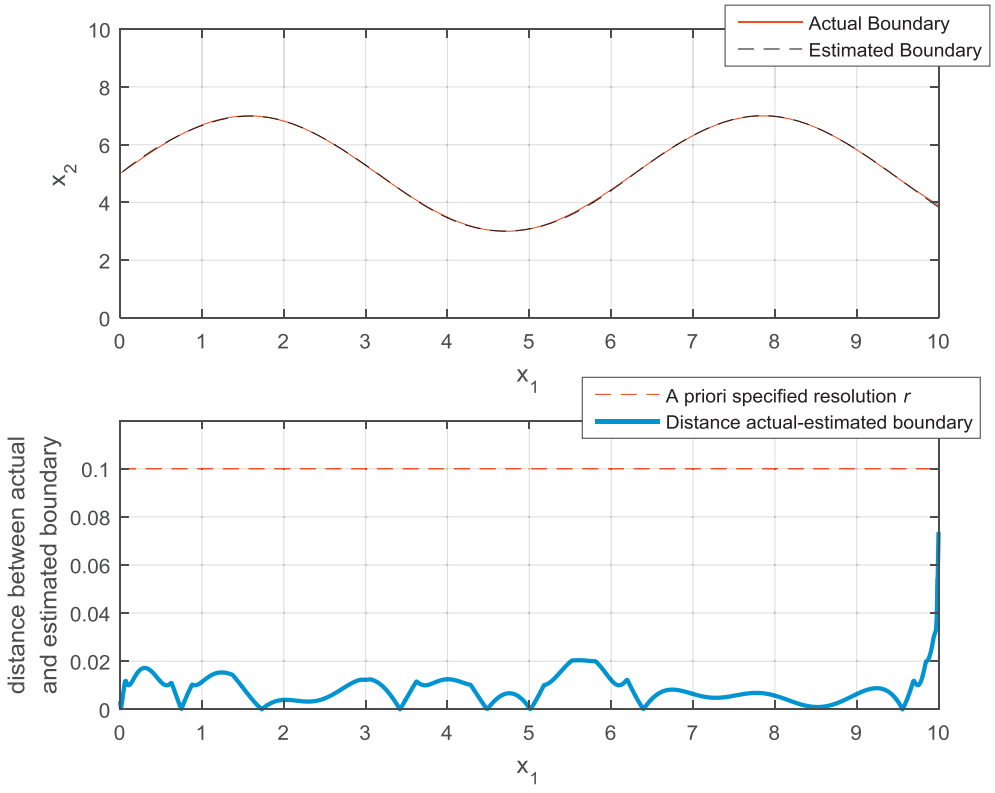

Fig. 3. Comparison of actual vs estimated boundary: direct graphical representation (top), and distance between the two curves (bottom).

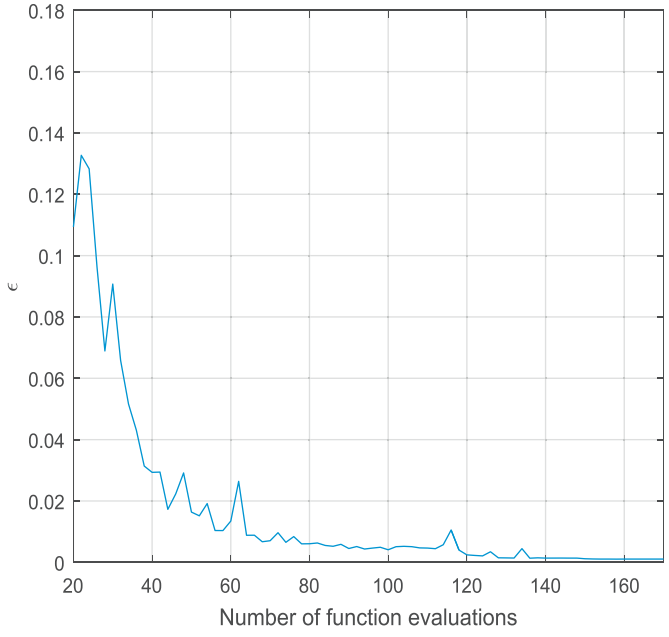

Fig. 4. Evolution of the error measure ϵ as a function of the number of function evaluations. For a comparison with Fig. 11 in Basudhar and Missoum (2010), note that in that reference, the number of function evaluations is equal to $20 + 3\,\boldsymbol{k}$.

low-discrepancy of Sobol sequences, i.e. $\mathcal{S}_K = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ with $K < M$ is also a Sobol sequence. The cardinality of this initial subset is chosen according to an *a priori* hypothesis on the volumes defined by the hyper-surfaces $\mathcal{B}_k$ of the boundary and the limits of I^s. In particular, K is selected as:

$$K \approx \frac{1}{V_{min}} \tag{20}$$

where V_{min} is the smallest volume among the same-class portions of the space:

$$V_{min} = \min_{k,\ell} \left(Vol\left(C_{-,k}\right), Vol(C_{+,\ell}) \right) \tag{21}$$

With this approach it is expected that at least one point will be placed within each sub-volume, thus having at least one point on each side of each hyper-surface $\mathcal{B}_{k,\ell}$. The evaluation of the score function $f(\mathbf{x})$ and the consequent classification (Eqs. (3) and (5)) of the initial subset $\mathcal{S}_K$ results in the initial training dataset $\mathcal{D}^{(0)}$:

$$\mathcal{D}^{(0)} = \{(\mathbf{x}_i, c_i)\} \text{ with } \mathbf{x}_i \in \mathcal{S}_K \tag{22}$$

This allows the first estimate $\hat{\mathcal{B}}^{(0)}$ of the unknown actual boundary $\mathcal{B}$ and in turn the beginning of an iterative refinement of the surface estimate by selecting additional points from the candidate set located close to the estimate of the boundary. The decision value of each point is used as an estimate of proximity to the boundary.

Therefore, each iteration of the *refinement* (Fig. 1) cycle starts with the evaluation of the decision value for all the points of the candidate set $\mathcal{D}$:

$$d^{(k)}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i^{(k)} c_i^{(k)} K^{(k)}(\mathbf{x}, \mathbf{x}_i) + b^{(k)} \tag{23}$$

Two points $\mathbf{x}_+^{(k)}$ and $\mathbf{x}_-^{(k)}$ are selected from class C_+ and C_- respectively, based on their decision values (closest to zero):

$$\mathbf{x}_+^{(k)} = \underset{\mathbf{x} \in C_+ \cap \left(\mathcal{D} \setminus \mathcal{D}^{(k)}\right)}{\arg\min} \; d^{(k)}(\mathbf{x}) \tag{24}$$

$$\mathbf{x}_-^{(k)} = \underset{\mathbf{x} \in C_- \cap \left(\mathcal{D} \setminus \mathcal{D}^{(k)}\right)}{\arg\max} \; d^{(k)}(\mathbf{x}) \tag{25}$$

The selection constraint $\mathcal{D} \setminus \mathcal{D}^{(k)}$ ensures that points already used in previous iterations are excluded. The classes $c(\mathbf{x}_+^{(k)})$ and

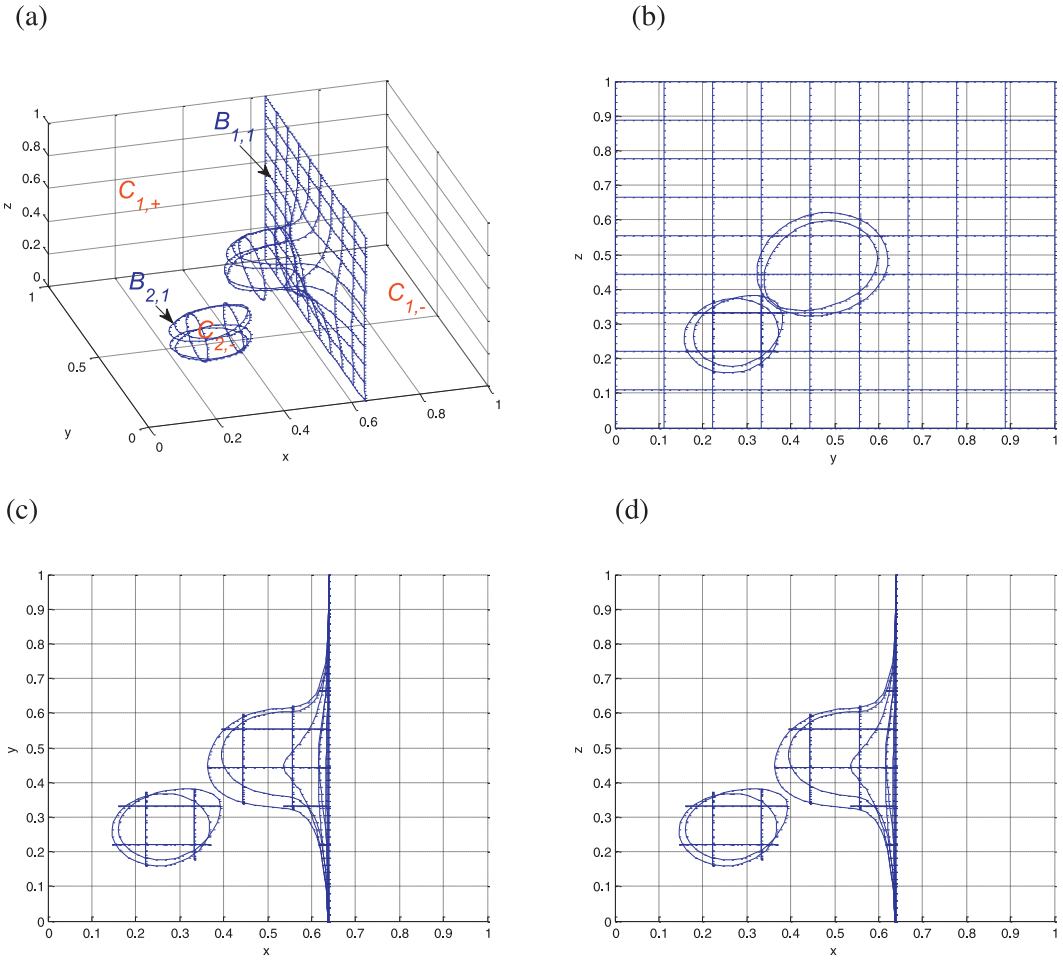

Fig. 5. Graphical representation of the "toy" function: (a) 3D-view, (b) y-z plane, (c) x-y plane, (d) x-z plane.

$c(\mathbf{x}_{-}^{(k)})$ of the new points are evaluated by computing $f(\mathbf{x}_{+}^{(k)})$ and $f(\mathbf{x}_{-}^{(k)})$, therefore allowing the definition of the training set $\mathcal{D}^{(k+1)}$ for the next iteration:

$$\mathcal{D}^{(k+1)} = \mathcal{D}^{(k)} \cup \left\{ \left(\mathbf{x}_{+}^{(k)}, c\ \mathbf{x}_{+}^{(k)} \right), \left(\mathbf{x}_{-}^{(k)}, c\ \mathbf{x}_{-}^{(k)} \right) \right\} \tag{26}$$

The procedure described so far is run for N_{iter} iterations

$$N_{iter} \approx \frac{\alpha}{2 \cdot r^{s-1}} \tag{27}$$

where r is the same linear resolution desired for the grid of points defining the surface, s is the dimensionality of the space and α depends on hypotheses on the dimension of the boundary. In particular, in a three-dimensional case, α represents the expected area of the boundary surface within the normalized parameter space I^3.

Fig. 1 shows a graphical summary of the proposed algorithm. The design choices are shown as dashed boxes. It can be clearly seen that the algorithm requires $K + 2N_{iter}$ evaluations of the scoring function. This quantity represents the computational complexity under the hypothesis of highly expensive (dominant) evaluation of the scoring function (simulations), which motivated this study.

3. Numerical simulations with a-priori known functions

3.1. Two-dimensional case

In this section the procedure described in the previous chapter is applied to the function proposed by Basudhar and Missoum (2010) in order to allow a benchmarking with the most relevant previous work and examine the accuracy of the *a priori* specification of the resolution. The score function has the following analytical form

$$f(\mathbf{x}) = x_2 - 2\sin(x_1) - 5 \tag{28}$$

with $\mathbf{x} = (x_1, x_2)$ representing the raw design parameters, $\mathbf{x} \in [0, 10]^2$.

The relatively simple analytical expression and the low dimensionality of the problem also allows for a more detailed analysis of the quality of the boundary identification result.

After proper normalisation of the design parameters to the unit-area space I^2, the procedure presented in the previous section is applied imposing a resolution $r = 0.01$ (corresponding to a resolution of 0.1 in the original space $[0, 10]^2$). This, with a choice of α between 1 and 2, leads to a suggested number of generations N_{iter} between 50 and 100. In order to ensure fairness in the comparison of the proposed method results with the previously published methodology, N_{iter} is set to 75, and the initial number of points is set to 20 (in this case chosen independently from V_{min}). This corresponds to the same number of initial points and total number of function evaluations as in Basudhar and Missoum (2010), who added to the same initial pool 3 points per each of the 50 iterations.

The following figure shows the result of the SVM iterative refinement procedure. The concentration of the points near the boundary demonstrates the correct behaviour of the active-learning methodology, which successfully identifies points near the boundary. The fairly even distribution of points along the boundary

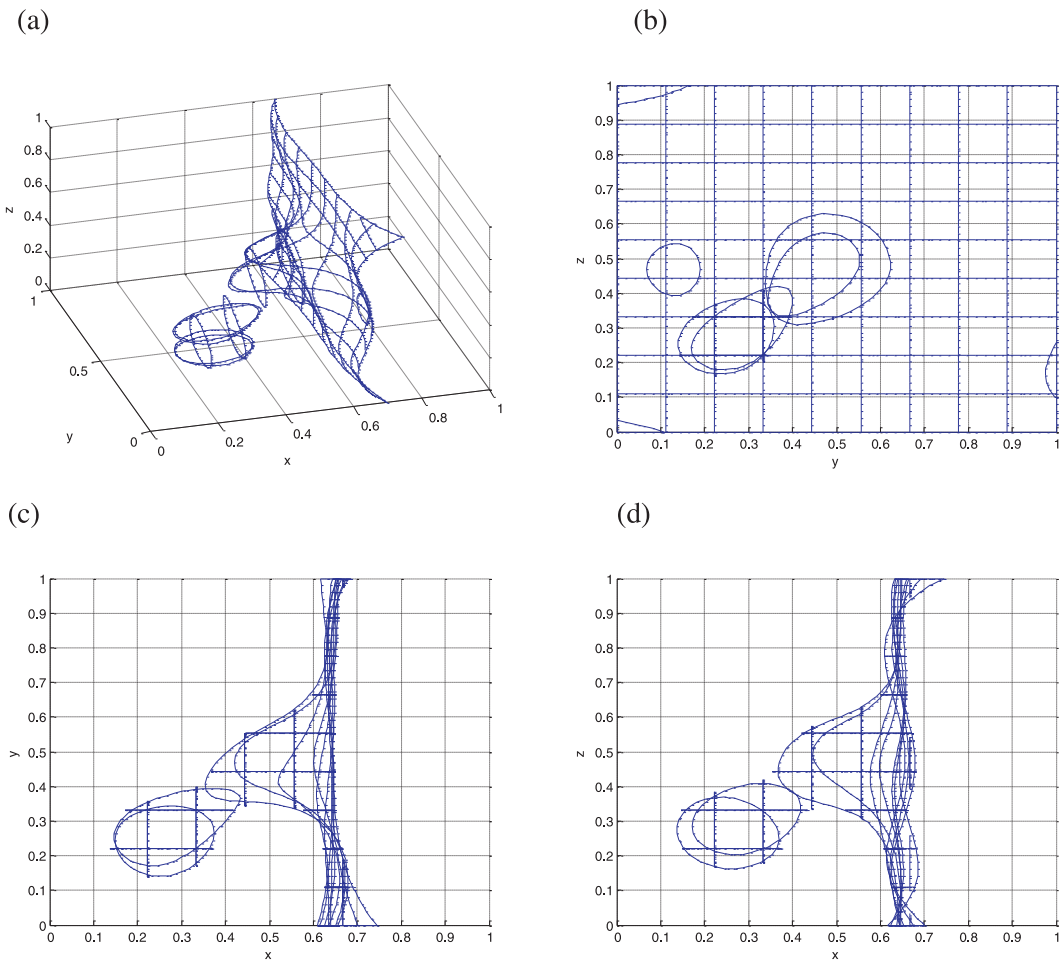

Fig. 6. Result of the boundary identification using the newly proposed algorithm: (a) 3D-view, (b) y-z plane, (c) x-y plane, (d) x-z plane.

shows the additional property of this methodology which intrinsically balances local refinement and coverage of the whole bound- ary.

The result in Fig. 3 can be compared with the corresponding result presented in Fig. 10 of Basudhar and Missoum (2010). The top diagram of Fig. 3 qualitatively confirms the high accuracy of the proposed methodology, which shows a similar (if not superior)
quality to the boundary estimation. The representation of the estimated boundary in this and in the following figures is obtained by a contour command in Matlab.

The bottom diagram shows the Euclidean distance between the boundaries as a function of the coordinate x_1 and has been obtained by discretising the two boundary representations using 1000 uniformly spaced x_1 values. The behaviour of the distance function and of its maximum value support the effectiveness of the *a priori* specification of the resolution target, which is also plotted in the figure.

A final comparison with the methodology proposed by Basudhar and Missoum (2010) is shown in Fig. 4, which represents the evolution of the error measure ϵ used by the same authors. This error measure represents the misclassification on a large set of N_{test} points $\mathbf{x}$, generated randomly (uniformly) in the parameter space:

$$\epsilon = \frac{\text{num}\left(\mathbf{x} \in \left(\hat{\mathcal{C}}_+ \cap \mathcal{C}_-\right) + \text{num}\left(\mathbf{x} \in \left(\hat{\mathcal{C}}_- \cap \mathcal{C}_+\right)\right.\right.}{N_{test}} \tag{29}$$

where num(c) is the number of elements satisfying the condition c.

The result suggests a superior convergence rate and final result, achieving a final $\epsilon \approx 0.1\%$ (0.001), which appear to be below that reported in Basudhar and Missoum (2010) (where $\epsilon \approx 0.005$)

3.2. Three-dimensional case

The procedure proposed in this paper is applied in this section to a known function, in order to assess the accuracy of the method. The "toy" function $f(\boldsymbol{x})$ is $\mathbb{R}^3 \rightarrow \mathbb{R}$ is defined as follows

$$f(\boldsymbol{x}) = \tan^{-1}\left\{ \frac{||\boldsymbol{x} - \boldsymbol{p}_1||^2}{\rho_1^2} \cdot \frac{||\boldsymbol{x} - \boldsymbol{p}_2||^2}{\rho_2^2} \cdot \sin(1.5\pi x + 0.1) \right\} - 0.5 \tag{30}$$

where $\boldsymbol{x} = (x, y, z)$, $||\cdot||$ is the Euclidean norm, $\boldsymbol{p}_1 = (0.5, 0.5, 0.5)$, $\boldsymbol{p}_2 = (0.25, 0.25, 0.25)$ and $\rho_1 = 0.5$ and $\rho_2 = 0.125$. A plot of the function can been seen in Fig. 5.

The function is designed such that the boundary consists of two disconnected sets having strongly different dimensions and shapes. The presence of two sets is intended to test the capability of the algorithm to explore and refine a disconnected boundary. These as- pects challenge in particular the refinement phase of the proposed algorithm: the presence of two disconnected sets of C_- requires a balanced selection of subsequent points for refinement. The defini tion of the localised protrusion on the surface $\mathcal{B}_{1,\ 1}$ will also contribute to the evaluation of the refinement phase. On the other hand, the presence of a small "bubble" (disconnected set $C_{2,\ -}$) validates the ability of the initialisation phase to identify the presence

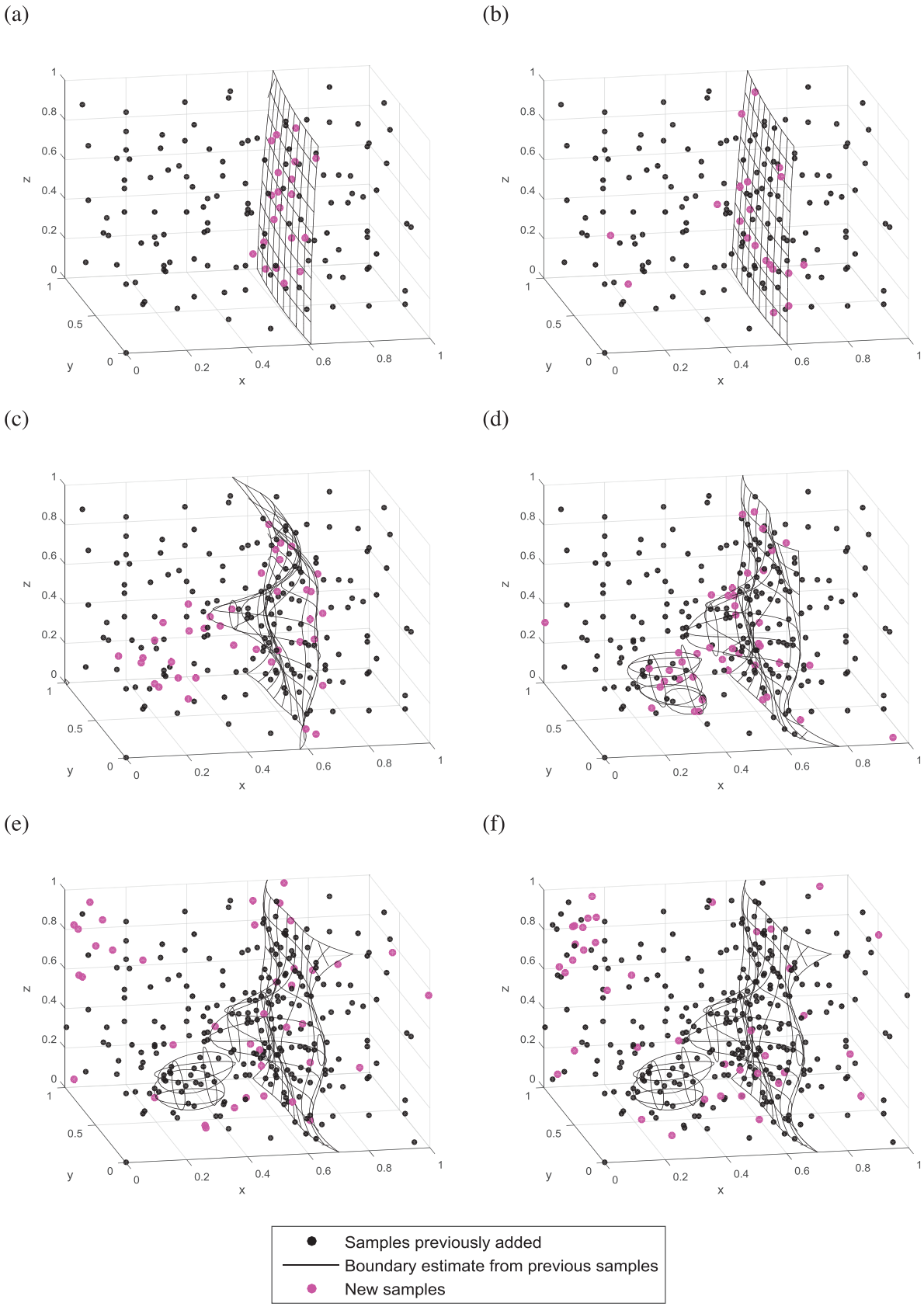

Fig. 7. Evolution of the boundary identification using the newly proposed algorithm: (a) situation at iteration 0, (b) situation at iteration 10, (c) situation at iteration 20, (d) situation at iteration 40 iterations, (e) situation at iteration 60, and (f) situation at iteration 80.

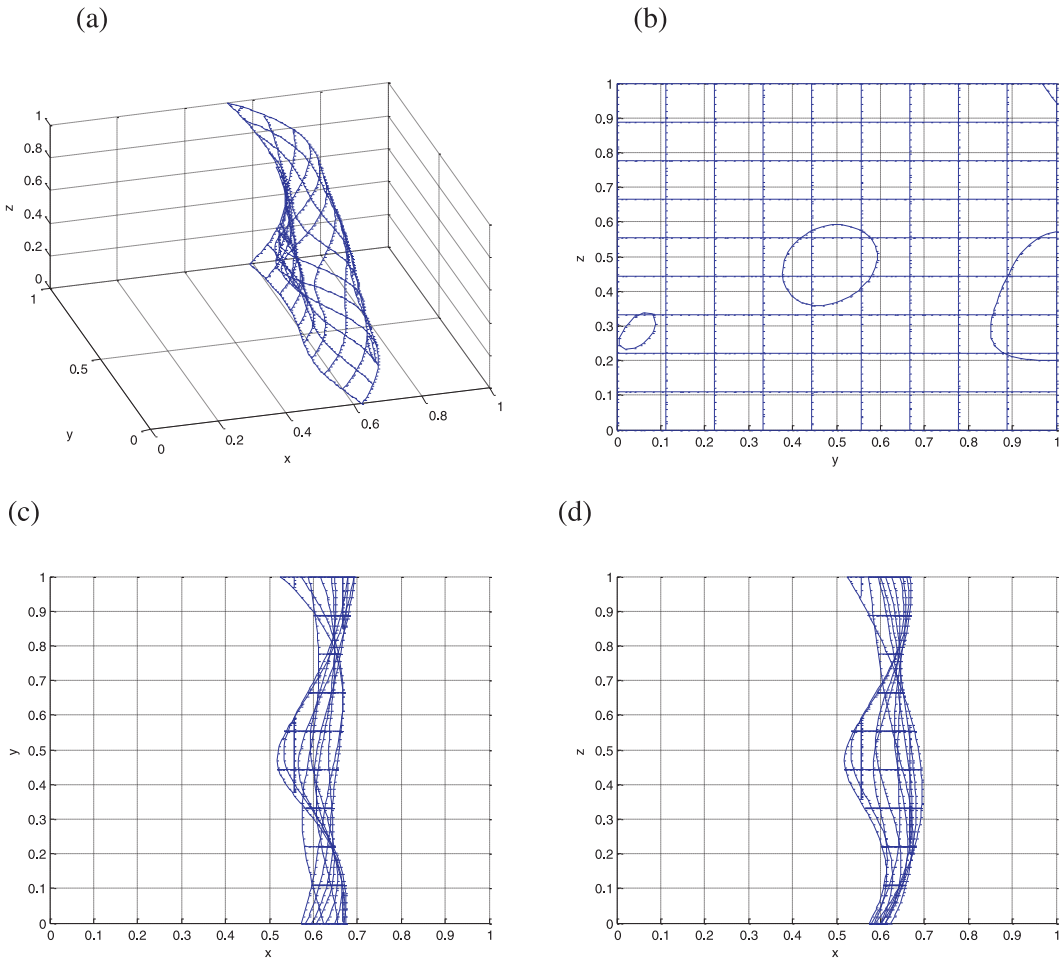

Fig. 8. Result of the boundary identification adopting a purely space filling approach.

of this small volume completely contained within the other volume $C_{1,+}$.

The parameters of the algorithm are set according to the following considerations:

- $M = 1000$ computed using Eq. (19) with a desired 1D resolution of $s = 0.1$
- $K = 125$ computed using Eq. (20) and an expected volume of the smallest set to be Vol $(C_{2,\_}) \approx 0.2^3$
- $N_{iter} = 100$ computed using Eq. (27) with a somehow conservative shape factor $\alpha = 2$ (boundary surface expected to have an area lower than twice that of a face of the unit cube I^3).

The results of the application of the proposed technique are shown in Fig. 6. The definition of the surface is almost perfect (see comparison with Fig. 5), with a maximum error within the chosen resolution. The grid representation of the surfaces in all the following figures is obtained by a contour command in Matlab. Fig. 7 shows the evolution of the boundary identification at dif- ferent iterations. The initial surface estimate, based on the $K = 125$ initial space-filling points captures the dominant feature of the boundary, i.e. the vertical surface, but completely neglects its lo- calised features ("bubble" and protrusion). As expected, the subse- quently samples are selected along this first-approximation bound- ary. Later (Fig. 7 b to c), as the density of points along the main surface grows, connected details of the same are identified (i.e. protrusion in the middle of the main surface). Once the main sur- face is well explored (Fig. 7 c and d), the next samples are selected on the secondary disconnected element ("bubble"). Fig. 7 d shows that, once all the disconnected sets have been identified, the refinement of the two surfaces progresses somewhat uniformly. It is also possible to notice that the final boundary estimation is achieved near iteration 60, with little change thereafter. This is explained by the conservative choice of the parameter $\alpha = 2$, which overestimates the actual boundary surface area, closer to a value of 1.3. If known *a priori* and substituted into Eq. (27), this value of α would yield a number of iterations of 65, which is extremely close to the critical iteration number for which the final estimation is achieved.

In order to benchmark the effectiveness of the refinement procedure versus a pure space-filling approach, the same function is estimated by a single step of SVM fitting over a Sobol set with a number of points equal to $M' = 525$. This corresponds to limiting the proposed algorithm to the initialisation phase with the same computational cost in term of number of evaluations of the score function $f(\mathbf{x})$. The results are displayed in Fig. 8, which clearly shows the inferior performance of this direct approach. In particular, the isolated section of the boundary $\mathcal{B}_{2,\_1}$ is not identified at all, and the shape of $\mathcal{B}_{1,\_1}$ is poorly approximated. The missing recognition of the second surface is due to the scarcity of points obtained in the "bubble" with a purely Sobol-based point selection. This is visible in Fig. 9, showing the different sets obtained with the two approaches. While the points of the iterative algorithm create a rich set of support vectors, the low number of points on the boundary for the second case results in a low overall penalty for the misclassification (e.g. only one point in the set $C_{2,\mathcal{B}_{2,1+}}$) and is therefore insufficient to identify the additional

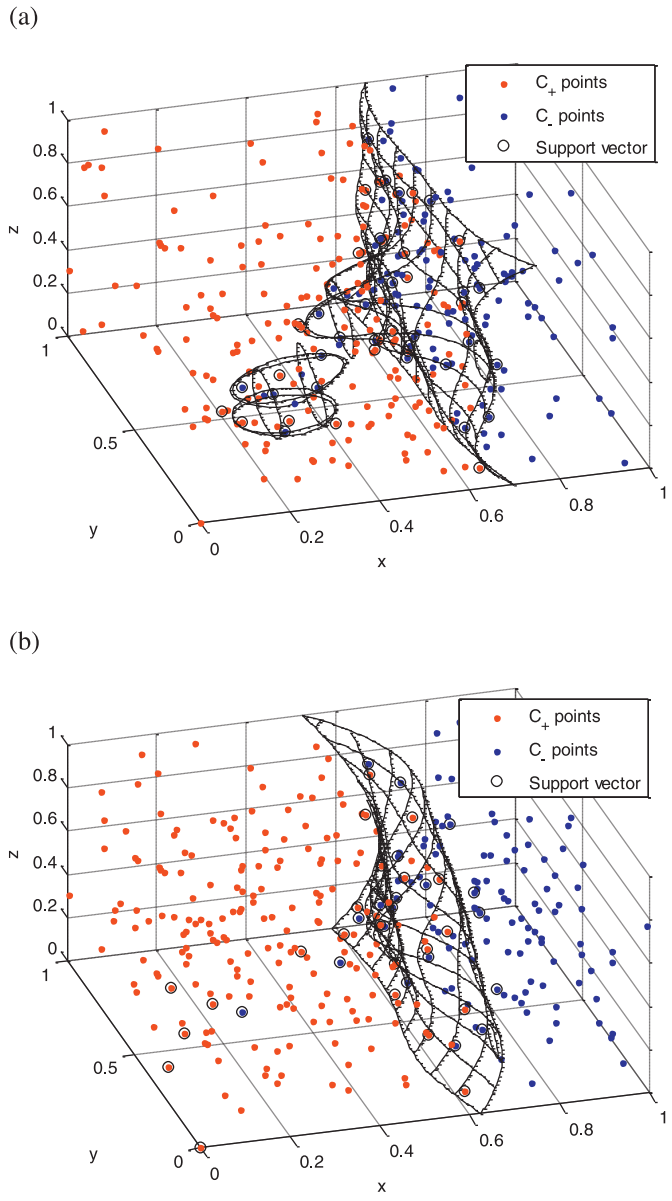

Fig. 9. Different sets causing different performances in (a) the iterative refinement algorithm and (b) the purely space-filling method.

An accuracy evaluation of the result has been obtained by generating 1 million samples from a 3D uniform distribution within the parameter's space and verifying the correct classification of the two classifiers corresponding to Fig. 9 (a) and (b). The boundary identified using the methodology proposed in this paper (Fig. 9(a)) resulted in a classification error $\epsilon = 1.5\%$, while the classification error of the boundary obtained with the same number of samples
using a purely space-filling algorithm provided a classification error of 3.3%.

Since the accuracy measure is only a "volumetric" representation of the error, an additional assessment of "distance" between the actual and the identified surface is introduced. In particular,
this second measure aims at identifying the maximum/average distance of misclassified points from the actual boundary. In a practical design application, knowledge of this distance (or its upper bound) would allow choosing a design solution outside of a "safety margin" from the estimated boundary. In this complex 3D scenario, the Euclidean approach used in Fig. 3 is approximated by:

Table 1
Misclassification distance results for the two boundary estimation algorithms.

	Maximum distance	Average distance
Iterative refinement algorithm	0.0530	0.0002
Purely space-filling method	0.4626	0.0032

- randomly selecting points according to a 3D uniform distribution;
- classifying the points using the SVM model that defines the estimated boundary;
- finding all misclassified points;
- for each misclassified point, calculating its distance from the closest correctly classified point of the same actual class.

The results with 1 million random samples for the two boundaries of Fig. 9 are reported in
Table 1.
The ten-fold improvement of the iterative refinement algorithm is due to the correct identification of local features of the surface (e.g. disconnected set or "bubble") which, despite having lower volume, would constitute a high risk for a designer operating with the purely space-filling model. The quantitative result of maximum
distance obtained with the iterative algorithm is not only showing the relative improvement against the reference case, but also demonstrates that the boundary resolution target (set a-priori to 0.1) is achieved.

4. An application case study: train dynamics

Recent works in railway engineering have proposed numeri- cal vehicle simulation methods used mainly for certification purposes. The study (Bigoni, True, & Engsig-Karup, 2014) presents an approach based on total sensitivity indexes in order to quantify the effect of the uncertainty on suspension parameters on the so-called *vehicle critical speed* (True, 1994). The latter is defined as the minimum speed at which the so-called hunting motion onsets (instability). This has a practical implication; in fact, the vehicle should not run at speeds where it exhibits hunting in order to avoid large dynamic oscillation which can lead to fatigue (or even damage) of vehicle components or can compromise the running safety. Numerical simulations have been used in other recent studies to obtain a simulation-based certification of the rail vehicle (Bezin, Funfschilling, Kraft, & Mazzola, 2015) or to evaluate the effect of the propagation of the variability of the parameters (Funfschilling, Perrin, & Kraft, 2012).

However, the computational cost has so far represented one of the main obstacles to extend this simulation-based approach to the vehicle design phase. This situation, combined with the economical unfeasibility of building actual prototypes for experimental testing, results in a lack of proper design tools available to railway engineers in the design phase.
Some attempts in this field have been made recently: the first approach (Mousavi Bideleh & Berbyuk, 2016) envisaged the use of global sensitivity analysis in order to determine the most influencing parameters on specific phenomena which can reduce the number of input design parameters for the optimization of a bogie suspension system; the second approach (Mousavi Bideleh, Berbyuk,
& Persson, 2016), on the contrary, makes use of a Pareto optimisation based on a multi-objective genetic algorithm in order to define the optimal design parameters of the bogie. However, the results of this work show that even employing a large number of simulations ($> 10\,0\,0$) it is difficult to define the Pareto front with enough accuracy.

The algorithm proposed in this paper is applied to this problem in the attempt to decrease the computational cost of a simulation-

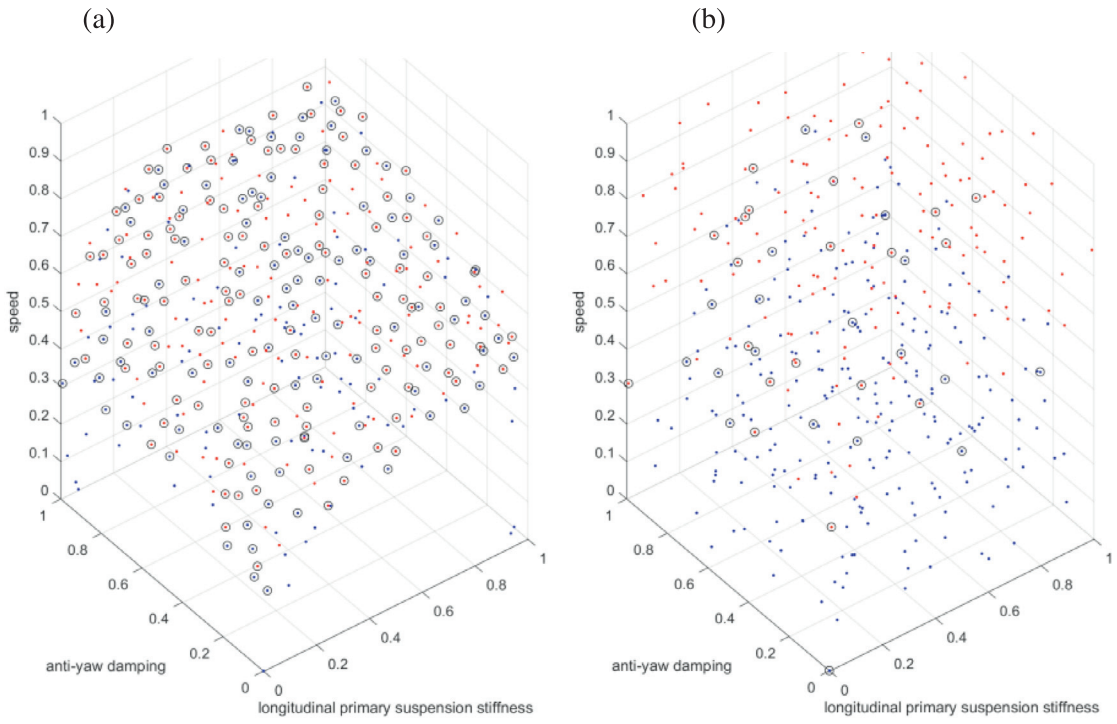

Fig. 10. Results obtained with the proposed iterative refinement algorithm (a) and with the benchmarking case of pure space filling (b).

based approach for vehicle design and design validation. In particular, the evaluation of the design parameters is obtained by the determination of the vehicle's critical speed by means of a vehicle dynamic model. This numerical model is based on a multibody approach where the kinematics of each body is linearized with respect to moving reference frames following the track centreline at the same speed of the vehicle (which is imposed to a constant value). The nonlinearities due to the nonlinear characteristics of the suspensions or to the wheel-rail contact (geometry of the profiles and creep-force characteristics) are fully accounted for. More details on the simulation program can be found in references (Bruni et al., 20 0 0; Di Gialleonardo, Braghin, & Bruni, 2012).

In the framework of this study, the score function $f(\mathbf{x})$ is represented by the vehicle dynamic simulation software, whose input $\mathbf{x}$ is the combination of three selected design/operational parameters considered to have the main influence on the insurgence of hunting motion:

- longitudinal primary suspension stiffness;
- anti-yaw damping;
- vehicle speed.

The output v of the score function is represented by the rms value of the sum of the lateral forces on the wheelset filtered (+ /- 2 Hz) around the hunting frequency. This is chosen in accordance to the standard EN14363, which also provides a threshold value v_0 for the insurgence of the hunting motion. The classification algorithm described in the previous sections is therefore applied to the zero-finding problem $\mathcal{B} = \{ \mathbf{x},\ f(\mathbf{x}) = v_0 \}$.

Fig. 10 reports the results of the analysis for the full algorithm (a) and for the benchmarking case using space-filling only (b). In both cases the total number of simulations (i.e. dots in the left side of the figures) is set to 410. Fig. 11 shows the results obtained with the full procedure described in Section 4.5, with a Sobol set of $M = 4100$, an initial number of points $K = 10$ and refinement iterations $N_{iter} = 200$ (corresponding to a resolution $r \approx 0.0625$ and $\alpha \approx 1.6$).

The benchmarking case is, on the contrary, obtained on a single iteration with a Sobol sequence of 410 points. In both cases the normalisation of the input variables follows the same linear interpolation of plausible design limits.

The proposed approach shows a significantly higher level of detail in comparison with the pure space-filling approach, without any additional computational cost (excluding the iterative SVM fitting). In fact, the presence of a rich set of support vectors allows for the identification of the limit surface in Fig. 10 (a) without any contour operation. On the contrary, the contouring operation (Fig. 11) is necessary for the pure space-filling case (b). Even with the contouring, the definition of the limit surface is much more de- tailed in the iterative algorithm results of Fig. 11 (a), as a conse- quence of the high number of support vectors distributed along the different areas of the boundary.

The results of the analysis confirm the typical considerations made by railway engineers: it is observed, as expected, that increasing both the longitudinal stiffness of the primary suspension and the anti-yaw damping the critical speed increases. The identified boundary allows also the quantification of the effect of each parameter on the critical speed.

The algorithm also provides a rich and well-distributed set of points (support vector) on the boundary itself, representing a computationally cheap proxy for the calculation of a "distance" measure from the boundary. This is in turn extremely valuable for the evaluation of the risks of production variability and equipment degradation over the design parameters (stiffness and damping), which would result in a spread of the expected design condition along the two horizontal axes.

For instance, the identified boundary is very steep close to the vertical plane represented by the null normalised longitudinal suspension stiffness. Thus, stiffness variations in the lower part of the considered stiffness range are more risky.

Considering the desired distance from the surface and the expected variability of the vehicle properties in operation, it is pos-

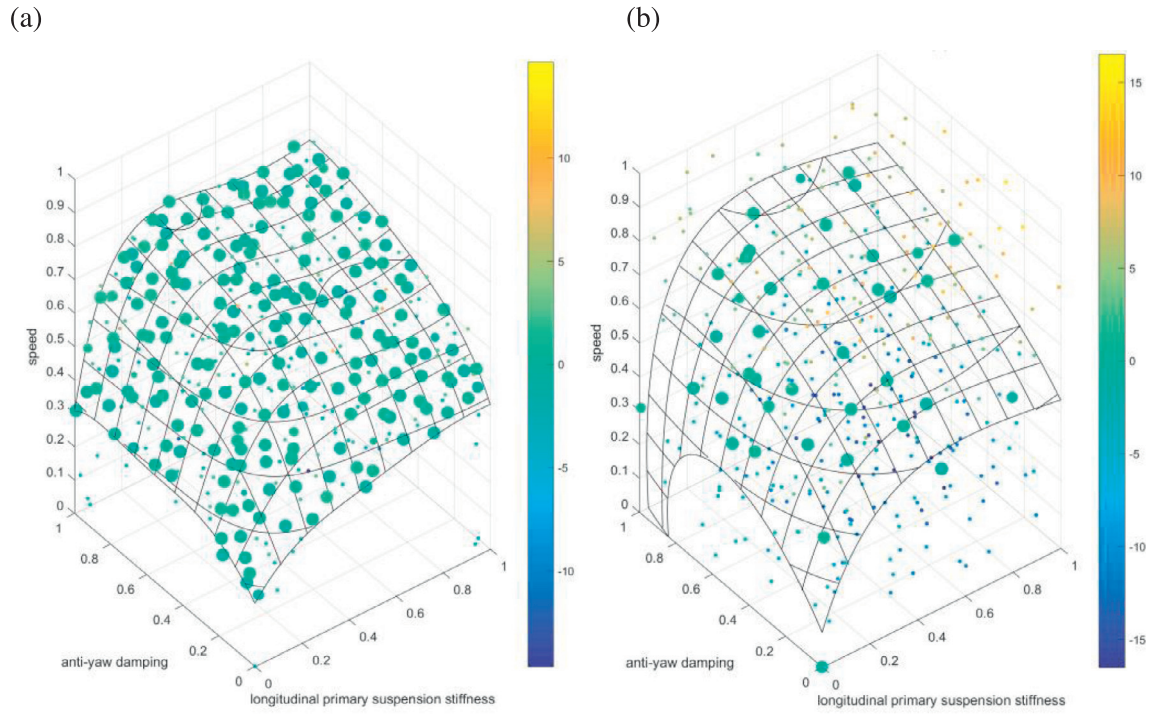

Fig. 11. Contoured surfaces obtained from the results of the proposed iterative refinement algorithm (a) and with the benchmarking case of pure space filling (b). Large points represent support vectors while the color represents the decision value of each point.

sible to establish the optimal design parameters (adding cost constraints).

5. Conclusions and future work

In this paper, a new methodology for the efficient numerical estimation of complex multi-dimensional boundaries has been proposed. This problem is typical of engineering design criteria defining the boundary between acceptable and unacceptable design parameter values. The proposed procedure is particularly valuable when the number of parameters and/or the complexity of the criterion requires computationally expensive numerical simulations. A support vector machine (SVM) is used to identify the boundary hyper-surface from a set of simulation results. The combination of a low-discrepancy sequence (Sobol) of parameter values and an SVM-based active learning approach for the selection of subsequent simulations is the key factor that allows the efficient refinement of the boundary estimation. The selection of informative samples from the Sobol set enables a balancing of exploration and exploitation without complex *ad hoc* diversification mechanisms present in previous studies while achieving similar (if not superior) boundary accuracy.

The methodology has been applied to two highly-nonlinear known functions to demonstrate the accuracy of the resulting boundary estimation and *a priori* resolution target. Subsequently, the algorithm has been utilised in a real-world engineering design application: the acceptability of train suspension parameters in terms of running stability.

References

Abe, S. (2010). *Support vector machines for pattern classification*. London: Springer-Verlag (Second).

Alibrandi, U., Alani, A. M., & Ricciardi, G. (2015). A new sampling strategy for SVM-based response surface for structural reliability analysis. *Probabilistic Engineering Mechanics, 41*, 1–12. https://doi.org/10.1016/j.probengmech.2015.04.001.

Basudhar, A., & Missoum, S. (2008). Adaptive explicit decision functions for probabilistic design and optimization using support vector machines. *Computers & Structures, 86*(19–20), 1904–1917. https://doi.org/10.1016/j.compstruc.2008.02.008.

Basudhar, A., & Missoum, S. (2010). An improved adaptive sampling scheme for the construction of explicit boundaries. *Structural and Multidisciplinary Optimization, 42*(4), 517–529.

Bezin, Y., Funfschilling, C., Kraft, S., & Mazzola, L. (2015). Virtual testing environment tools for railway vehicle certification. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 229*(6), 755–769.

Bigoni, D., True, H., & Engsig-Karup, A. P. (2014). Sensitivity analysis of the critical speed in railway vehicle dynamics. *Vehicle System Dynamics, 52*(sup1), 272–286.

Bourinet, J. M., Deheeger, F., & Lemaire, M. (2011). Assessing small failure probabilities by combined subset simulation and support vector machines. *Structural Safety, 33*(6), 343–353. https://doi.org/10.1016/j.strusafe.2011.06.001.

Bruni, S., Collina, A., Diana, G., & Vanolo, P. (2000). Lateral dynamics of a railway vehicle in tangent track and curve: Tests and simulation. *Vehicle System Dynamics, 33* (SUPPL), 464–477.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*, 121–167.

Chang, C.-C., & Lin, C.-J. (2011). Libsvm. *ACM Transactions on Intelligent Systems and Technology, 2*(3), 1–27.

Chen, H. L., Yang, B., Liu, J., & Liu, D. Y. (2011). A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis. *Expert Systems with Applications, 38* (7), 9014–9022.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

Dai, H., Zhang, H., Wang, W., & Xue, G. (2012). Structural reliability assessment by local approximation of limit state functions using adaptive markov chain simulation and support vector regression. *Computer-Aided Civil and Infrastructure Engineering, 27*(9), 676–686. https://doi.org/10.1111/j.1467-8667.2012.00767.x.

Di Gialleonardo, E., Braghin, F., & Bruni, S. (2012). The influence of track modelling options on the simulation of rail vehicle dynamics. *Journal of Sound and Vibration, 331*(19), 4246–4258.

Funfschilling, C., Perrin, G., & Kraft, S. (2012). Propagation of variability in railway dynamic simulations: Application to virtual homologation. *Vehicle System Dynamics, 50*(sup1), 245–261.

Gorissen, D., Couckuyt, I., Demeester, P., Dhaene, T., & Crombecq, K. (2010). A surrogate modelling and adaptive sampling toolbox for computer based design. *Journal of Machine Learning Research, 11*, 2051–2055.

Goswami, S., Ghosh, S., & Chakraborty, S. (2016). Reliability analysis of structures by iterative improved response surface method. *Structural Safety, 60*, 56–66. https: //doi.org/10.1016/j.strusafe.2016.02.002.

Guyon, I., Cawley, G., Dror, G., & Lemaire, V. (2011). Results of the active learning challenge. In *Proceedings of the workshop on active learning and experimental design: 16* (pp. 19–45).

Hachicha, W., & Ghorbel, A. (2012). A survey of control-chart pattern-recognition literature (1991–2010) based on a new conceptual classification scheme. *Computers and Industrial Engineering, 63* (1), 204–222. https://doi.org/10.1016/j.cie.2012.03.002.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Schölkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and Their Applications, 13*(4), 18–28.

Ho, C., Tsai, M., & Lin, C. (2011). Active learning and experimental design with SVMs. *Journal of Machine Learning Research, 16*, 71–84.

Huang, C.-L., Chen, M.-C., & Wang, C.-J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33*(4), 847–856. https://doi.org/10.1016/j.eswa.2006.07.007.

Huang, J., Shao, X., & Wechsler, H. (1998). Face pose discrimination using support vector machines (SVM). In *Proceedings. fourteenth international conference on pattern recognition (Cat. No.98EX170): 1* https://doi.org/10.1109/ICPR.1998.711102.

Hurtado, J. E. (2004). An examination of methods for approximating implicit limit state functions from the viewpoint of statistical learning theory. *Structural Safety, 26*(3), 271–293. https://doi.org/10.1016/j.strusafe.2003.05.002.

Hurtado, J. E., & Alvarez, D. A. (2010). An optimization method for learning statistical classifiers in structural reliability. *Probabilistic Engineering Mechanics, 25*(1), 26–34. https://doi.org/10.1016/j.probengmech.20 09.05.0 06.

Kim, H.-E., Tan, A. C. C., Mathew, J., & Choi, B.-K. (2012). Bearing fault prognosis based on health state probability estimation. *Expert Systems with Applications, 39*(5), 5200–5213.

Kotsia, I., & Pitas, I. (2007). Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Transactions on Image Processing, 16*(1), 172–187. https://doi.org/10.1109/TIP.2006.884954.

Kremer, J., Steenstrup Pedersen, K., & Igel, C. (2014). Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4*(4), 313–326.

Kuipers, L., & Niederreiter, H. (2012). *Uniform distribution of sequences*. Dover Publications.

Lewis, D., & Gale, A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. New York: Springer-Verlag.

Lin, K., Qiu, H., Yao, Z., & Wu, T. (2012). A local minimum sampling strategy for the construction of boundaries using support vector machines. In *Proceedings of the 2012 IEEE 16th international conference on computer supported cooperative work in design (CSCWD)* (pp. 299–303).

Lu, C.-J., Shao, Y. E., & Li, P.-H. (2011). Mixture control chart patterns recognition using independent component analysis and support vector machine. *Neurocomputing, 74*(11), 1908–1914. https://doi.org/10.1016/j.neucom.2010.06.036.

Madeo, R. C. B., Peres, S. M., & Lima, C. A. D. M. (2016). Gesture phase segmentation using support vector machines. *Expert Systems with Applications, 56*, 100–115. https://doi.org/10.1016/j.eswa.2016.02.021.

Mousavi Bideleh, S. M., & Berbyuk, V. (2016). Global sensitivity analysis of bogie dynamics with respect to suspension components. *Multibody System Dynamics, 37*(2), 145–174.

Mousavi Bideleh, S. M., Berbyuk, V., & Persson, R. (2016). Wear/comfort pareto optimisation of bogie suspension. *Vehicle System Dynamics, 54*(8), 1–24.

Musselman, M., & Djurdjanovic, D. (2012). Time-frequency distributions in the classification of epilepsy from EEG signals. *Expert Systems with Applications, 39*(13), 11413–11422.

Rojas, A., & Nandi, A. K. (2006). Practical scheme for fast detection and classification of rolling-element bearing faults using support vector machines. *Mechanical Systems and Signal Processing, 20*(7), 1523–1536. https://doi.org/http://dx.doi.org/10.1016/j.ymssp.2005.05.002.

Roussouly, N., Petitjean, F., & Salaun, M. (2012). A new adaptive response surface method for reliability analysis. *Probabilistic Engineering Mechanics, 32*, 103–115. https://doi.org/10.1016/j.probengmech.2012.10.001.

Samanta, B. (2003). Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection. *Engineering Applications of Artificial Intelligence, 16* (7–8), 657–665.

Scholkopf, B., Sung, K., Burges, C., Girosi, F., Niyogi, P., Poggio, T., et al. (1997). Comparing support vector machines witn gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing, 45* (11), 2758–2765.

Settles, B. (2009). Active learning literature survey. *Active learning literature survey*: 1648. University of Wisconsin–Madison *Computer sciences technical report*.

Sobol', I. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics, 7*(4), 86–112.

Song, H., Choi, K. K., Lee, I., Zhao, L., & Lamb, D. (2013). Adaptive virtual support vector machine for reliability analysis of high-dimensional problems. *Structural and Multidisciplinary Optimization, 47*(4), 479–491. https://doi.org/10.1007/s00158-012-0857-6.

Sun, Z., Wang, J., Li, R., & Tong, C. (2017). LIF: A new Kriging based learning function and its application to structural reliability analysis. *Reliability Engineering & System Safety, 157*, 152–165. https://doi.org/10.1016/j.ress.2016.09.003.

Sundar, V. S., & Shields, M. D. (2016). Surrogate-enhanced stochastic search algorithms to identify implicitly defined functions for reliability analysis. *Structural Safety, 62*, 1–11. https://doi.org/10.1016/j.strusafe.2016.05.001.

True, H. (1994). Does a critical speed for railroad vehicles exist? *railroad conference, 1994.*. In *Proceedings of the 1994 ASME/IEEE joint (in conjunction with area 1994 annual technical conference)*.

Van Der Herten, J., Couckuyt, I., Deschrijver, D., & Dhaene, T. (2016). Adaptive classification under computational budget constraints using sequential data gathering. *Advances in Engineering Software, 99*, 137–146. https://doi.org/10.1016/j.advengsoft.2016.05.016.

Viana, F. A. C., Haftka, R. T., & Watson, L. T. (2012). Sequential sampling for contour estimation with concurrent function evaluations. *Structural and Multidisciplinary Optimization, 45*(4), 615–618. https://doi.org/10.1007/s00158-011-0733-9.

Widodo, A., & Yang, B. S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing, 21*(6), 2560–2574.